FunboxEasyEnum

Recon/Enumeration

Initiating several scans simultaneously so we never stop enumerating stuff.

```
maxxis@kali:~/Desktop/offsec-labs/FunboxEasyEnum$ sudo nmap --top-ports 25 -Pn 192.168.76.132
[sudo] password for maxxis:
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-21 12:05 IST
Nmap scan report for 192.168.76.132
Host is up (0.25s latency).

PORT     STATE  SERVICE
21/tcp   closed ftp
22/tcp   open   ssh
23/tcp   closed telnet
25/tcp   closed smtp
53/tcp   closed domain
80/tcp   open   http
110/tcp  closed pop3
111/tcp  closed rpcbind
135/tcp  closed msrpc
139/tcp  closed netbios-ssn
143/tcp  closed imap
199/tcp  closed smux
443/tcp  closed https
```

```
445/tcp  closed microsoft-ds
587/tcp  closed submission
993/tcp  closed imaps
995/tcp  closed pop3s
1025/tcp closed NFS-or-IIS
1720/tcp closed h323q931
1723/tcp closed pptp
3306/tcp closed mysql
3389/tcp closed ms-wbt-server
5900/tcp closed vnc
8080/tcp closed http-proxy
8888/tcp closed sun-answerbook

Nmap done: 1 IP address (1 host up) scanned in 13.50 seconds
```

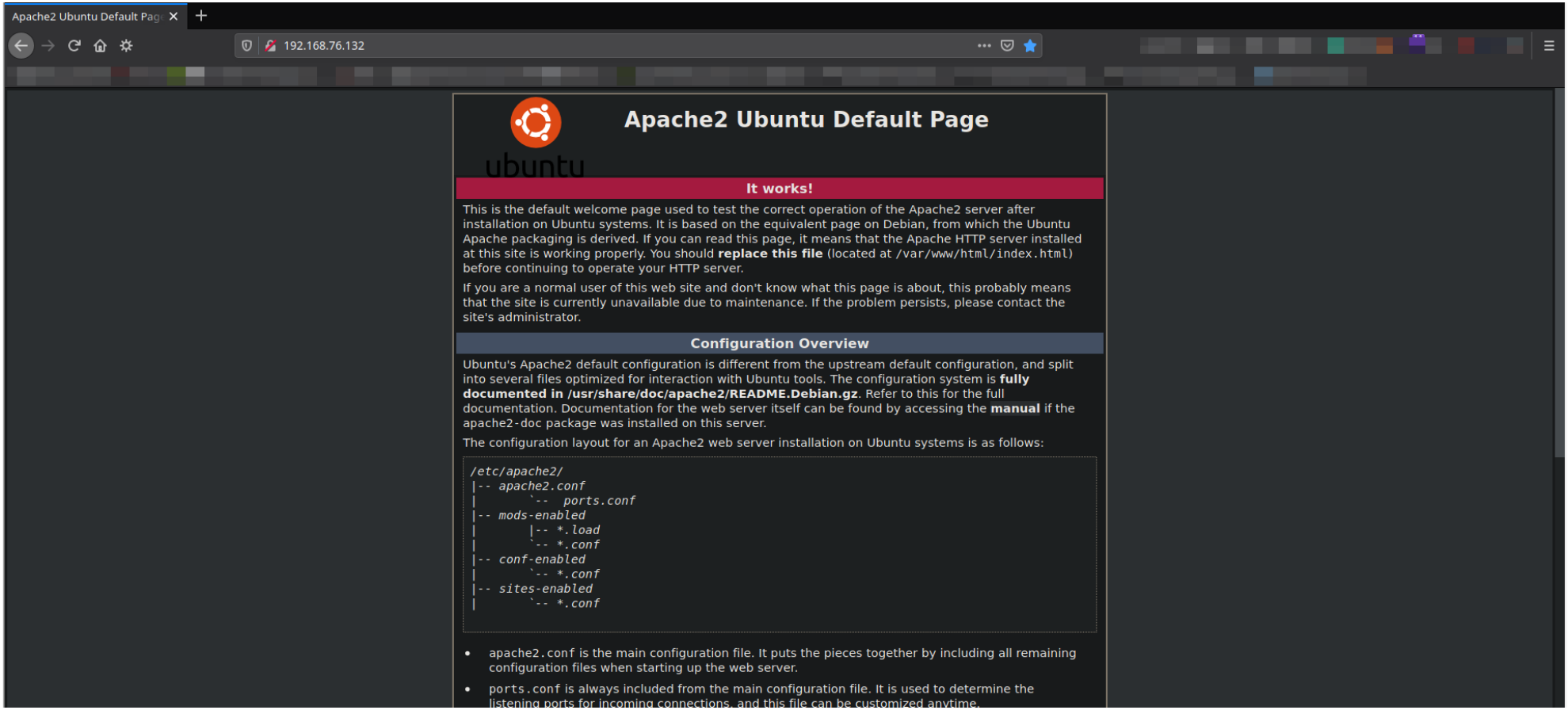
I always prefer running this `--top-ports` flag with nmap just to get quick overview of the ports, like if find port 21 i.e ftp open, I can directly hop onto enumerating FTP port without any further delay. So, from above results I have port 80 and port 22 open. As I have port 80 open, I can initiate nikto scan for better results with respective of port 80.

```
maxxis@kali:~/Desktop/offsec-labs/FunboxEasyEnum$ nikto --host 192.168.76.132
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = "en_US",
    LC_ALL = (unset),
    LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.76.132
```

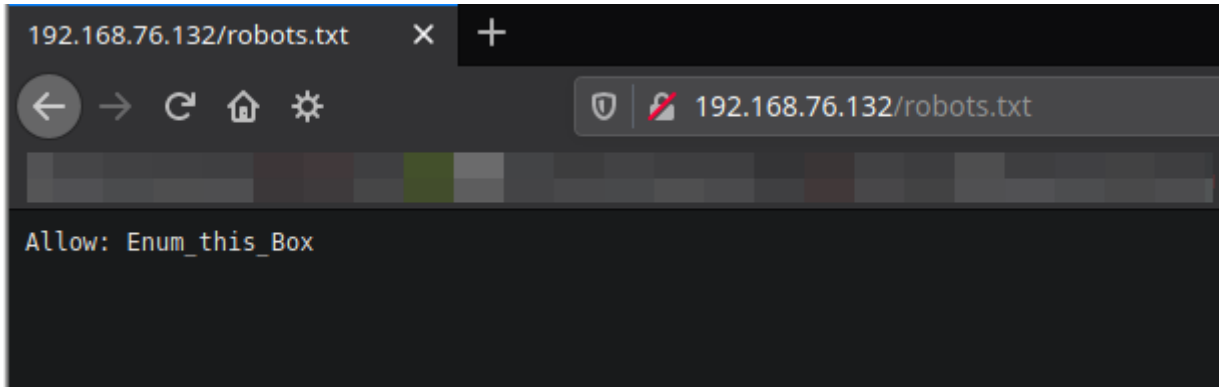
```
+ Target Hostname:    192.168.76.132
+ Target Port:        80
+ Start Time:         2021-05-21 12:05:48 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of
XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a
different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x
branch.
+ Server may leak inodes via ETags, header found with file /, inode: 2aa6, size: 5af9903d91639, mtime: gzip
+ Allowed HTTP Methods: POST, OPTIONS, HEAD, GET
+ Uncommon header 'x-ob_mode' found, with contents: 1
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpmyadmin/: phpMyAdmin directory found
+ 8069 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2021-05-21 12:42:41 (GMT5.5) (2213 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Also, incase if I miss any part of enumeration, I use Autorecon to full proof the reconnaisance process.

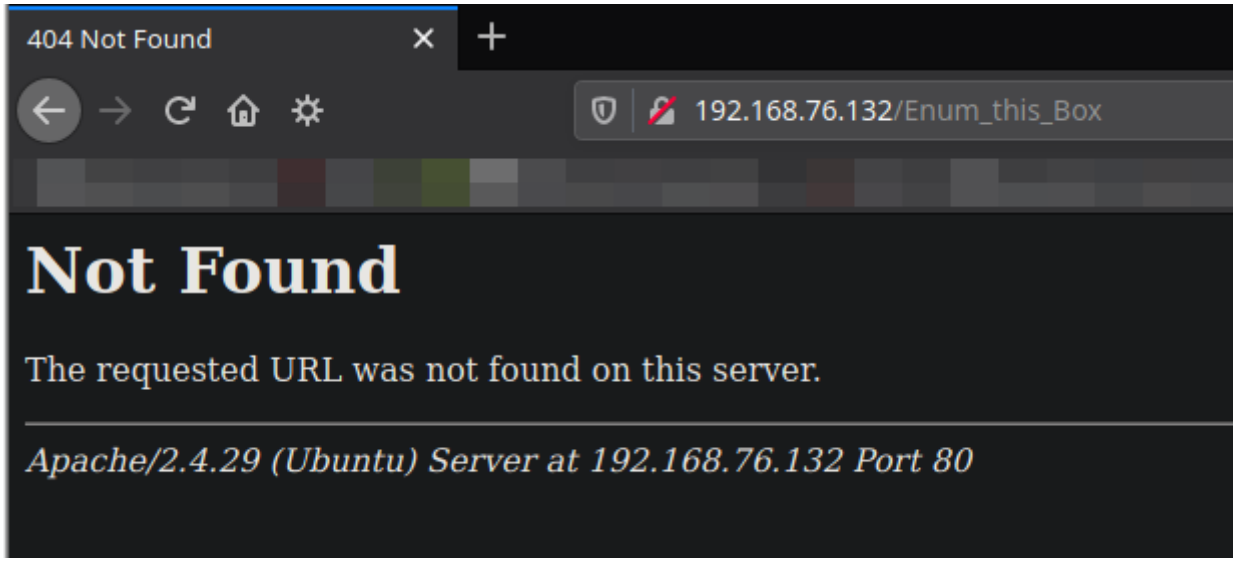
Let's check out the webpage in the browser.



All I found is apcahe default web-page, with no hidden information/clues in its source code. Trying some default directories like `/admin` `/login` `/register` `/index.html` before initiating gobuster directory bruteforcing scan (try it with extensions for instance, `/admin.php` `/admin`). I found nothing, then came across `/robots.txt` and looks like we found something. It looked like...



I tried visiting that directory, and I think we need to enumerate more because it is just a rabbit hole.



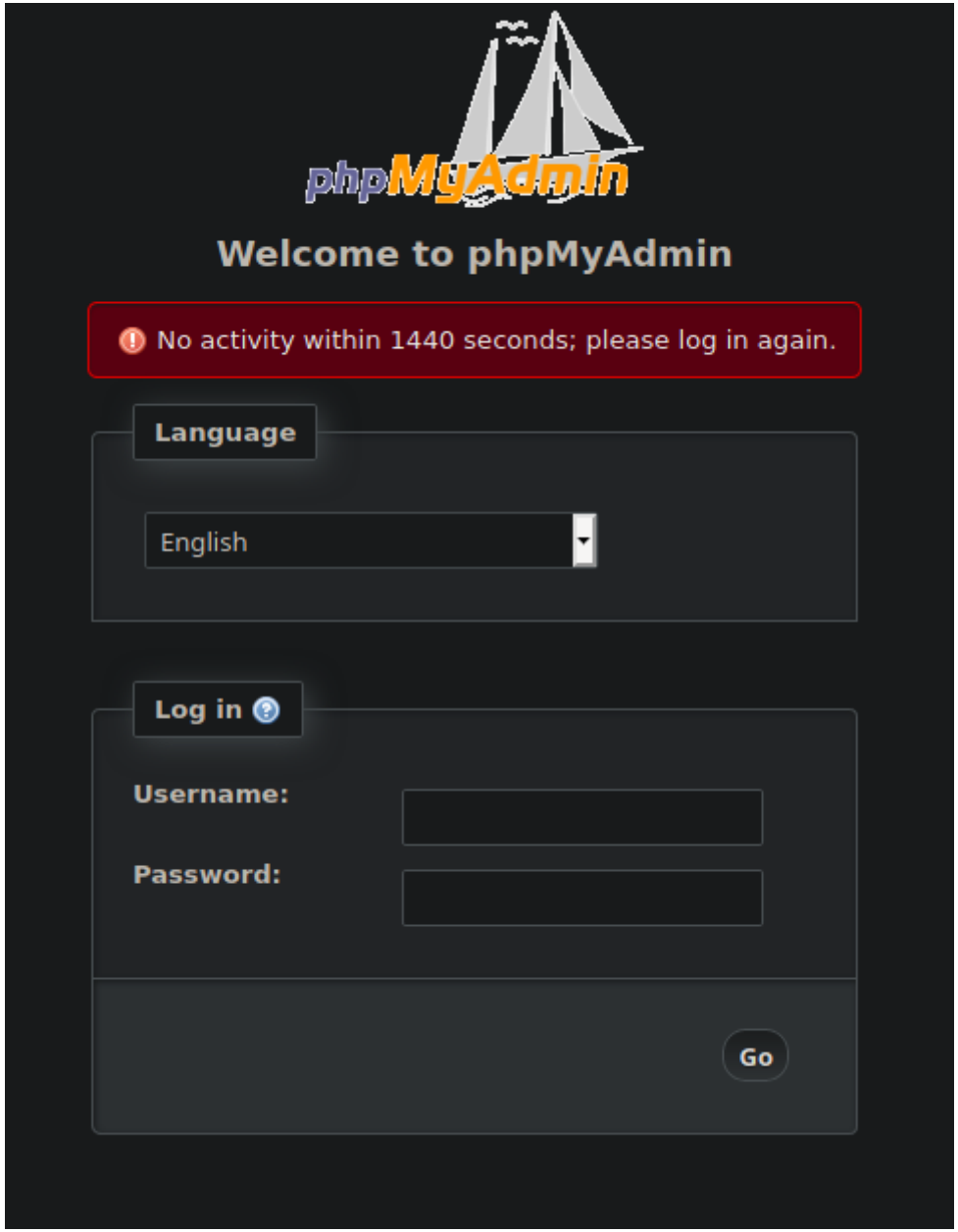
After this cramming up with some manual directory bruteforcing, I initiated the gobuster directory bruteforcing tool. And seems like we have something here.

```
maxxis@kali:~/Desktop/offsec-labs/FunboxEasyEnum$ gobuster dir -u http://192.168.76.132/ -w
/usr/share/seclists/Discovery/Web-Content/raft-small-directories-lowercase.txt -t 10 -x html,php,txt
===============================================================
```

```
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.76.132/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/raft-small-directories-lowercase.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              html,php,txt
[+] Timeout:                 10s
===============================================================
2021/05/21 12:07:27 Starting gobuster in directory enumeration mode
===============================================================
/javascript           (Status: 301) [Size: 321] [--> http://192.168.76.132/javascript/]
/index.html           (Status: 200) [Size: 10918]
/phpmyadmin           (Status: 301) [Size: 321] [--> http://192.168.76.132/phpmyadmin/]
/robots.txt           (Status: 200) [Size: 21]
/mini.php             (Status: 200) [Size: 3828]
/server-status        (Status: 403) [Size: 279]

===============================================================
2021/05/21 12:38:12 Finished
===============================================================
```

So now we have 2 interesting directories here, `/phpmyadmin` and `/mini.php` I started with `/phpmyadmin` .

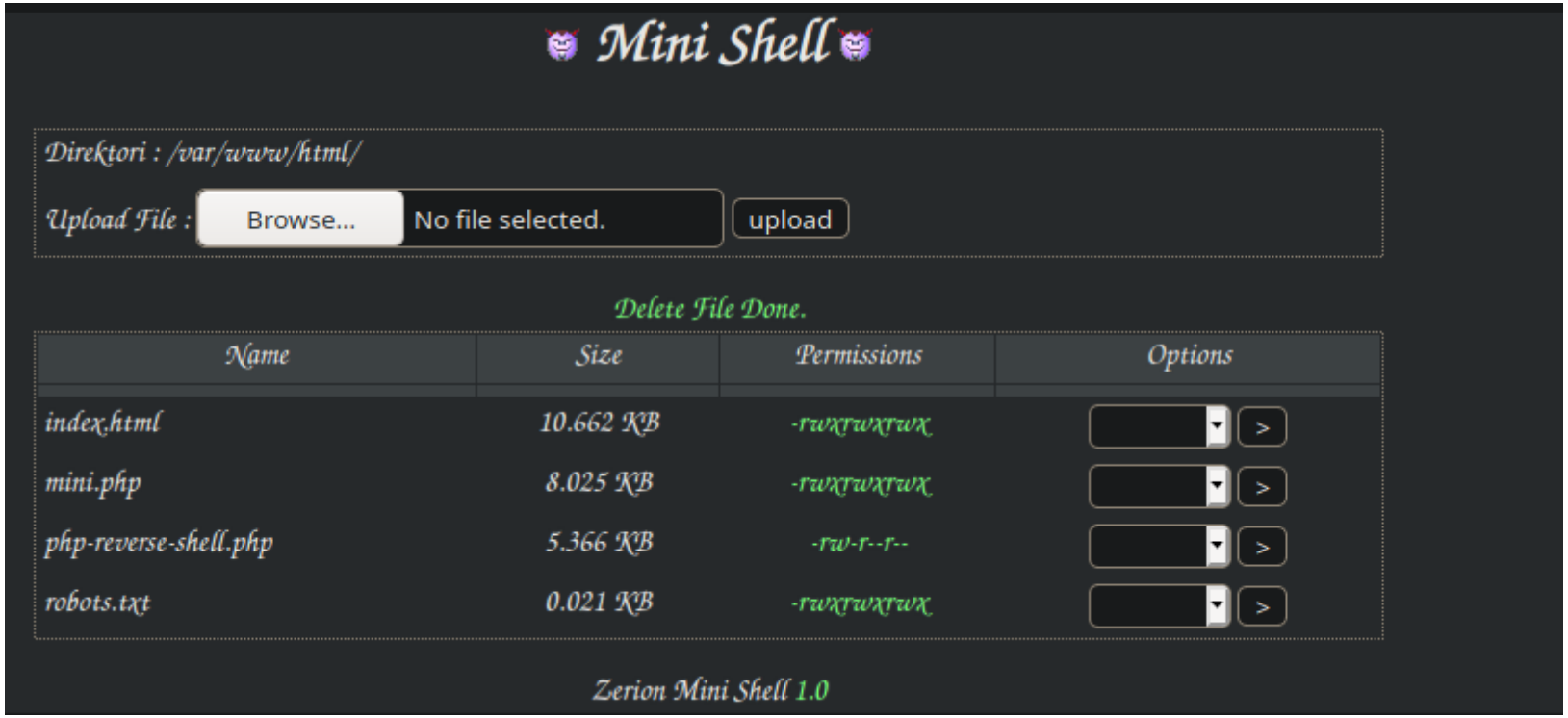


Just a simple phpmyadmin console login page, tried some default credentials like admin:admin, admin:password, admin123:123456, and few more. Also tried with SQL injection to see if we can get any SQL error but nothing. I moved on keeping

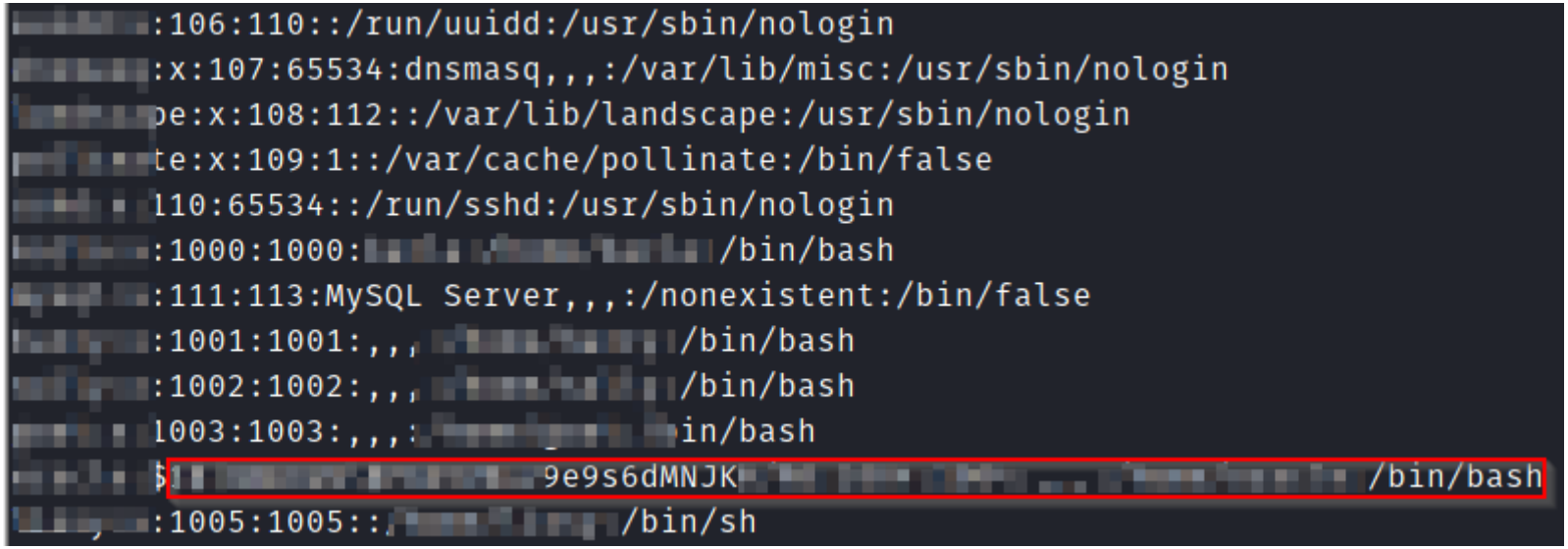
phpmyadmin console on standby.

Getting a reverse shell

Moving onto `/mini.php` looks like we have something here, we can view the contents hosted on the webserver in `/var/www/html` also we have the upload option. I tried uploading a php-reverse-shell and voila, I got my shell uploaded and all I need to do is start a listener and trigger the reverse shell.



Once we trigger the shell, we have a low-privileged reverse shell with us as `www-data` first thing I tried is viewing the contents of `/etc/passwd/` to check the users or if lucky look for any potential hash.



Privilege Escalation

And we got hash for one user we can try cracking that hash using john.

```
sudo /usr/sbin/john orcalehash --wordlist=/usr/share/wordlists/rockyou.txt
```

And we get his password, we already have reverse shell as `www-data` we can simply `su oracle` with his password, very first thing I tried is running `sudo -l` command to get his permissions but turns out that oracle user can run sudo.

After enumerating lot more, I came across `phpmyadmin` credentials, I still remember leaving that phpmyadmin console on standy but still we have a revershell as oracle now we can login to mysql to check if there exists something we can find useful.

```
mysql -u <username> -h localhost -p
```

I came across a Database, which might be useful for us to scrap some information regarding Privilege Escalation. But I found nothing interesting which could help us gain more privileges.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| phpmyadmin         |
+--------------------+
2 rows in set (0.00 sec)
```

So I randomly tried those phpmyadmin credentials I found to bruteforce for SSH. And I found karla credentials too with same credentials as phpmyadmin. So, now we have shell as karla. I tried running `sudo -l` to view the permissions.

```
karla@funbox7:/var/www/html$ sudo -l
[sudo] password for karla:
Matching Defaults entries for karla on funbox7:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User karla may run the following commands on funbox7:
    (ALL : ALL) ALL
```

So karla may run ALL commands on the machine. So all we need to do now is

```
karla@funbox7:/var/www/html$ sudo su
```

And we are **root** now!

```
karla@funbox7:/var/www/html$ sudo su
root@funbox7:/var/www/html#
```