Poison

Synopsis

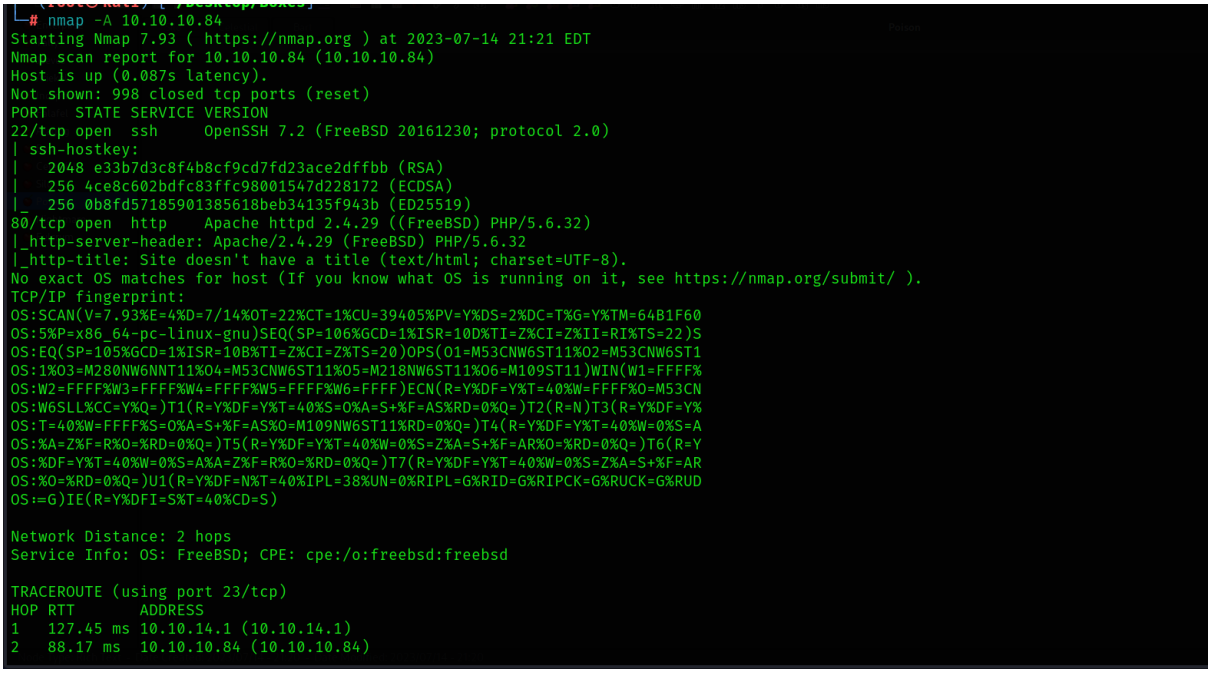
Poison focuses mainly on log poisoning and port forwarding/tunnelling. The machine is running FreeBSD which presents a few challenges for novice users as many common binaries from other distros are not available

Skills

- Knowledge of Linux
- Understanding of PHP local file inclusion
- Apache log poisoning
- Tunnelling ports over SSH

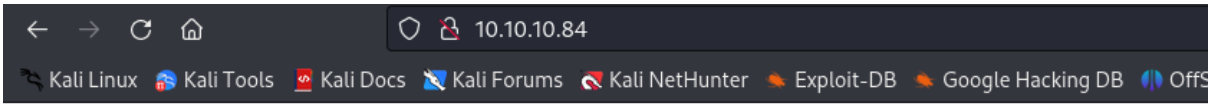
Exploitation

As always we start with the nmap to check what services/ports are open



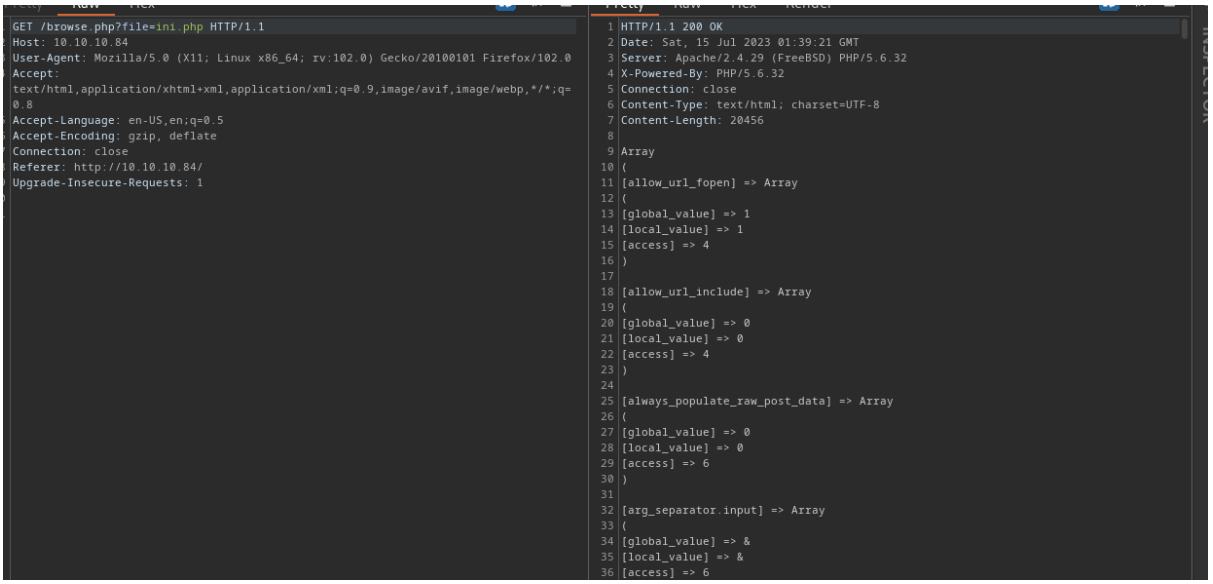
We can see only two ports open, let us then start from the web port cuz it has much broader attack surface

Opening the browser presents us with some basic page allowing us to view some of the PHP files



Temporary website to test local .php scripts.

Sites to be tested: ini.php, info.php, listfiles.php, phpinfo.php
Scriptname:
Submit

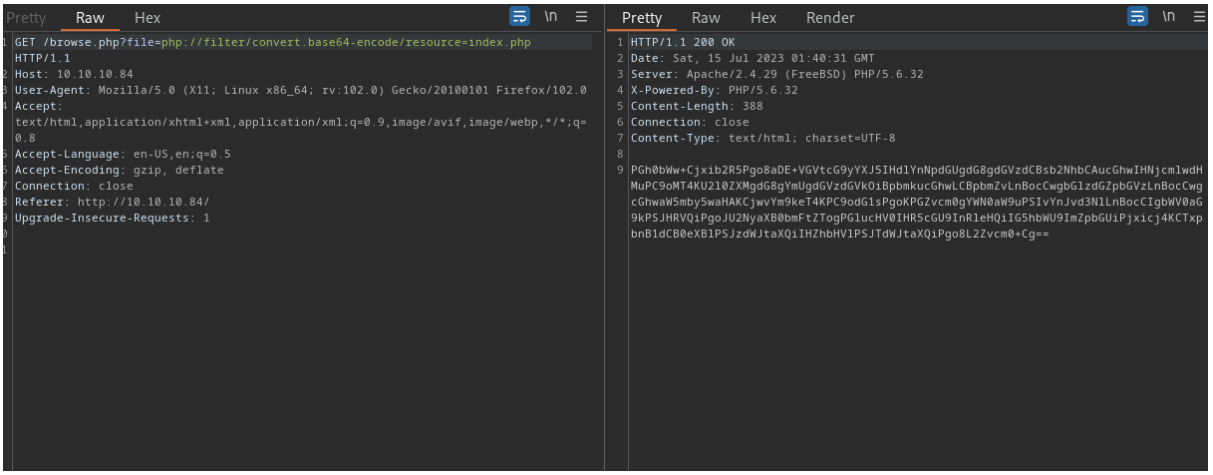


```
GET /browse.php?file=ini.php HTTP/1.1
Host: 10.10.10.84
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: http://10.10.10.84/
Upgrade-Insecure-Requests: 1
```

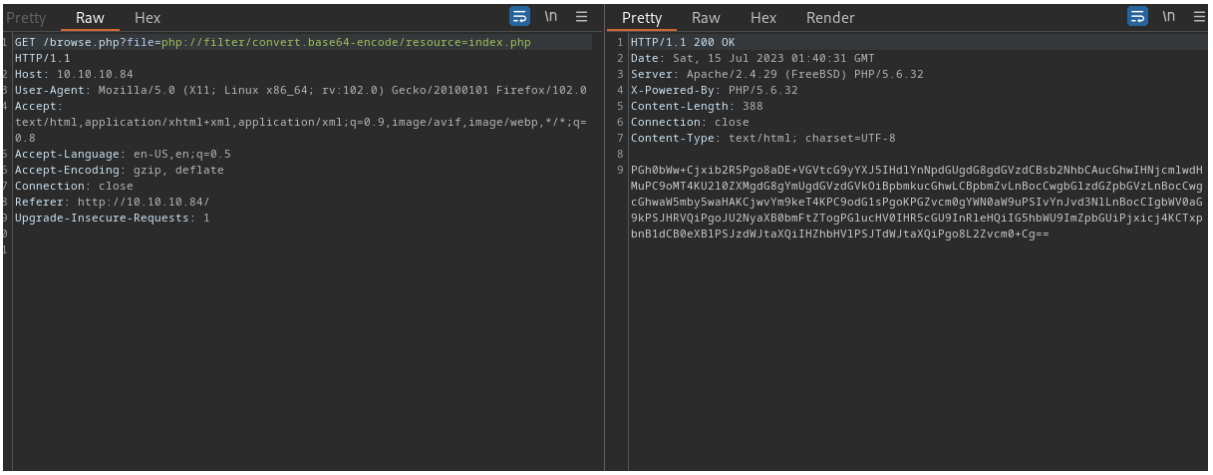
```
HTTP/1.1 200 OK
Date: Sat, 15 Jul 2023 01:39:21 GMT
Server: Apache/2.4.29 (FreeBSD) PHP/5.6.32
X-Powered-By: PHP/5.6.32
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Length: 20456

Array
(
[allow_url_fopen] => Array
(
[global_value] => 1
[local_value] => 1
[access] => 4
)

[allow_url_include] => Array
(
[global_value] => 0
[local_value] => 0
[access] => 4
)

[always_populate_raw_post_data] => Array
(
[global_value] => 0
[local_value] => 0
[access] => 6
)

[arg_separator.input] => Array
(
[global_value] => &
[local_value] => &
[access] => 6
```

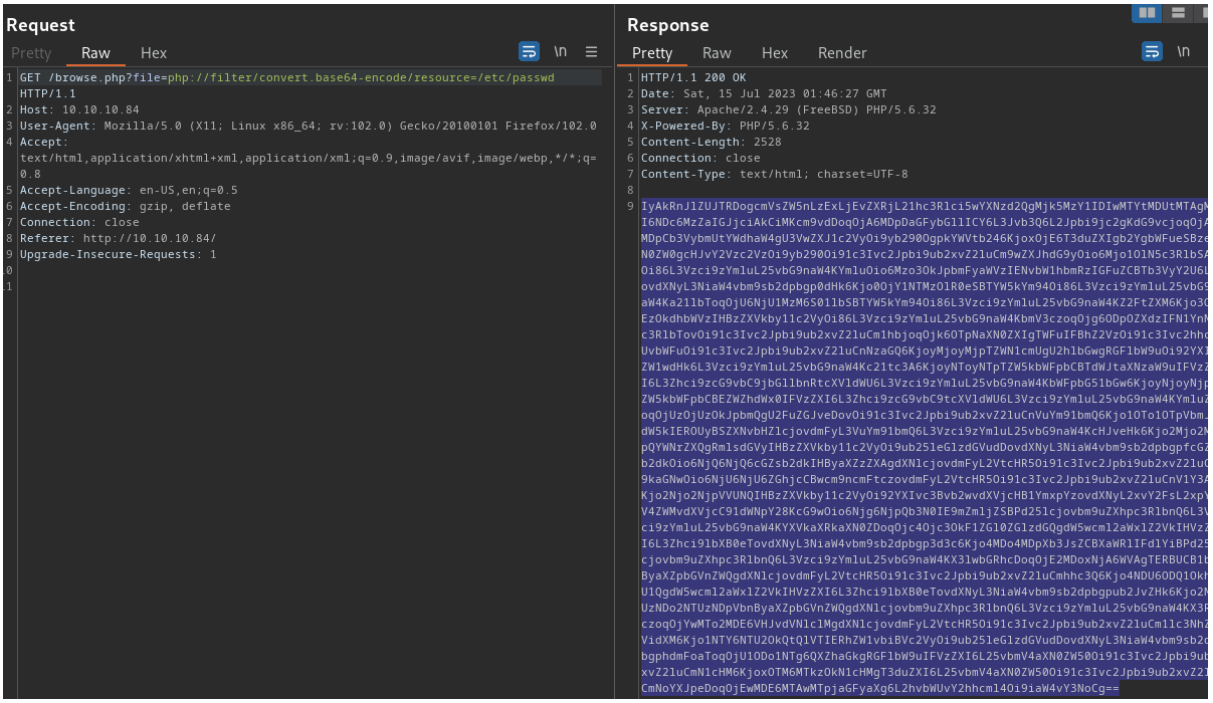
After inspecting how the application works, the conclusion is - it’s a perfect opportunity for PHP local file inclusion attack

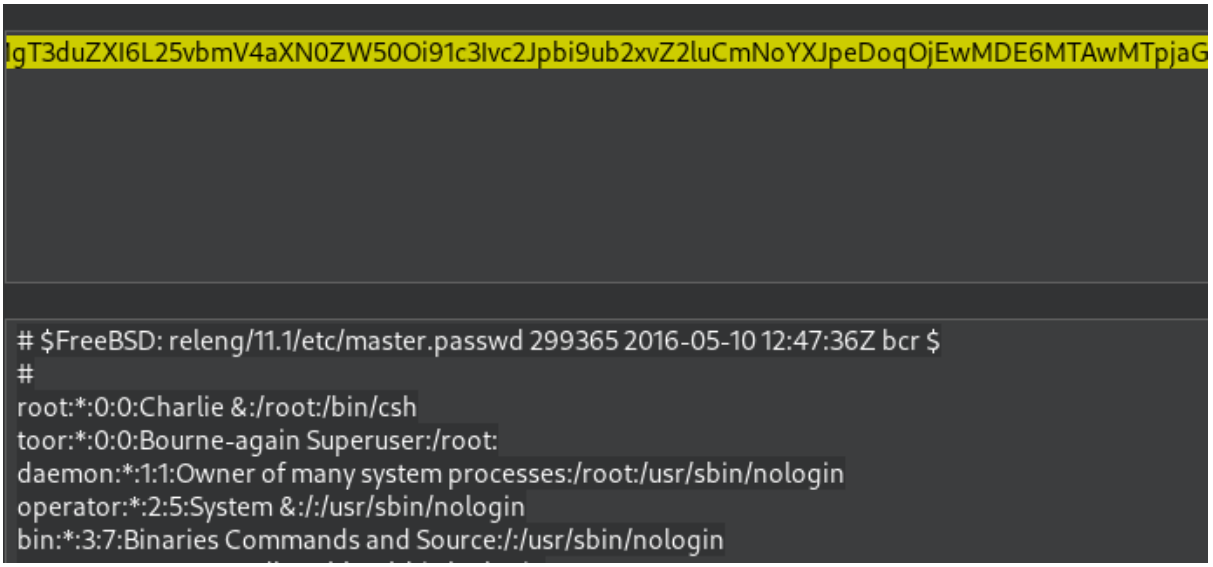
Let us verify if the application is vulnerable for that kind of attack



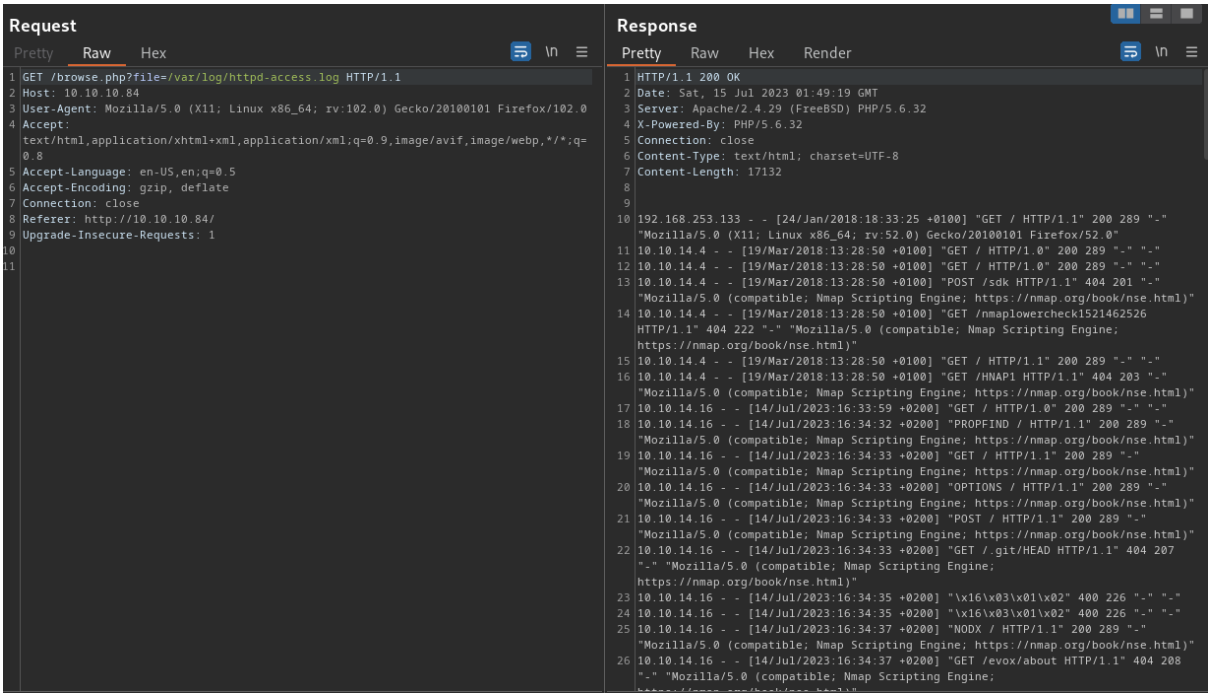
And we successfully performed PHP local file inclusion, what resulted in extracting a content of files that we shouldn’t be seeing

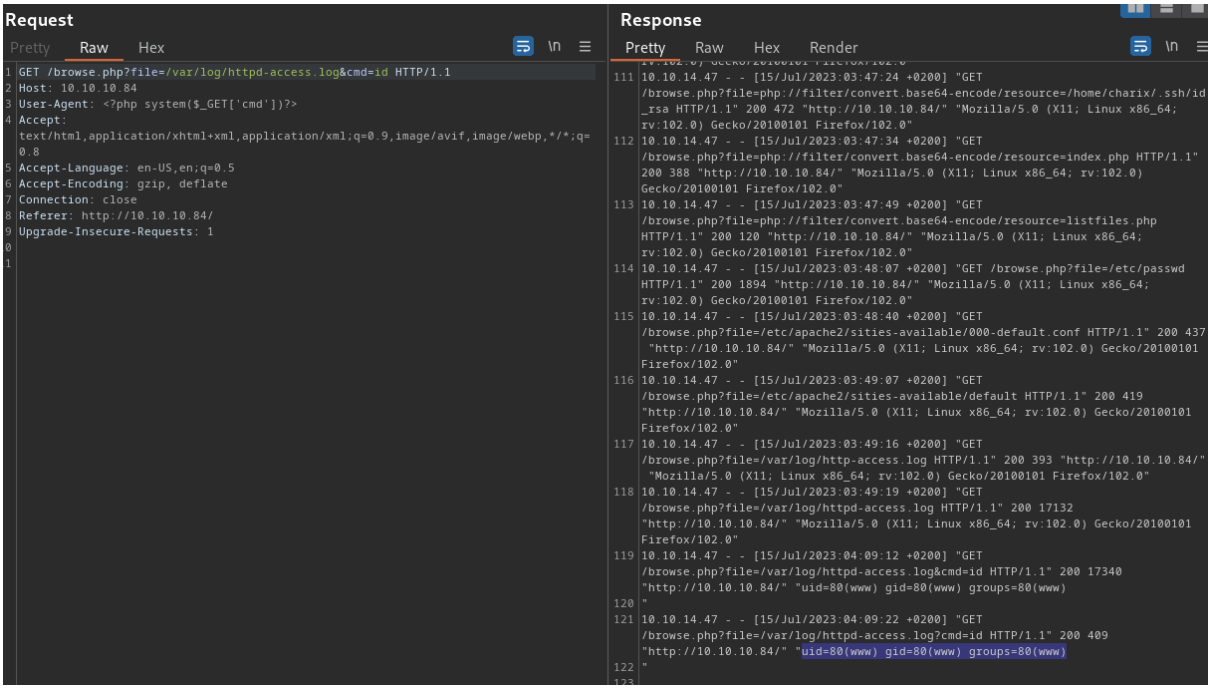




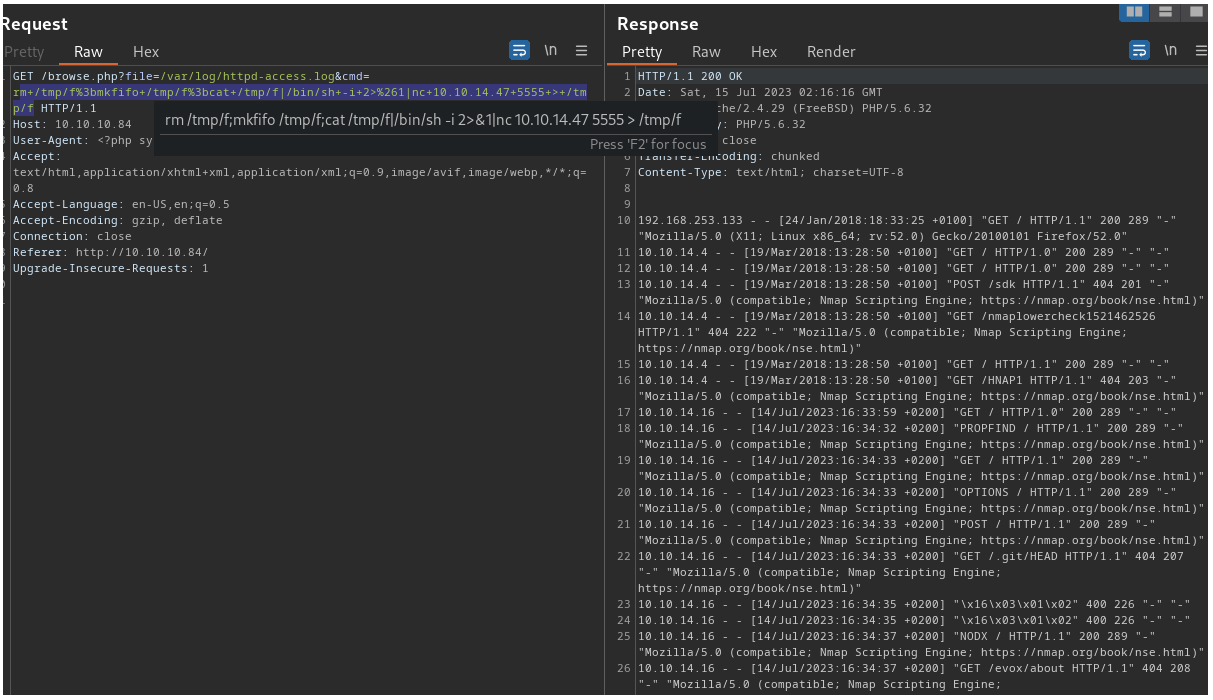


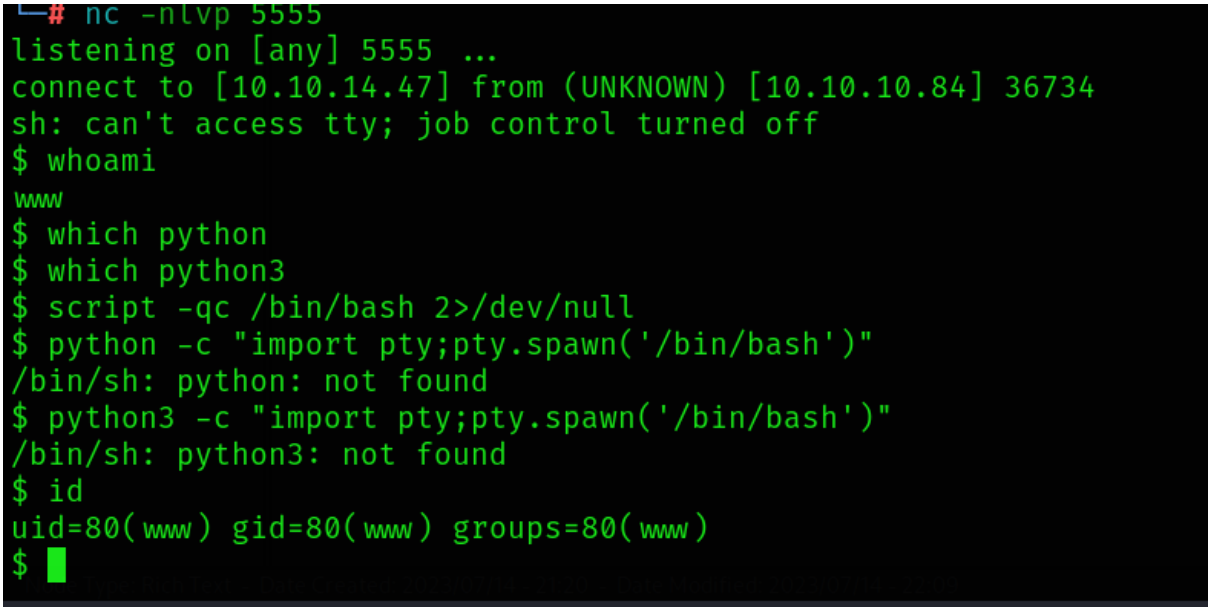
The LFI vulnerability was used not only to extract files from the server but also to get access to the apache web server logs and if we can access web server logs we can also poison them to get a remote code execution





After confirming the remote code execution, the next step is to get a reverse shell on the system





And we got an access to the system