

0. Introducción

El principal objetivo de esta práctica sería sacar información de un entorno preparado para ser atacado y saber que herramientas y cómo podemos explotar dicho entorno.

1. Reconocimiento

El primer paso sería averiguar la ip de nuestra máquina víctima,para ello, usamos el comando:

- nmap -sn 192.168.1.0/24(lo hago desde esta ip pq mi ip es 192.168.1.X y queremos ver todos los dispositivos conectados a la misma red):

```
(balbino@balbino)-[~]
$ nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-09 13:29 CEST
Nmap scan report for Livebox (192.168.1.1)
Host is up (0.0075s latency).
MAC Address: 8C:19:B5:FF:E2:1F (Arcadyan)
Nmap scan report for 192.168.1.10
Host is up (0.10s latency).
MAC Address: D8:BC:38:68:01:80 (Espressif)
Nmap scan report for 192.168.1.11
Host is up (0.026s latency).
MAC Address: C8:2E:18:80:A3:00 (Espressif)
Nmap scan report for 192.168.1.12
Host is up (0.025s latency).
MAC Address: C0:95:CF:1A:72:C2 (Unknown)
Nmap scan report for 192.168.1.13
Host is up (0.092s latency).
MAC Address: 2C:93:FB:7D:C8:00 (Sercomm France Sarl)
Nmap scan report for 192.168.1.14
Host is up (0.0015s latency).
MAC Address: 2C:08:23:D8:D9:F0 (Sercomm France Sarl)
Nmap scan report for 192.168.1.23
Host is up (0.19s latency).
MAC Address: 76:21:85:4C:3F:C8 (Unknown)
Nmap scan report for 192.168.1.24
Host is up (0.00023s latency).
MAC Address: 08:00:27:54:14:FB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.19
Host is up.
Nmap done: 256 IP addresses (9 hosts up) scanned in 7.10 seconds
```

Sabemos que es un máquina virtual pero no sabemos exactamente qué sistema operativo es,para ello usamos un simple ping a la máquina y mediante sus TTL(Time To Live):

```
(balbino@balbino)~[~]  
$ ping 192.168.1.24  
PING 192.168.1.24 (192.168.1.24) 56(84) bytes of data.  
64 bytes from 192.168.1.24: icmp_seq=1 ttl=64 time=0.551 ms  
64 bytes from 192.168.1.24: icmp_seq=2 ttl=64 time=0.293 ms  
64 bytes from 192.168.1.24: icmp_seq=3 ttl=64 time=0.312 ms  
64 bytes from 192.168.1.24: icmp_seq=4 ttl=64 time=0.372 ms  
^C  
--- 192.168.1.24 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3054ms  
rtt min/avg/max/mdev = 0.293/0.382/0.551/0.101 ms
```

Operating System	TCP	UDP	ICMP
Linux	64	64	255
FreeBSD	64	64	255
Mac OS X	64	64	255
Solaris	255	255	255
Windows	32	32	255
95/98/ME			
Windows XP,7,8, 2003, 2008	128	128	255

Y como podemos ver en la siguiente tabla como los TTL son de 64 posiblemente nos encontremos antes un Linux(Ubuntu server),FreeBSD o Mac OS X.

2. Escaneo de puertos

Una vez hemos hecho el escaneo de la red, y hemos detectado el IP de la máquina que está actualmente corriendo empezamos a analizar qué puertos activos tiene la máquina mediante el comando:

`-nmap -sV -p- 192.168.1.24(ip_victima):`

```
(balbino@balbino)-[~]
$ nmap -sV -p- 192.168.1.24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-09 14:12 CEST
Nmap scan report for 192.168.1.24
Host is up (0.00013s latency).
Not shown: 65506 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
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) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
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)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbb)
35195/tcp open  mountd       1-3 (RPC #100005)
39089/tcp open  java-rmi     GNU Classpath grmiregistry
39242/tcp open  nlockmgr     1-4 (RPC #100021)
46839/tcp open  status       1 (RPC #100024)
MAC Address: 08:00:27:54:14:FB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; 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 128.17 seconds
```

Y ahora con el comando que acabamos de ejecutar podemos ver como se trata de una máquina Linux, lo podemos saber gracias a los puertos que están abiertos como por ejemplo el 22 que sirve para una conexión por SSH

3. Preparación del entorno

Si queremos obtener más información o si se está lanzando algún servicio, usamos:

- nslookup (ip_victima):

A terminal window with a dark background. The prompt is '(balbino@balbino)-[~]'. The user enters '\$ nslookup 192.168.1.24'. The output is '** server can't find 24.1.168.192.in-addr.arpa: NXDOMAIN'.

```
(balbino@balbino)-[~]  
$ nslookup 192.168.1.24  
** server can't find 24.1.168.192.in-addr.arpa: NXDOMAIN
```

Como podemos ver, no existe ningún servicio lanzado por lo que podríamos pasar directamente a la fase de la búsqueda de vulnerabilidades.

4. Detección de vulnerabilidades

El siguiente paso sería comprobar mediante un nikto -h que posibles vías de ataque existen:

```
└─$ nikto -h 192.168.1.24
- Nikto v2.5.0

-----
+ Target IP:      192.168.1.24
+ Target Hostname: 192.168.1.24
+ Target Port:    80
+ Start Time:     2025-05-09 18:19:17 (GMT2)
-----

+ Server: Apache/2.2.8 (Ubuntu) DAV/2
+ /: Retrieved x-powered-by header: PHP/5.2.4-2ubuntu5.10.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.php. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15, https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /: HTTP TRACE method is active which suggests the host is vulnerable to XST. See: https://owasp.org/www-community/attacks/Cross\_Site\_Tracing
+ /phpinfo.php: Output from the phpinfo() function was found.
+ /doc/: Directory indexing found.
+ /doc/: The /doc/ directory is browsable. This may be /usr/doc. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0678
+ /?=PHPB8B5F2A0-3C92-11d3-A3A9-4C7B08C10000: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F36-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F34-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /?=PHPE9568F35-D428-11d2-A769-00AA001ACF42: PHP reveals potentially sensitive information via certain HTTP requests that contain specific QUERY strings. See: OSVDB-12184
+ /phpMyAdmin/changelog.php: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/ChangeLog: Server may leak inodes via ETags, header found with file /phpMyAdmin/ChangeLog, inode: 92462, size: 40540, mtime: Tue Dec 9 18:24:00 2008. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /phpMyAdmin/ChangeLog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /test/: Directory indexing found.
+ /test/: This might be interesting.
+ /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information. See: CWE-552
+ /icons/: Directory indexing found.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /phpMyAdmin/: phpMyAdmin directory found.
+ /phpMyAdmin/Documentation.html: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ /phpMyAdmin/README: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts. See: https://typo3.org/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
```

Gracias a este comando podemos ver posibles vulnerabilidades que existen, como podemos ver tiene una vulnerabilidad con la versión de apache 2.2.8 las cuales son:

1. Ruta vulnerable /phpinfo.php:

CWE:-552 Accesible a directorio y archivos

CAPEC-137:inyección de parámetros.

CVSS:7.5 tiene una nota media-alta, ya que se puede llegar a sacar información sensible

EPSS:0.916

Posible solución:eliminar el archivo de esa ruta o restringir su acceso.

2.Http trace habilitado:

CWE-201:información expuesta.

CAPEC-111:Http response

CVSS:6.1

EPSS:0.723

Posible solución: deshabilitar el método TRACE en la configuración del servicio apache,para evitar XSS.

3. Falta de cabecera en X-Frame-Options:esto hace que sea vulnerable a clickjacking

CWE: 693 – Protection Mechanism Failure

CAPEC: 103 – Clickjacking

CVSS: 4.3

EPSS: 0.455

Posible solución: Añadir encabezados HTTP de seguridad como X-Frame-Options: DENY.

5. Fuerza Bruta y enumeración de archivos/directorios

Una vez sabemos esto,empezamos a probar diccionarios como el de SecList para hacer un ataque de fuerza bruta,con gobuster y dirb:

```
(balbino@balbino)~[~]
$ gobuster dir -u http://192.168.1.24 -w /home/balbino/SecLists/Discovery/Web-Content/common.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.1.24
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /home/balbino/SecLists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/.htaccess (Status: 403) [Size: 294]
/.hta (Status: 403) [Size: 289]
/.htpasswd (Status: 403) [Size: 294]
/cgi-bin/ (Status: 403) [Size: 293]
/dav (Status: 301) [Size: 315] [--> http://192.168.1.24/dav/]
/index (Status: 200) [Size: 891]
/index.php (Status: 200) [Size: 891]
/phpMyAdmin (Status: 301) [Size: 322] [--> http://192.168.1.24/phpMyAdmin/]
/phpinfo (Status: 200) [Size: 47978]
/server-status (Status: 403) [Size: 298]
/phpinfo.php (Status: 200) [Size: 47990]
/test (Status: 301) [Size: 316] [--> http://192.168.1.24/test/]
/twiki (Status: 301) [Size: 317] [--> http://192.168.1.24/twiki/]
Progress: 4746 / 4747 (99.98%)
=====
Finished
=====
```

```
-$ dirb http://192.168.1.24 /home/balbino/SecLists/Discovery/Web-Content/common.txt

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Fri May 9 19:59:18 2025
URL_BASE: http://192.168.1.24/
WORDLIST_FILES: /home/balbino/SecLists/Discovery/Web-Content/common.txt

-----

GENERATED WORDS: 4745

---- Scanning URL: http://192.168.1.24/ ----
+ http://192.168.1.24/cgi-bin/ (CODE:403|SIZE:293)
==> DIRECTORY: http://192.168.1.24/dav/
+ http://192.168.1.24/index (CODE:200|SIZE:891)
+ http://192.168.1.24/index.php (CODE:200|SIZE:891)
==> DIRECTORY: http://192.168.1.24/phpMyAdmin/
+ http://192.168.1.24/phpinfo (CODE:200|SIZE:48062)
+ http://192.168.1.24/phpinfo.php (CODE:200|SIZE:48074)
+ http://192.168.1.24/server-status (CODE:403|SIZE:298)
==> DIRECTORY: http://192.168.1.24/test/
==> DIRECTORY: http://192.168.1.24/twiki/

---- Entering directory: http://192.168.1.24/dav/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.24/phpMyAdmin/ ----
+ http://192.168.1.24/phpMyAdmin/ChangeLog (CODE:200|SIZE:40540)
+ http://192.168.1.24/phpMyAdmin/LICENSE (CODE:200|SIZE:18011)
+ http://192.168.1.24/phpMyAdmin/README (CODE:200|SIZE:2624)
+ http://192.168.1.24/phpMyAdmin/TODO (CODE:200|SIZE:235)
+ http://192.168.1.24/phpMyAdmin/calendar (CODE:200|SIZE:4145)
+ http://192.168.1.24/phpMyAdmin/changelog (CODE:200|SIZE:74593)
==> DIRECTORY: http://192.168.1.24/phpMyAdmin/contrib/
+ http://192.168.1.24/phpMyAdmin/docs (CODE:200|SIZE:4583)
+ http://192.168.1.24/phpMyAdmin/error (CODE:200|SIZE:1063)
+ http://192.168.1.24/phpMyAdmin/export (CODE:200|SIZE:4145)
+ http://192.168.1.24/phpMyAdmin/favicon.ico (CODE:200|SIZE:18902)
+ http://192.168.1.24/phpMyAdmin/import (CODE:200|SIZE:4145)
+ http://192.168.1.24/phpMyAdmin/index (CODE:200|SIZE:4145)
+ http://192.168.1.24/phpMyAdmin/index.php (CODE:200|SIZE:4145)
==> DIRECTORY: http://192.168.1.24/phpMyAdmin/js/
==> DIRECTORY: http://192.168.1.24/phpMyAdmin/lang/
==> DIRECTORY: http://192.168.1.24/phpMyAdmin/libraries/
+ http://192.168.1.24/phpMyAdmin/license (CODE:200|SIZE:18011)
+ http://192.168.1.24/phpMyAdmin/main (CODE:200|SIZE:4227)
+ http://192.168.1.24/phpMyAdmin/navigation (CODE:200|SIZE:4145)
+ http://192.168.1.24/phpMyAdmin/phpinfo (CODE:200|SIZE:0)
+ http://192.168.1.24/phpMyAdmin/phpinfo.php (CODE:200|SIZE:0)
```

Con el primer comando que he usado, gobuster, lo que obtenemos son códigos (200, 403), estos nos indican si el archivo es público, es decir un 200 o si el archivo está oculto o prohibido como un 403 o sino existe como sería un 404.

Y con el comando dirb, podemos sacar los directorios accesibles, como serían phpMyAdmin/, phpinfo.php, /phpinfo:

CVE: CVE-2019-12922 (ejecución remota de comandos en phpMyAdmin)

CWE: CWE-284 – Improper Access Control

CAPEC: CAPEC-137 – Parameter Injection

CVSS v3: 8.8 (Alta)

Posible solución:

Restringir el acceso a /phpMyAdmin/ solo desde IPs autorizadas, actualizar phpMyAdmin a la última versión, eliminar archivos innecesarios como README, LICENSE, etc. que puedan contener información sensible.

6. Herramientas utilizadas

Las herramientas que potencialmente hemos usado son:

- nmap: Nos ha servido tanto para averiguar cuál era la ip de nuestra máquina víctima como para sacar los puertos que tenía abiertos.
- nslookup: ambos los hemos usado para saber y obtener más información de que era a lo que queremos atacar, su sistema operativo, su mac etc...
- nikto: con él hemos podido ver las posibles vulnerabilidades existentes en la máquina víctima.
- gobuster, dirb: ambos nos han servido para poder a qué ficheros o archivos podríamos llegar a acceder mediante fuerza bruta.

7. Conclusiones

Gracias a esta práctica he podido asentar de mejor manera la base para saber como tengo que operar para ver las vulnerabilidades y vías de posible ataque. Y sobre todo poco a poco aprendiendo más sobre los estándares de ciberseguridad.