Logotipo

Descrição gerada automaticamente com confiança baixa

Relatório de CTF

Mustacchio – TryHackMe

|  |  |
| --- | --- |
| **Informações do documento** | |
| **Referência** | Mustacchio – Gabriel F. Sabino |
| **N° Revisão** |  |
| **Data de publicação** | 29/09/2025 |
| **Link** | https://tryhackme.com/room/mustacchio |

|  |  |  |
| --- | --- | --- |
| **Redação** | Gabriel F. Sabino | Estudante |
| **Revisão** |  |  |
| **Aprovação** |  |  |

|  |  |  |
| --- | --- | --- |
| **Histórico de revisões** | | |
| **N°** | **Entregas** | **Descrição** |
| **0** | 29/09/2025 | Gabriel F. Sabino |
| **1** |  |  |
| **2** |  |  |

|  |  |
| --- | --- |
| **Informações do CTF** | |
| **Nível de Dificuldade** | Fácil |
| **Tipo de acesso** | Gratuito |
| **Conceitos envolvidos** | Exploração Web, Escalação de Privilégio, Arquivos Backup, XML e XXE, Path Hijack |
| **Plataforma** | Tryhackme |
| **Área** | Red |

**Sumário**

[Contextualização 3](#_heading=h.gjdgxs)

[Desenvolvimento 3](#_heading=h.1fob9te)

What is the user flag 3

What is the root flag 12

[Conclusão](#_heading=h.1t3h5sf) 13

[Referências](#_heading=h.4d34og8) 13

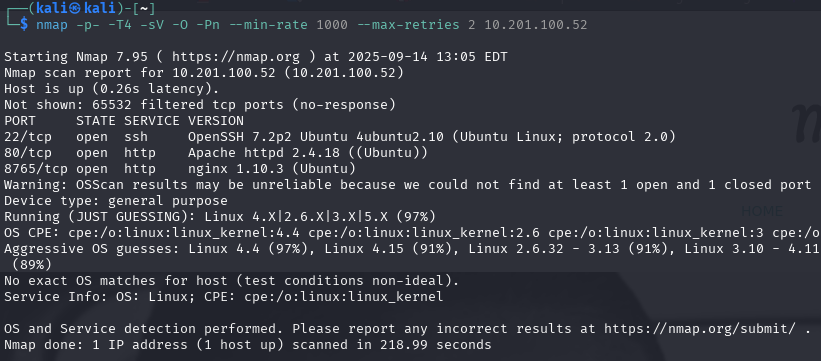
**Contextualização**

O CTF "Mustacchio" do TryHackMe é um desafio visando testar habilidades de webhacking, como exploração web e escalação de privilégio.

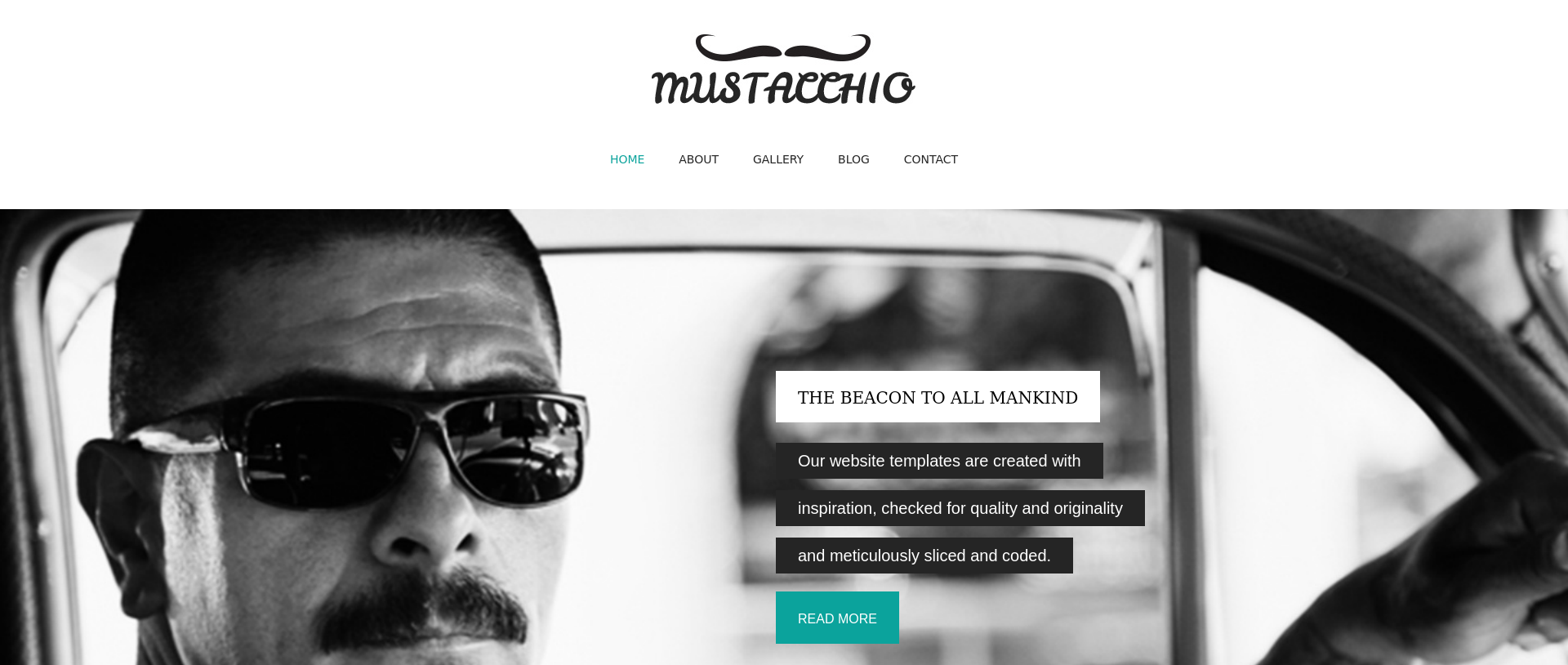
**Desenvolvimento**

**What is the USER flag?**

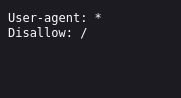
Iniciamos fazemos um scan usando nmap:



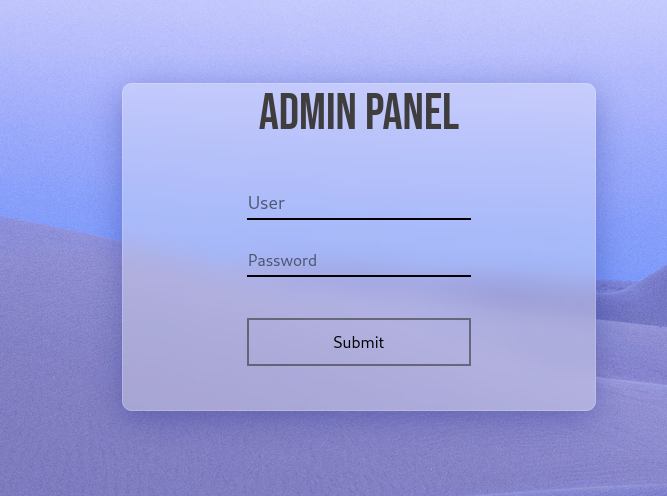
Observa-se que há 3 portas, vamos analisar a porta 80. Indo até ele, vemos um site relacionado a bigodes:



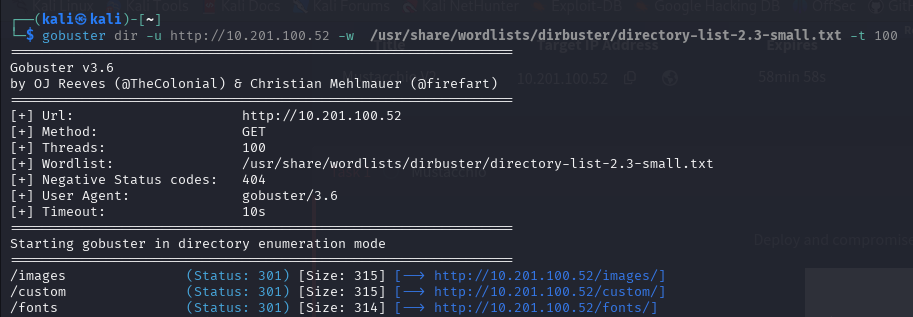
Analisando o site e o código fonte da página não encontramos nada. Tentando ir até /robots.txt também não gera resultados úteis:



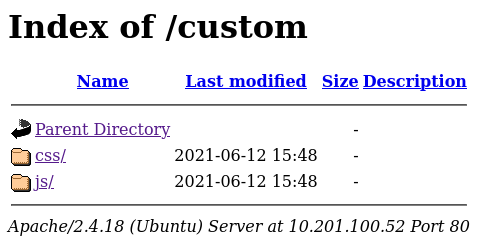
Vamos então analisar a porta 8765 então, nela somos levados até uma página de login para admin:



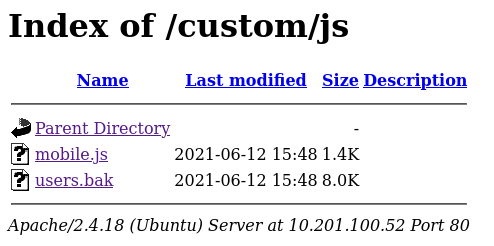
Como não temos muito o que fazer aqui no momento, vamos explorar os diretórios com o gobuster:



Tanto o /images quanto o /fonts não parecem ter nada demais, analisando o /custom temos:



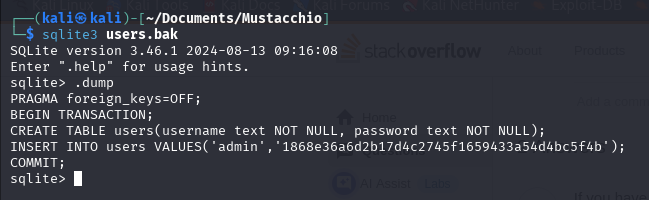
Na pasta js/ temos um arquivo que parece ser relevante, o users.bak:



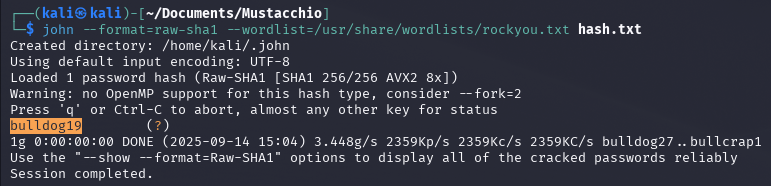
O formato .bak é um backup, para abrirmos ele, vamos precisar saber qual sua extensão original, então, baixando esse arquivo e usando o comando file, descobrimos que tipo de arquivo ele é:



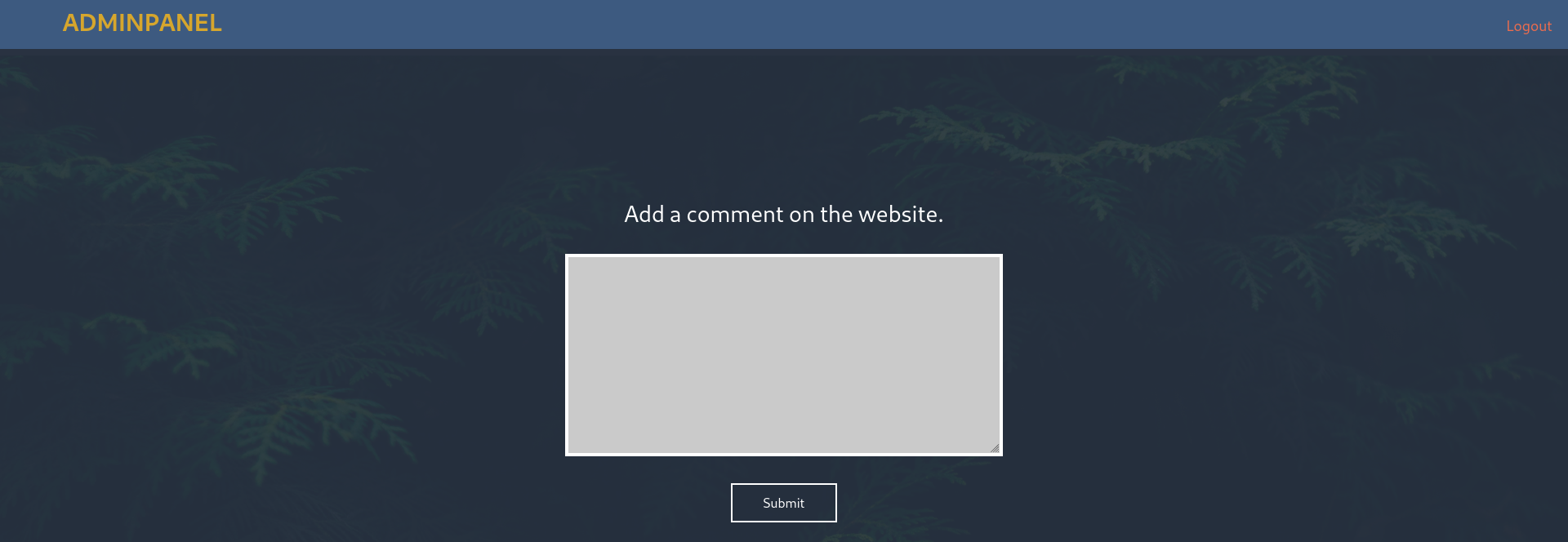
Ele era um SQLite, com isso vamos abrir ele e ver seu conteúdo:



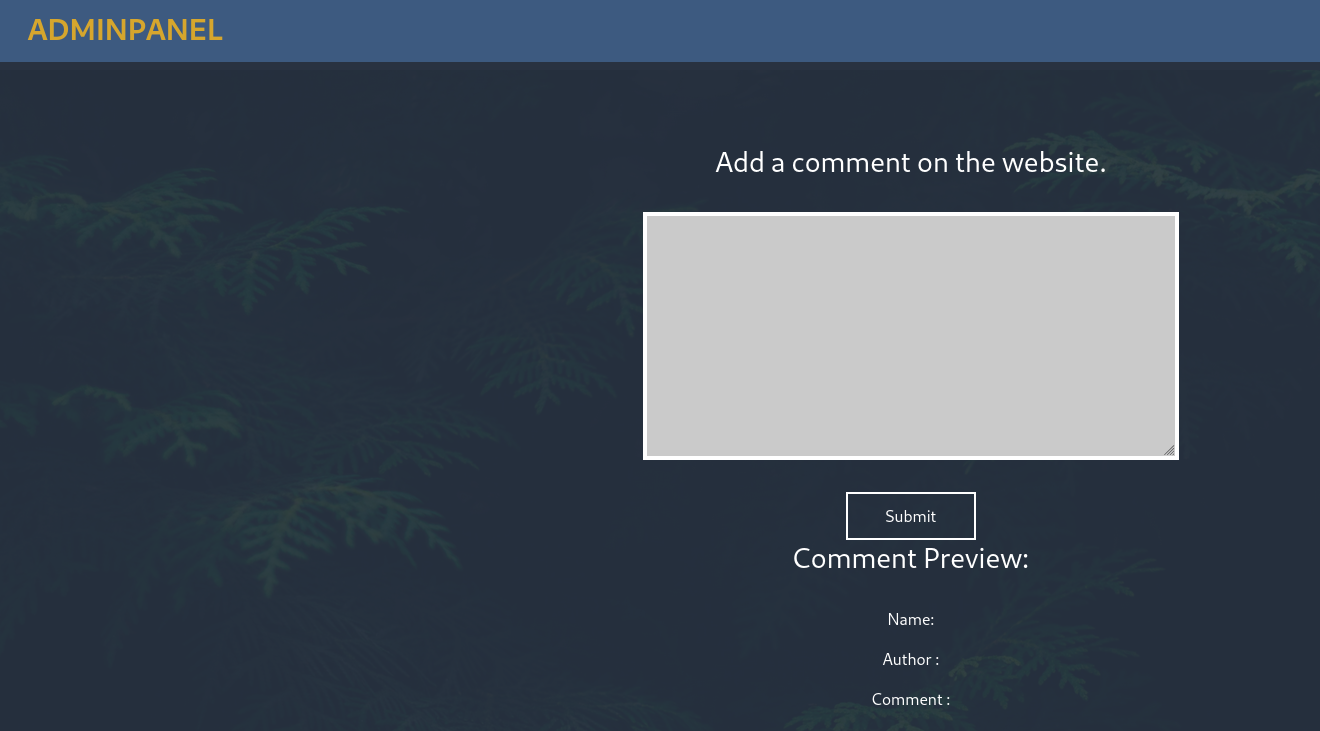
Conseguimos o login (admin) e senha (1868e36a6d2b17d4c2745f1659433a54d4bc5f4b), tentando entrar com esses dados, não conseguimos, talvez a senha esteja criptografada:



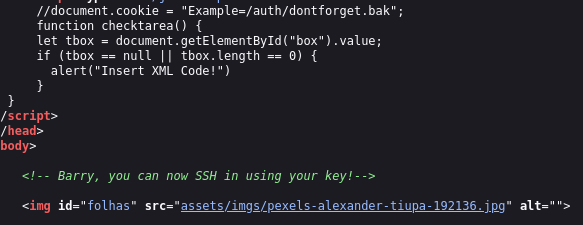
Tentando fazer login com essa senha agora conseguimos entrar e temos:



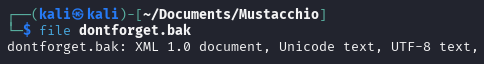
Ao darmos submit, a pagina troca para:

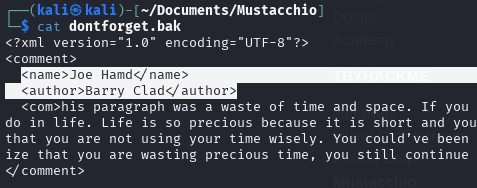


Analisando o código fonte da página temos:



Descobrimos 2 coisas, o login do ssh (Barry) e uma url/auth/dontforget.bak. Vamos ver o que tem nesse arquivo .bak:



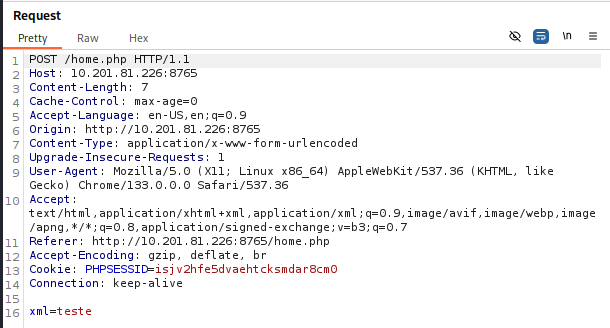


Ele contém um texto dizendo que estamos perdendo tempo ao ler ele, mas o nome e o autor podem vir a ser as informações importantes para entrar no ssh.

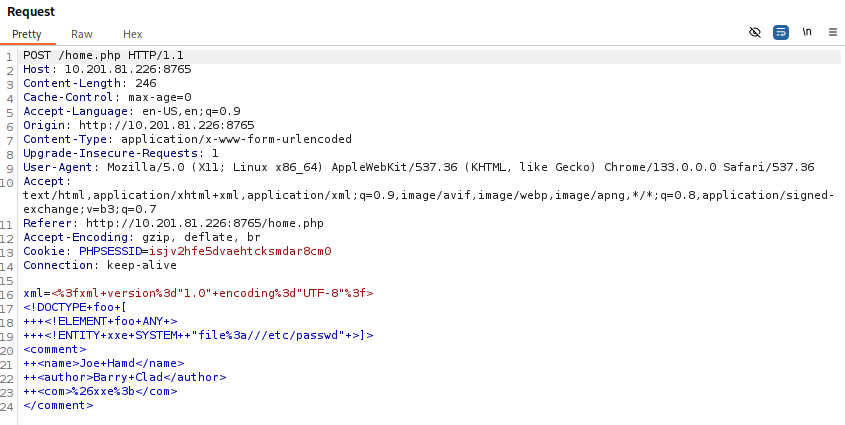
Após exaurir as possibilidades, vemos que ainda não temos a senha. Vamos, portanto, nos atentar ao fato do arquivo ser XML e no fato dele poder ser explorado via XXE.

XXE se baseia no fato de que algumas aplicações usam o formato XML para transmitir dados entre browser e servido, de forma a interceptar essa transmissão e poder alterar o que é transmitido.

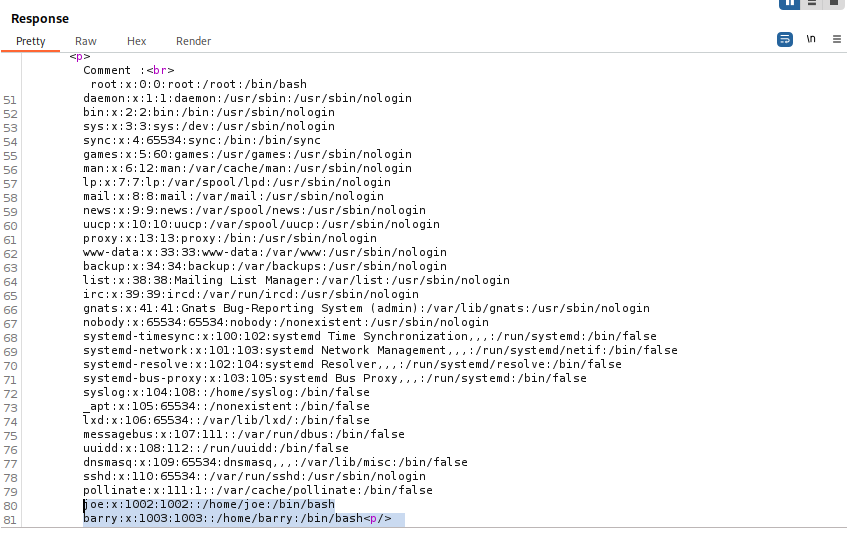
Vamos interceptar a requisição utilizando BurpSuite:



Alterando o conteúdo do ‘xml=’ :

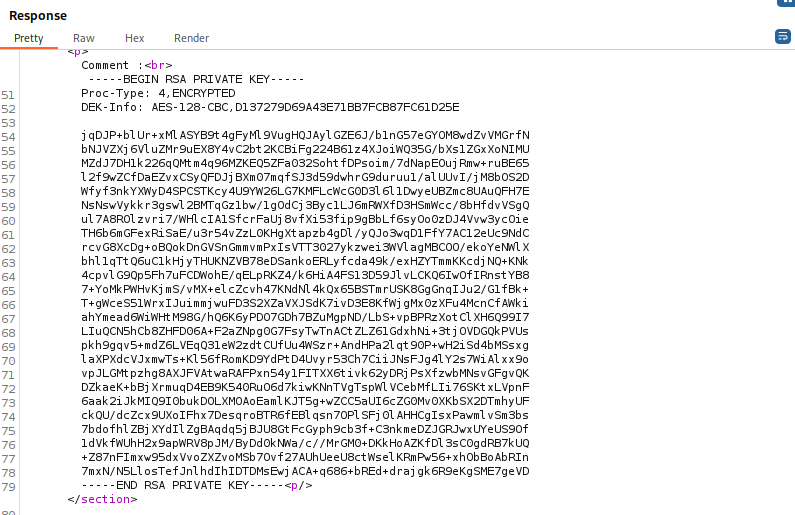


Teremos:

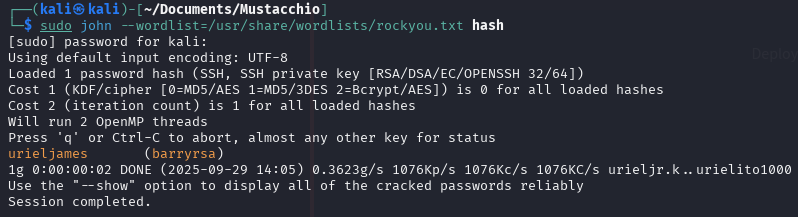


Nas últimas linhas vemos que existem informações relacionadas ao Barry, vamos acessá-las:

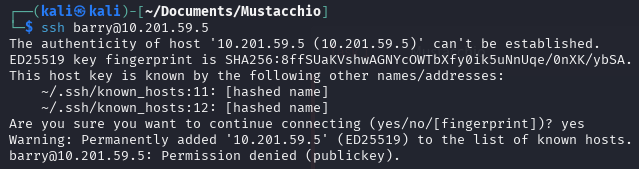




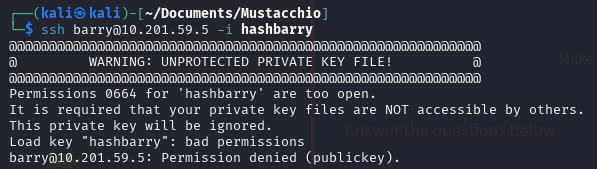
Utilizando o johnthereaper pra descriptografar a chave RSA, temos:



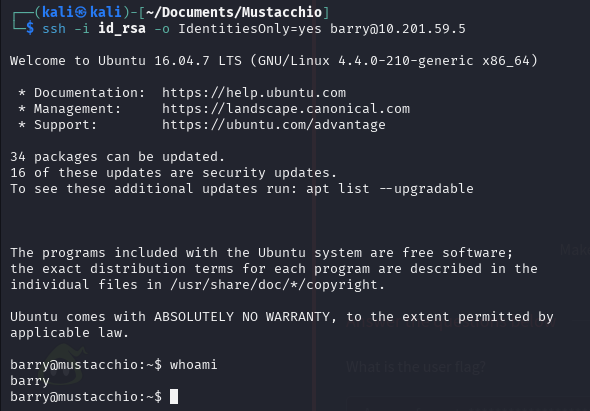
Com a senha, vamos fazer login no ssh:



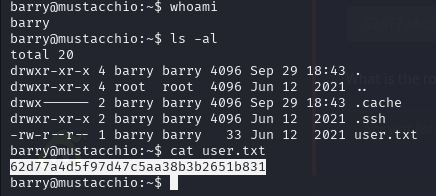
Algo deu errado, parece que precisamos identificar a senha:



Outra coisa deu errado, precisamos tornar a senha acessível usando chmod 600:



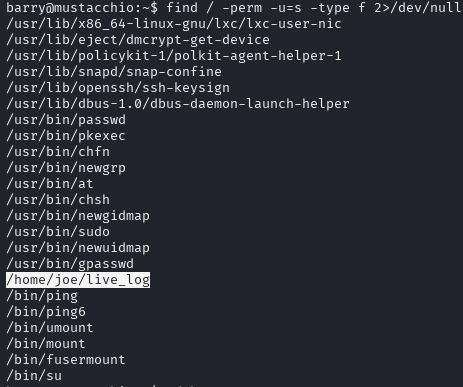
Conseguimos entrar, vamos ver o que temos:



Encontrando o arquivo user.txt parece promissor, abrindo ele, encontramos a user flag.

**What is the ROOT flag?**

Para acessarmos como root, vamos tentar por SUID primeiro, usando o comando find, vemos quais comandos temos permissão para usar:



A linha em destaque é suspeita pois Joe era o nome da outra pessoa, vamos analisar o que podemos fazer com esse comando.

Executar ele não mostra nada, mas rodando alguns testes, descobrimos que podemos fazer path hijack, isso é, não foi especificado o caminho que deveria ser seguido pelo comando, então podemos adulterar ele para o caminho que quisermos.

Criamos o caminho /tmp/tail de forma a procurar arquivos que possam ser a nossa flag e depois exibe o seu conteúdo, ou seja exibira a flag se a encontrar.

Vamos configurar esse caminho e rodar o comando utilizando o caminho que criamos para encontrar a flag:



Ao rodarmos tudo, encontramos a root flag.

**Conclusão**

Este CTF leva a pessoa a saber sobre XML e XXE, além de um pouco de path hijack. É uma boa porta de entrada para esses conteúdos, já que não se aprofunda muito no tema mas faz o usuário buscar um pouco sobre para conseguir entender como funciona e como se aproveitar dele.

**Referências**

<https://chatgpt.com/>

<https://portswigger.net/web-security/xxe>