MÁSTER INTERUNIVERSITARIO EN CIBERSEGURIDAD

Práctica 1: Descubrimiento y enumeración

Maria Andrea Ugarte Valencia

1. Realiza un barrido ping (ICMP) sobre las máquinas proporcionadas. Captura con wireshark e identifica los cuatro paquetes esenciales del barrido.

Para realizar el barrido ping ejecutaremos el comando **nmap 192.168.56.1-254 -disable-arp-ping**. Nmap nos permite escanear direcciones IP y puertos en una red. Con la opción **-disable-arp-ping** se habilitarán los ping.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 192.168.56.1-254 -disable-arp-ping
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-26 07:48 EST
Nmap scan report for 192.168.56.6
Host is up (0.0018s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49154/tcp open  unknown
49155/tcp open  unknown
MAC Address: 08:00:27:E9:6E:CE (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.9
Host is up (0.00062s latency).
Not shown: 977 closed tcp ports (reset)
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
23/tcp  open  telnet
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
512/tcp open  exec
513/tcp open  login
514/tcp open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:CC:1A:AF (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.100
Host is up (0.0000050s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 254 IP addresses (3 hosts up) scanned in 58.03 seconds
```

Figura 1: Escaneo de la red 192.168.56.0/24

Gracias a este comando, ahora sabemos que las ip de las máquinas victima son: **192.168.56.6** y **192.168.56.9**. Ahora, identificaremos los paquetes esenciales del barrido:

a) **Paquetes TCP**: Al realizarse un barrido ping, por defecto se envían paquetes TCP por los puertos 80 y 443. En mi caso, seguramente debido a algún problema de configuración, solo aparecen los paquetes TCP correctamente cuando ejecuto el comando sin sudo.

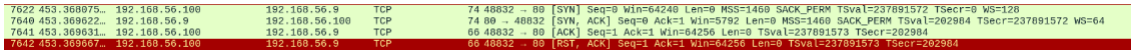

Figura 2: Paquetes TCP intercambiados con 192.168.56.9

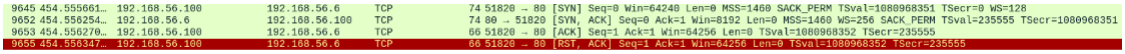

Figura 3: Paquetes TCP intercambiados con 192.168.56.6

b) **Paquetes ICMP**: Al realizar un barrido ping, identificamos las direcciones IP activas, ya que estas son las únicas que responden a los pings que se envían. Nmap envía estos pings a las máquinas que le contestan a las solicitudes ARP. A mí solo me aparecen estos paquetes cuando ejecuto el comando con sudo.

Figura 4: Paquetes ICMP intercambiados con 192.168.56.9 y 192.168.56.6

c) Paquetes ICMP Timestamp: También podríamos recibir un paquete ICMP Timestamp en lugar de un Echo Reply de alguna de las máquinas activas. En mi caso, estos paquetes no aparecen.

2. Llevar a cabo un escaneo sigiloso (Stealth) de toda la red virtualizada. Comprobar el tráfico producido con Wireshark.

Para llevar a cabo un escaneo sigiloso de toda la red usaremos el comando **sudo nmap -sS 192.168.56.1-254 -T2**. La opción **-sS** es la que nos permite realizar un escaneo sigiloso utilizando paquetes TCP SYN para establecer conexiones sin completarlas, lo que minimiza el riesgo de dejar rastro. Con **-T2** minimizamos aún más la posibilidad de ser detectados ya que se reduce la velocidad del escaneo.

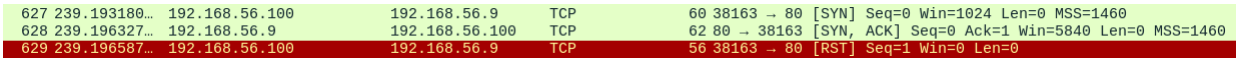

Figura 5: Conexión incompleta con 192.168.56.9

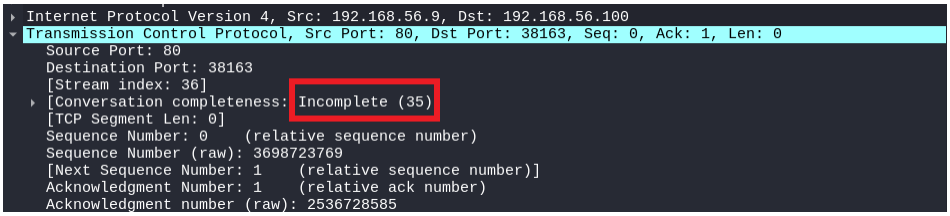

Figura 6: Conexión incompleta con 192.168.56.9

Figura 7: Conexión incompleta con 192.168.56.6

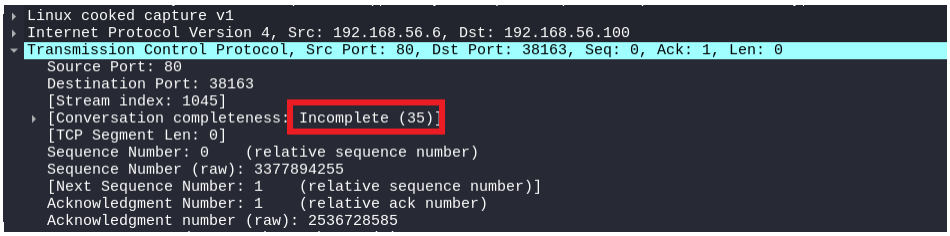

Figura 8: Conexión incompleta con 192.168.56.6

3. Realizar un escaneo agresivo sobre una máquina de internet (por ejemplo http://scanme.nmap.org) y sobre alguna de la red virtualizada. Ayudarse de otras herramientas como Wireshark o la opción –packet-trace de Nmap para comprobar similitudes y diferencias.

Al hacer un escaneo agresivo se envía una gran cantidad de paquetes de datos en poco tiempo para así recopilar información. Este tipo de escaneo es mucho más detectable que los tipos anteriores. Para realizar el escaneo agresivo se ha usado la opción **-A** de nmap. Realizamos este escaneo para la máquina de internet y para una de la red virtualizada:

Figura 9: Escaneo agresivo a la máquina 192.168.56.6

Figura 10: Escaneo agresivo a la máquina http://scanme.nmap.org)

Una diferencia que pude observar es que el escaneo a la máquina de la red virtualizada tarda menos que el de la máquina de internet. Esto seguramente se deba a que el número de paquetes de la máquina de la red virtualizada es menor que el de la máquina de internet.

Figura 11: Paquetes de la máquina de la red virtualizada

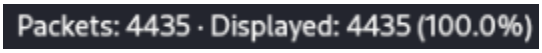

Figura 12: Paquetes de la máquina de internet

Otra obvia diferencia es que la información recopilada de cada máquina es diferente.

```
└─$ sudo nmap 192.168.56.6 -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 10:39 EST
Nmap scan report for 192.168.56.6
Host is up (0.0012s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http         Microsoft IIS httpd 7.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Site doesn't have a title.
|_http-server-header: Microsoft-IIS/7.5
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows Server 2008 R2 Enterprise 7601 Service Pack 1 microsoft-ds (workgroup: MUNICS)
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge|general purpose|switch
Running (JUST GUESSING): Oracle Virtualbox (96%), QEMU (91%), Bay Networks embedded (86%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu cpe:/h:baynetworks:baystack_450
Aggressive OS guesses: Oracle Virtualbox (96%), QEMU user mode network gateway (91%), Bay Networks BayStack 450 switch (software version 3.1.0.22) (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Host: META-FLAVOUR2; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 11h39m58s, deviation: 4h37m07s, median: 8h59m58s
| smb2-time:
|   date: 2024-02-28T00:40:17
|_  start_date: 2024-02-28T00:34:26
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_nbstat: NetBIOS name: META-FLAVOUR2, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:e9:6e:ce (Oracle VirtualBox virtual NIC)
| smb-os-discovery:
|   OS: Windows Server 2008 R2 Enterprise 7601 Service Pack 1 (Windows Server 2008 R2 Enterprise 6.1)
|   OS CPE: cpe:/o:microsoft:windows_server_2008::sp1
|   Computer name: Meta-Flavour2
|   NetBIOS computer name: META-FLAVOUR2\x00
|   Workgroup: MUNICS\x00
|_  System time: 2024-02-27T16:40:17-08:00

TRACEROUTE (using port 139/tcp)
HOP RTT     ADDRESS
1   0.31 ms 10.0.2.2
2   1.36 ms 192.168.56.6

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

Figura 13: Escaneo agresivo a la máquina 192.168.56.6

```
┌──(kali㉿kali)-[~]
└─$ nmap scanme.nmap.org -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-27 19:49 EST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.21s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
|_http-title: Go ahead and ScanMe!
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-favicon: Nmap Project
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: 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 26.36 seconds
```

Figura 14: Escaneo agresivo a la máquina http://scanme.nmap.org

Por el resto, el escaneo es prácticamente igual.

4. Realizar un escaneo lo más rápido posible (insane) sobre las máquinas disponibles. Comprobar además la versión de los servicios implementados. Buscar al menos una vulnerabilidad en https://cve.mitre.org/ para cada uno de esos servicios.

Para realizar un escaneo lo más rápido posible comprobando la versión de los servicios ejecutamos el comando **nmap -sV 192.168.56.X -T5**. Gracias a la opción **-T5** aumentamos la velocidad del escaneo. Con **-sV** obtenemos la versión de los servicios. Lo hacemos para las dos máquinas disponibles:

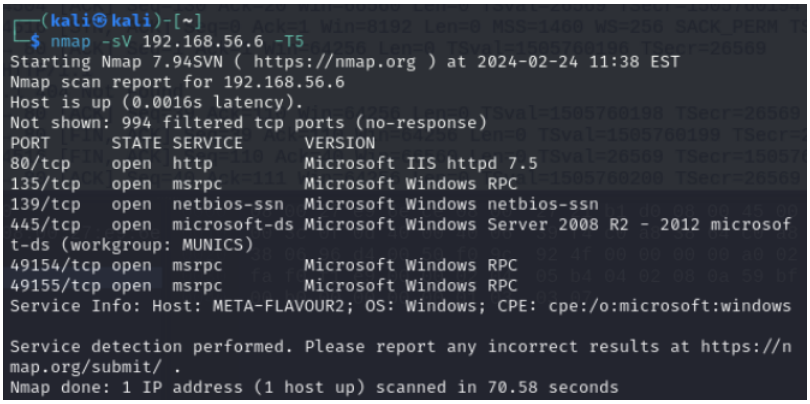

Figura 15: escaneo a la máquina 192.168.56.6

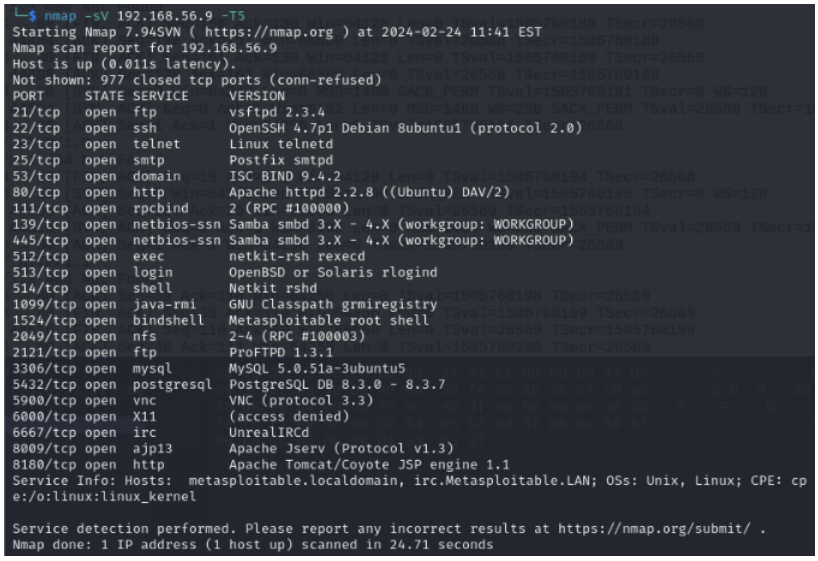

Figura 16: escaneo a la máquina 192.168.56.9

Ahora, buscaremos vulnerabilidades para los servicios encontrados. Para algunos servicios no ha sido posible incluir vulnerabilidades ya que no son compatibles con la versión del sistema operativo de las máquinas víctima.

- **Vulnerabilidad para HTTP en la versión Microsoft IIS httpd 7.5:** CVE-2010-2730: Buffer overflow in Microsoft Internet Information Services (IIS) 7.5, when FastCGI is enabled, allows remote attackers to execute arbitrary code via crafted headers in a request, aka Request Header Buffer Overflow Vulnerability."

- **Vulnerabilidad para FTP en la versión vsftpd 2.3.4:** CVE-2011-2523 vsftpd 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which opens a shell on port 6200/tcp.
- **Vulnerabilidad para SSH en la versión OpenSSH 4.7p1 Debian Subuntu1 (protocol 2.0):** CVE-2008-5161 Error handling in the SSH protocol in OpenSSH 4.7p1 and possibly other versions, when using a block cipher algorithm in Cipher Block Chaining (CBC) mode, makes it easier for remote attackers to recover certain plaintext data from an arbitrary block of ciphertext in an SSH session via unknown vectors.
- **Vulnerabilidad para telnet en la versión Linux telnetd:** CVE-2000-1195 telnet daemon (telnetd) from the Linux netkit package before netkit-telnet-0.16 allows remote attackers to bypass authentication when telnetd is running with the -L command line option.
- **Vulnerabilidad para smtp en la versión Postfix smtpd:** Vulnerability in Postfix SMTP server before 20010228-pl07, when configured to email the postmaster when SMTP errors cause the session to terminate, allows remote attackers to cause a denial of service (memory exhaustion) by generating a large number of SMTP errors, which forces the SMTP session log to grow too large.
- **Vulnerabilidad para rpcbind en la versión 2(RPC 100000):** CVE-1999-0461 Versions of rpcbind including Linux, IRIX, and Wietse Venema's rpcbind allow a remote attacker to insert and delete entries by spoofing a source address.
- **Vulnerabilidad para nfs en la versión 2-4 (RPC 100003):** CVE-2021-31976 Server for NFS Information Disclosure Vulnerability.
- **Vulnerabilidad para ftp en la versión ProFTPD 1.3.1:** CVE-2008-4242 ProFTPD 1.3.1 interprets long commands from an FTP client as multiple commands, which allows remote attackers to conduct cross-site request forgery (CSRF) attacks and execute arbitrary FTP commands via a long ftp:// URI that leverages an existing session from the FTP client implementation in a web browser.
- **Vulnerabilidad para mysql en la versión MySQL 5.0.51a-3ubuntu5:** MySQL 5.0.51a allows local users to bypass certain privilege checks by calling CREATE TABLE on a MyISAM table with modified (1) DATA DIRECTORY or (2) INDEX DIRECTORY arguments that are associated with symlinks within pathnames for subdirectories of the MySQL home data directory, which are followed when tables are created in the future.
- **Vulnerabilidad para vnc en la versión PostgreSQL VNC (protocol 3.3)** MySQL 5.0.51a allows local users to bypass certain privilege checks by calling CREATE TABLE on a MyISAM table with modified (1) DATA DIRECTORY or (2) INDEX DIRECTORY arguments that are associated with symlinks within pathnames for subdirectories of the MySQL home data directory, which are followed when tables are created in the future.

5. Describir las diferencias observadas en relación al descubrimiento de los equipos disponibles.

A lo largo de la práctica se ha usado Nmap para el descubrimiento de los equipos. Ahora usaré otra herramienta para esta tarea y haré una comparación de las diferencias.
He utilizado un script con **Scapy** que manda pings a un rango de direcciones, los que contesten serán considerados hosts activos:

```python
#!/usr/bin/env python

from scapy.all import *

start_ip = "192.168.56.1"
end_ip = "192.168.56.254"

def send_icmp_packet(ip):
    target_ip = ip
    print("Enviando paquete ICMP a", target_ip)
    icmp_packet = IP(dst=target_ip)/ICMP()
    response = sr1(icmp_packet, timeout=1, verbose=False)
    if response:
            print(target_ip, " Host UP")
    else:
        print(target_ip, " Host DOWN")

for i in range(int(start_ip.split('.')[3]), int(end_ip.split('.')[3]) + 1):
    ip = start_ip.rsplit('.', 1)[0] + '.' + str(i)
    send_icmp_packet(ip)
```

Figura 17: Script con Scapy

```
┌──(kali㉿kali)-[~]
└─$ sudo ./script_scapy.py
Enviando paquete ICMP a 192.168.56.1
192.168.56.1  Host UP
Enviando paquete ICMP a 192.168.56.2
192.168.56.2  Host DOWN
Enviando paquete ICMP a 192.168.56.3
192.168.56.3  Host DOWN
Enviando paquete ICMP a 192.168.56.4
192.168.56.4  Host DOWN
Enviando paquete ICMP a 192.168.56.5
192.168.56.5  Host DOWN
Enviando paquete ICMP a 192.168.56.6
192.168.56.6  Host DOWN
Enviando paquete ICMP a 192.168.56.7
192.168.56.7  Host DOWN
Enviando paquete ICMP a 192.168.56.8
192.168.56.8  Host DOWN
Enviando paquete ICMP a 192.168.56.9
192.168.56.9  Host UP
Enviando paquete ICMP a 192.168.56.10
192.168.56.10  Host DOWN
```

Figura 18: Ejecución del script con Scapy

Como podemos ver, Nmap no es la única herramienta para el descubrimiento de dispositivos. Sin embargo, es una opción muy completa, eficiente y sencilla de usar. Nmap tiene la capacidad de enviar pings únicamente a las máquinas que responden a sus solicitudes ARP. Por otro lado, en el ejemplo utilizando Scapy se envían pings a cada una de las direcciones IP de la red, lo que resulta en un escaneo más lento.

6. Usando la funcionalidad NSE buscar las vulnerabilidades SMB de los equipos disponibles

Para buscar vulnerabilidades usando la funcionalidad NSE tendremos que añadir la opción –**script=smb-vuln***. Lo haremos para cada máquina.
Para la máquina Windows:

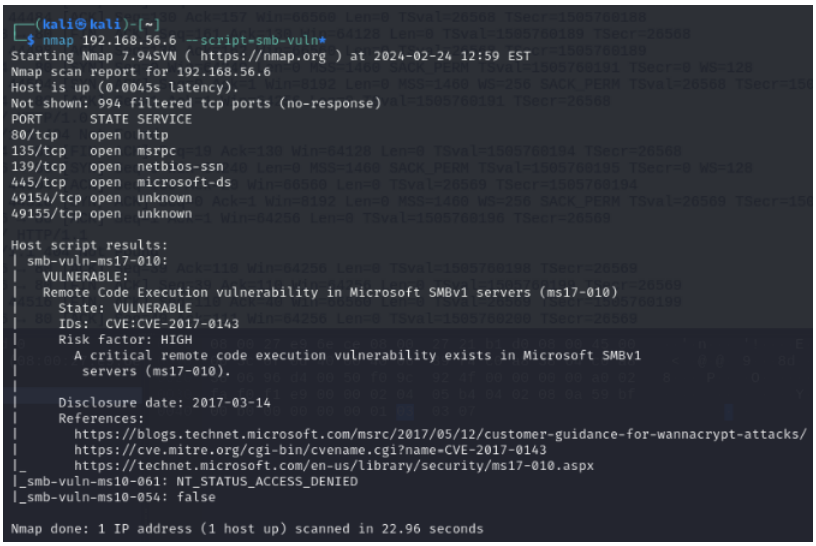

Figura 19: escaneo a la máquina 192.168.56.6

Como podemos ver, se ha encontrado una vulnerabilidad de ejecución de código remoto.
Para la máquina Linux:

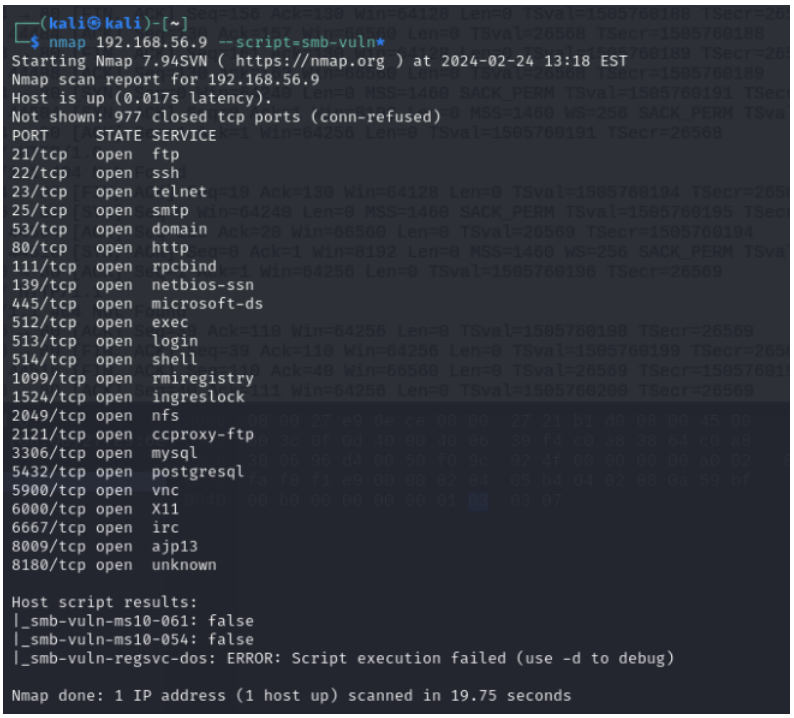

Figura 20: escaneo a la máquina 192.168.56.9

Podemos ver que la máquina no cuenta con vulnerabilidades SMB, por lo que no podemos explotarla

por ahí.

7. Usando la funcionalidad NSE comprueba si el servicio http permite negociación de contenido.

Para comprobar si el servicio http permite negociación de contenido con la funcionalidad NSE usamos la opción –**script=http-apache-negotiation**. Lo comprobamos en las dos máquinas.
En la máquina Windows:

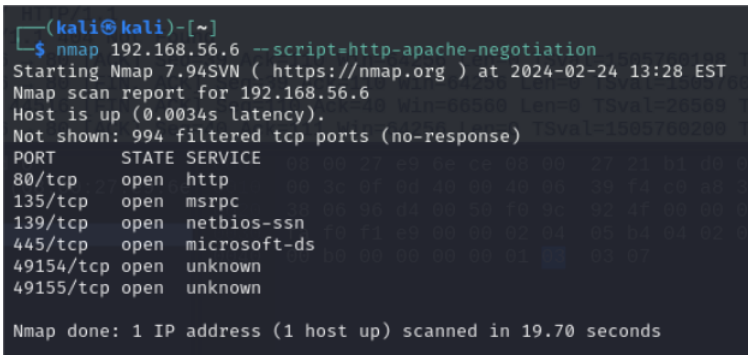

Figura 21: escaneo a la máquina 192.168.56.6

Como podemos ver, el servicio http no permite negociación de contenido.
En la máquina Linux:

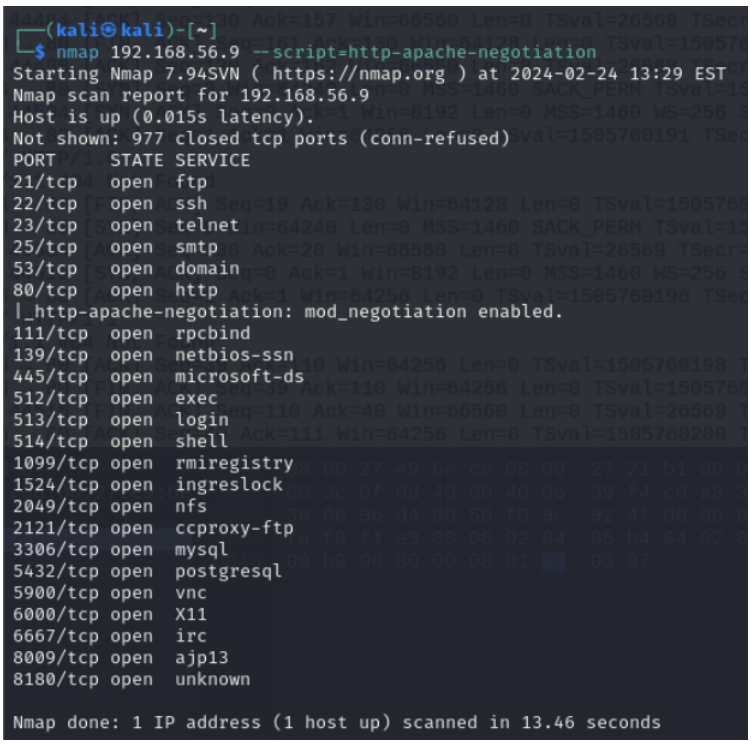

Figura 22: escaneo a la máquina 192.168.56.9