iGNITE Technologies

Nmap for Pentester

PACKET TRACE

WWW.HACKINGARTICLES.IN

Contenido

Introducción

Hola a todos. Hoy veremos cómo capturar paquetes de red usando nmap. Y usaremos Wireshark para comparar sus resultados con nmap. En este artículo, nos centramos principalmente en qué tipos de tráfico de red captura nmap mientras utilizamos varios escaneos de ping de nmap.

Se realiza un escaneo de ping en Nmap para verificar si el host de destino está vivo o no. Como sabemos, ping envía de forma predeterminada la solicitud de eco ICMP y obtiene una respuesta de eco ICMP si el sistema está activo. El escaneo de ping envía de forma predeterminada un paquete ARP y obtiene una respuesta para verificar si el host está activo.

NOTA: Los escaneos de Nmap cambian su comportamiento según la red que están escaneando.

- Escaneo de la red local con nmap donde nmap envía un paquete ARP con cada escaneo.
- Si se va a escanear una red externa; nmap envía los siguientes paquetes de solicitud:

Solicitud de eco ICMP

Solicitud de marca de tiempo ICMP

TCP SYN al puerto 443

TCP ACK al puerto 80

La técnica implica el seguimiento de paquetes a través de nmap.

El módulo nmap es una interfaz con las funciones internas y estructuras de datos de nmap. La API ofrece información del host de destino, como estados de puertos y resultados de detección de versiones. También proporciona una interfaz para la biblioteca Nsock para una E/S de red eficaz.

Nsock es una biblioteca de sockets paralelos utilizada por NSE, detección de servicios (service_scan.cc) y DNS (nmap_dns.cc). Actúa como una capa de abstracción sobre las operaciones de socket y está optimizado para manejar múltiples sockets. "mspool" se define en "nsock_internal.h" y contiene, entre otras cosas, una estructura event_lists, que es una estructura que mantiene información sobre todos los eventos pendientes.

Creación de eventos

Los eventos se representan con la estructura msevent (nsock_internal.h) que contiene (entre otras cosas)

- El controlador de devolución de llamada -> nsock_ev_handler (nsock_pool, nsock_event, void *)
- Un puntero a una estructura msiod -> msiod *iod, que contiene todos los descriptores de E/S (IOD) relacionados. información.
- Estructurar espacio de archivos iobuf (un búfer generalmente de 1024 bytes que contiene los bytes de escritura/lectura)
- El nse_type (nsock.h)
- El nse_status (nsock.h)
- Una identificación única -> nsock_event_id (EID)

Los eventos se crean con las siguientes funciones especiales:

nsock_connect.c

- nsock_connect_tcp
- nsock_connect_udp
- nsock_connect_ssl
- nsock_reconnect_ssl

nsock_read.c

- nsock_readlines
- nsock_readbytes
- nsock_read

nsock_write.c

- nsock_write
- nsock_printf

nsock_timer_create.c

- nsock_timer_create

fuente: https://sock-raw.org/nmap-ncrack/nsock.html

Análisis de ping de barrido de Nmap

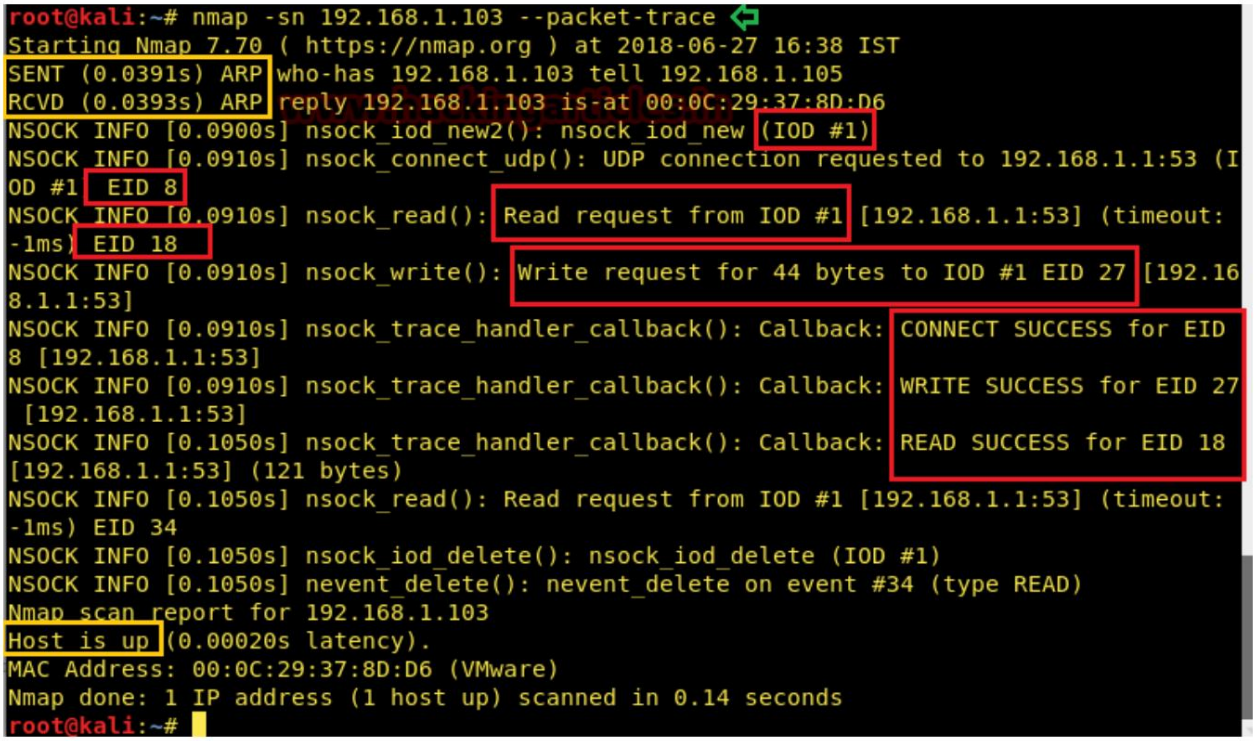
Los atributos -sn/ -sP se utilizan para hacer ping de barrido e intentan identificar el host activo en la red. Usando -

El seguimiento de paquetes a lo largo de la exploración de nmap podemos observar el paquete de red.

```
nmap -sn 192.168.1.103 --rastreo de paquetes
```

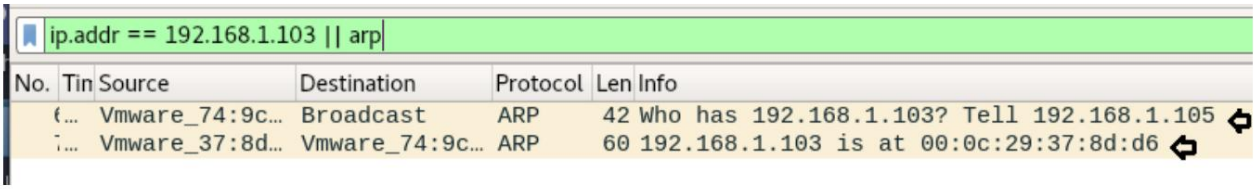
Aquí puede observar los dos primeros paquetes ENVIADOS/RECD (recibidos), que muestran un paquete de solicitud ARP de 192.168.1.105 a 192.168.1.103 y luego usar las bibliotecas NSOCK para indicar los paquetes de solicitud y respuesta reales que viajan entre el enrutador de origen y el de destino.

- NSOCK INFO que indica un nuevo nsock_event_id (EID) 8 se genera para representar el descriptor de E/S (IOD) n.º 1 para la solicitud de conexión NSOCK UDP al enrutador en el puerto 53.
- NSOCK INFO que denota otro (EID) 18 se genera para representar la solicitud de lectura del (IOD) #1.
- NSOCK INFO que denota otro (EID) 27 se genera para representar una solicitud de escritura de 44 bytes a (IOD) #1.
- INFORMACIÓN DE NSOCK que denota una operación EXITOSA cuando nsock usó callback_handler para conectarse para el DIE 8.
- INFORMACIÓN DE NSOCK que denota una operación EXITOSA cuando nsock usó callback_handler para escribir IDE 27.
- INFORMACIÓN DE NSOCK que denota una operación EXITOSA cuando nsock usó callback_handler para leer IDE 18.
- Información de NSOCK de que se eliminó el IOD #1.
- Información de NSOCK que nevent_delete está eliminando en el evento 34.
- En el último informe de escaneo de Nmap, el host está activo.



Puedes observar el mismo tráfico que hemos capturado desde Wireshark.

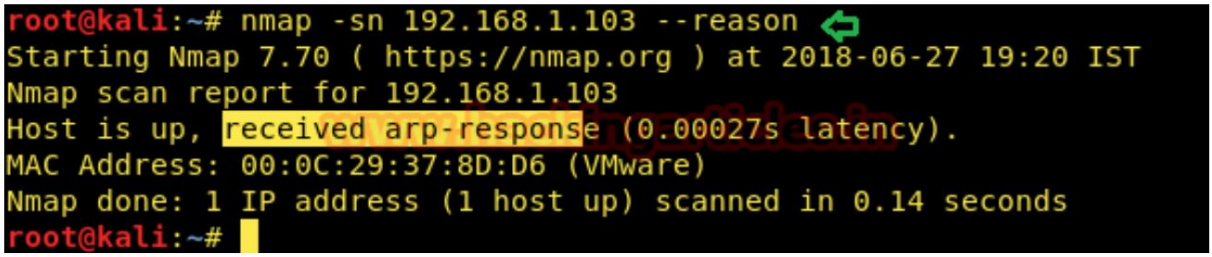
- Paquete de solicitud Arp de 192.168.1.105 a 192.168.1.103 • Paquete de respuesta Arp de 192.168.1.103 a 192.168.1.105



Para enumerar las respuestas de la red host, utilice la opción -reason con el comando nmap.

```
nmap -sn 192.168.1.103 --razón
```

Como puede observar, ha mostrado claramente que el host está activo cuando recibe una respuesta arp.



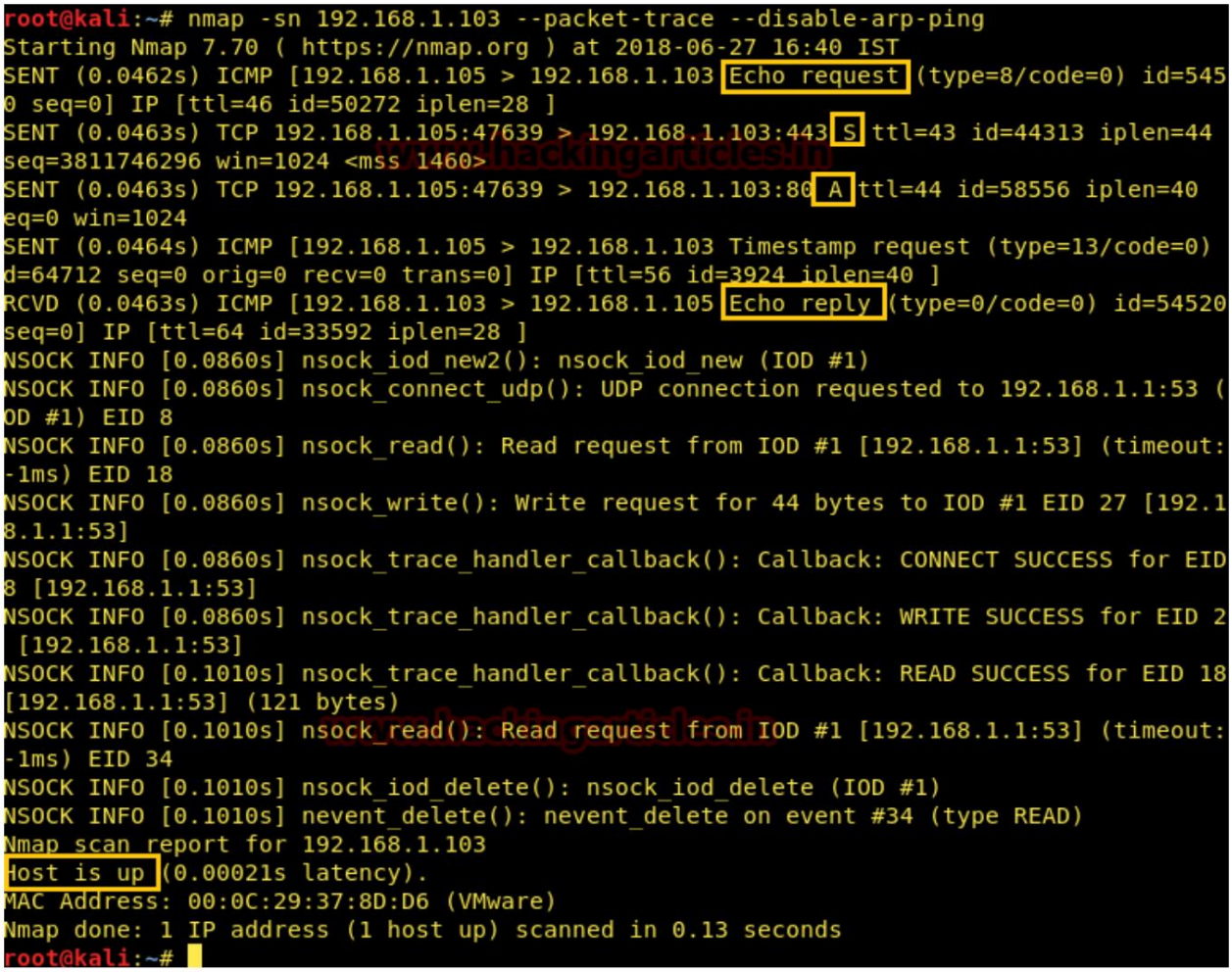
Como hemos visto, Nmap envía un paquete ARP de forma predeterminada para identificar el estado del host, por lo que ahora rastrearemos el paquete Nmap cuando -disable-arp-ping esté habilitado.

```
nmap -sn 192.168.1.103 --packet-trace --disable-arp-ping
```

Aquí puede observar los siguientes paquetes ENVIADOS desde el origen 192.168.1.105 al destino 192.168.1.103.

- Solicitud de eco ICMP •
  Solicitud de marca de tiempo ICMP
- TCP SYN al puerto 443 •
  TCP ACK al puerto 80

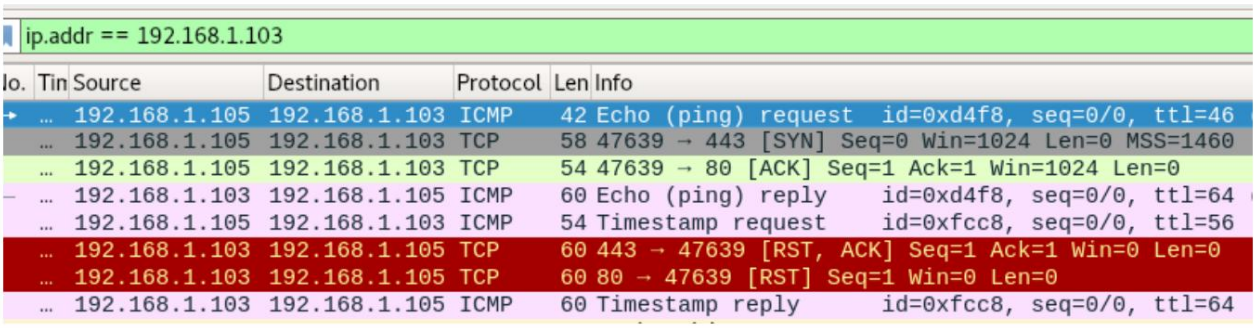
Luego, paquete RCVD ICMP Echo-resply desde el destino 192.168.1.103 y luego usó bibliotecas NSOCK para indicar que los paquetes de solicitud y respuesta reales viajan entre el origen y el enrutador de destino.



Demostrando el funcionamiento de Ping Sweep usando Wireshark

En la imagen que se muestra a continuación, puede observar el siguiente paquete de solicitud y respuesta entre ambas IP de red.

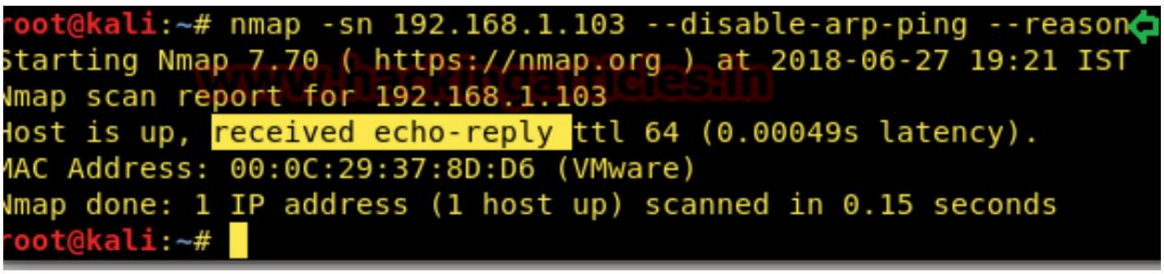
1. Solicitud de eco ICMP 2.
TCP SYN al puerto 443 3.
TCP ACK al puerto 80 4.
Solicitud de marca de tiempo ICMP
5. Respuesta de eco
ICMP 6. TCP RST, ACK al puerto
443 7. TCP RST al puerto
80 8. Respuesta de marca de tiempo ICMP



Para enumerar las respuestas de la red host, utilice la opción -reason con el comando nmap.

```
nmap -sn 192.168.1.103 --disable-arp-ping --razón
```

Como puede observar, se muestra claramente que el host está activo cuando recibe la respuesta de eco ICMP.



Análisis de ping de Nmap TCP-SYN

El atributo -PS envía el paquete TCP SYN en el puerto 80 de forma predeterminada; podemos cambiarlo especificando los puertos con él, como -P22.

```
nmap -PS -p22 192.168.1.103 --rastreo de paquetes
```

Aquí puede observar que este escaneo es la adición del escaneo de ping de nmap y el escaneo sigiloso de nmap porque, al principio, envía un paquete arp, luego usa bibliotecas nsock y, al final, nuevamente implica la mitad de la comunicación TCP.

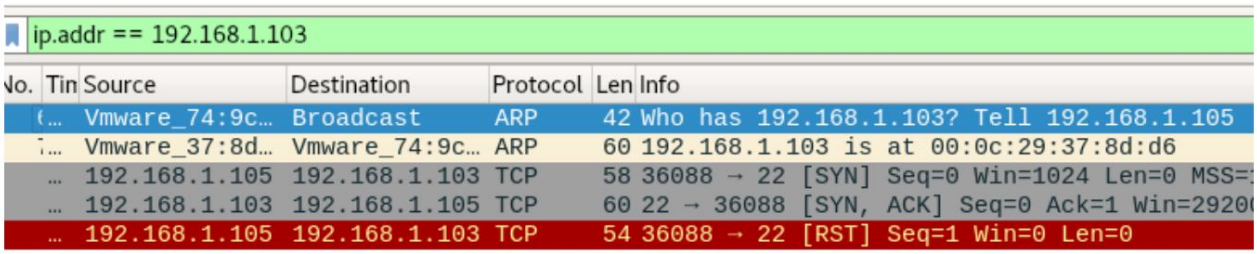
Entonces, puede observar la siguiente información que obtuvimos de nmap:

- Solicitud y respuesta ARP ENVIADA/RECD respectivamente. • Detalles de las bibliotecas Nsock
- Paquete TCP-SYN de 192.168.1.105:36088 a 192.168.1.103:22. • Paquete TCP-SYN/ACK de 192.168.1.103:22 a 192.168.1.105:36088.

```
root@kali:~# nmap -PS -p22 192.168.1.103 --packet-trace
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 16:58 IST
SENT (0.0670s) ARP who-has 192.168.1.103 tell 192.168.1.105
RCVD (0.0672s) ARP reply 192.168.1.103 is-at 00:0C:29:37:8D:D6
NSOCK INFO [0.1200s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1200s] nsock_connect_udp(): UDP connection requested to 192.168.1.1:53 (I
OD #1) EID 8
NSOCK INFO [0.1200s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout:
-1ms) EID 18
NSOCK INFO [0.1210s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [192.16
8.1.1:53]
NSOCK INFO [0.1210s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID
8 [192.168.1.1:53]
NSOCK INFO [0.1210s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27
 [192.168.1.1:53]
NSOCK INFO [0.1360s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18
[192.168.1.1:53] (121 bytes)
NSOCK INFO [0.1360s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout:
-1ms) EID 34
NSOCK INFO [0.1360s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.1360s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.1847s) TCP 192.168.1.105:36088 > 192.168.1.103:22 S ttl=42 id=19516 iplen=44  s
eq=683521233 win=1024 <mss 1460>
RCVD (0.1850s) TCP 192.168.1.103:22 > 192.168.1.105:36088 SA ttl=64 id=0 iplen=44  seq=
642256733 win=29200 <mss 1460>
Nmap scan report for 192.168.1.103
Host is up (0.00022s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:37:8D:D6 (VMware)
```

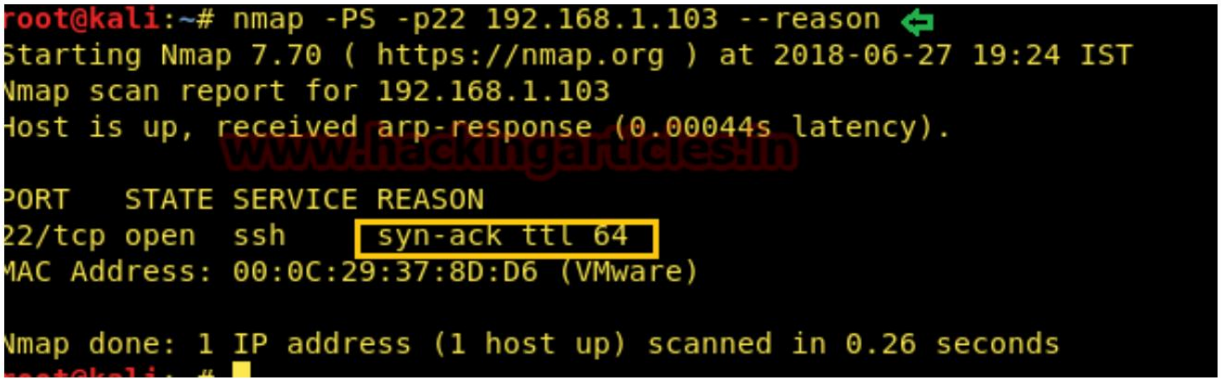
De manera similar, vimos el mismo patrón de tráfico de red en Wireshark.



De manera similar, también puede elegir la opción –reason con el comando nmap para enumerar la respuesta de la red host.

```
nmap -PS -p22 192.168.1.103 --razón
```

Aquí puede observar que el puerto 22 está abierto y cuando se recibe el paquete SYN/ACK del host.



Ahora averigüemos el tráfico de la red cuando se activa –disable-arp-ping.

```
nmap -PS -p22 192.168.1.103 --packet-trace --disable-arp-ping
```

Entonces, puede observar la siguiente información que obtuvimos de nmap:

• Paquete TCP-SYN ENVIADO en el puerto
80 • RCVD TCP-RST/ACK desde el puerto
80. • Detalles de bibliotecas
Nsock • Paquete TCP-SYN de 192.168.1.105:63581 a 192.168.1.103:22. • Paquete
TCP-SYN/ACK de 192.168.1.103:22 a 192.168.1.105:63851.

```
root@kali:~# nmap -PS -p22 192.168.1.103 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 17:02 IST
SENT (0.0687s) TCP 192.168.1.105:63595 > 192.168.1.103:80 S ttl=58 id=43386 iplen
=44  seq=3631585945 win=1024 <mss 1460>
RCVD (0.0689s) TCP 192.168.1.103:80 > 192.168.1.105:63595 RA ttl=64 id=35594 iple
n=40  seq=0 win=0
NSOCK INFO [0.1280s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1280s] nsock_connect_udp(): UDP connection requested to 192.168.1.1
:53 (IOD #1) EID 8
NSOCK INFO [0.1280s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (tim
eout: -1ms) EID 18
NSOCK INFO [0.1280s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [
192.168.1.1:53]
NSOCK INFO [0.1280s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS fo
r EID 8 [192.168.1.1:53]
NSOCK INFO [0.1280s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for
EID 27 [192.168.1.1:53]
NSOCK INFO [0.1430s] nsock_trace_handler_callback(): Callback: READ SUCCESS for E
ID 18 [192.168.1.1:53] (121 bytes)
NSOCK INFO [0.1430s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (tim
eout: -1ms) EID 34
NSOCK INFO [0.1430s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.1430s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.1948s) TCP 192.168.1.105:63851 > 192.168.1.103:22 S ttl=52 id=8113 iplen=
44  seq=3751894127 win=1024 <mss 1460>
RCVD (0.1952s) TCP 192.168.1.103:22 > 192.168.1.105:63851 SA ttl=64 id=0 iplen=44
  seq=1223132932 win=29200 <mss 1460>
Nmap scan report for 192.168.1.103
Host is up (0.00026s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:37:8D:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.28 seconds
```

De manera similar, también vimos el mismo patrón de tráfico de red en Wireshark.

ip.addr == 192.168.1.103

.	Tin	Source	Destination	Protocol	Len	Info
`	…	192.168.1.105	192.168.1.103	TCP	58	63595 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
{	…	192.168.1.103	192.168.1.105	TCP	60	80 → 63595 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
	…	192.168.1.105	192.168.1.103	TCP	58	63851 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
	…	192.168.1.103	192.168.1.105	TCP	60	22 → 63851 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len
	…	192.168.1.105	192.168.1.103	TCP	54	63851 → 22 [RST] Seq=1 Win=0 Len=0

Análisis de ping ICMP de Nmap

-PE envía un paquete de solicitud de eco ICMP [ICMP tipo 8] y recibe un paquete de respuesta de eco ICMP.

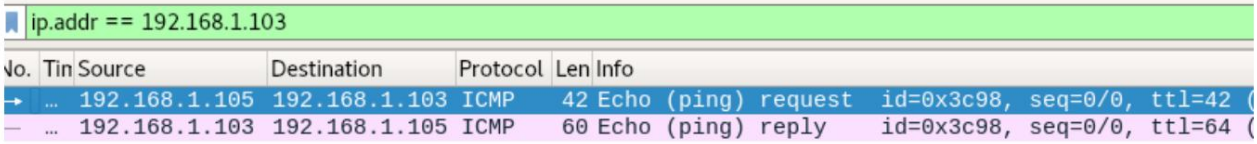
```
nmap -sP -PE 192.168.1.103 --packet-trace --disable-arp-ping
```

Aquí puede observar los paquetes de solicitud de eco ICMP ENVIADOS desde el origen 192.168.1.105 al destino 192.168.1.103

Luego, paquete RCVD ICMP Echo-resply desde el destino 192.168.1.103 y luego usó bibliotecas NSOCK para indicar los paquetes de solicitud y respuesta reales que viajan entre el enrutador de origen y el de destino.

```
root@kali:~# nmap -sP -PE 192.168.1.103 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 17:12 IST
SENT (0.0346s) ICMP [192.168.1.105 > 192.168.1.103 Echo request (type=8/code=0) i
d=15512 seq=0] IP [ttl=42 id=10543 iplen=28 ]
RCVD (0.0348s) ICMP [192.168.1.103 > 192.168.1.105 Echo reply (type=0/code=0) id=
15512 seq=0] IP [ttl=64 id=36594 iplen=28 ]
NSOCK INFO [0.0860s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.0860s] nsock_connect_udp(): UDP connection requested to 192.168.1.1
:53 (IOD #1) EID 8
NSOCK INFO [0.0860s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (tim
eout: -1ms) EID 18
NSOCK INFO [0.0860s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [
192.168.1.1:53]
NSOCK INFO [0.0860s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS fo
r EID 8 [192.168.1.1:53]
NSOCK INFO [0.0860s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for
EID 27 [192.168.1.1:53]
NSOCK INFO [0.1010s] nsock_trace_handler_callback(): Callback: READ SUCCESS for E
ID 18 [192.168.1.1:53] (121 bytes)
NSOCK INFO [0.1010s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (tim
eout: -1ms) EID 34
NSOCK INFO [0.1010s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.1010s] nevent_delete(): nevent_delete on event #34 (type READ)
Nmap scan report for 192.168.1.103
Host is up (0.00023s latency).
MAC Address: 00:0C:29:37:8D:D6 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

De manera similar, también vimos el mismo patrón de tráfico de red en Wireshark.



Análisis de escaneo sigiloso de Nmap

Capturemos el paquete de red para el escaneo nmap predeterminado, también llamado escaneo sigiloso, que sigue la media comunicación TCP.

```
mapa n -p22 192.168.1.103
```

```
root@kali:~# nmap -p22 192.168.1.103
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 16:25 IST
Nmap scan report for 192.168.1.103
Host is up (0.00069s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:37:8D:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
root@kali:~#
```

Aquí puede observar la comunicación TCP-mitad:

• Paquete TCP-SYN enviado desde el origen 192.168.1.105 a 192.168.1.103 en el puerto 22. • Paquete TCP-SYN, ACK recibido desde el origen 192.168.1.103 a 192.168.1.105. • Paquete TCP-RST enviado desde la fuente 192.168.1.105 a 192.168.1.103.

ip.addr == 192.168.1.103

lo.	Tin	Source	Destination	Protocol	Len	Info
←	…	192.168.1.105	192.168.1.103	TCP	58	55598 → 22 [SYN] Seq=0 Win=1024 Len=0
	…	192.168.1.103	192.168.1.105	TCP	60	22 → 55598 [SYN, ACK] Seq=0 Ack=1 Win
←	…	192.168.1.105	192.168.1.103	TCP	54	55598 → 22 [RST] Seq=1 Win=0 Len=0

Ahora verifiquémoslo con el parámetro -packet-trace y comparemos el resultado.

nmap -p22 192.168.1.103 --rastreo de paquetes

Entonces puede observar la siguiente información que obtuvimos de nmap, que es similar a TCP-SYN Ping.

• Solicitud y respuesta ARP ENVIADA/RECD respectivamente.
• Detalles de las bibliotecas

Nsock • Paquete TCP-SYN de 192.168.1.105:48236 a 192.168.1.103:22. • Paquete TCP-SYN/ACK de 192.168.1.103:22 a 192.168.1.105:48236.

```
root@kali:~# nmap -p22 192.168.1.103 --packet-trace
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 16:28 IST
SENT (0.1717s) ARP who-has 192.168.1.103 tell 192.168.1.105
RCVD (0.1722s) ARP reply 192.168.1.103 is-at 00:0C:29:37:8D:D6
NSOCK INFO [0.2290s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.2290s] nsock_connect_udp(): UDP connection requested to 192.168.1.1:53 (IOD #1) EID 8
NSOCK INFO [0.2300s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.2300s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [192.168.1.1:53]
NSOCK INFO [0.2300s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [192.168.1.1:53]
NSOCK INFO [0.2300s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [192.168.1.1:53]
NSOCK INFO [0.2450s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [192.168.1.1:53] (121 bytes)
NSOCK INFO [0.2450s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeout: -1ms) EID 34
NSOCK INFO [0.2450s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.2450s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.2865s) TCP 192.168.1.105:48236 > 192.168.1.103:22 S ttl=38 id=41206 iplen=44  seq=2585637670 win=1024 <mss 1460>
RCVD (0.2870s) TCP 192.168.1.103:22 > 192.168.1.105:48236 SA ttl=64 id=0 iplen=44  seq=2604218680 win=29200 <mss 1460>
Nmap scan report for 192.168.1.103
Host is up (0.00048s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:37:8D:D6 (VMware)
```

De manera similar, puede usar el comando nmap con la opción "-reason" para enumerar las respuestas de la red host.

```
nmap -p22 192.168.1.103 --razón
```

Aquí puede observar que el puerto 22 está abierto y cuando se recibe el paquete SYN/ACK del host.

```
root@kali:~# nmap -p22 192.168.1.103 --reason
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 16:27 IST
Nmap scan report for 192.168.1.103
Host is up, received arp-response (0.00053s latency).

PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
MAC Address: 00:0C:29:37:8D:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
root@kali:~#
```

Ahora averigüemos el comportamiento del tráfico de red cuando se activa –disable-arp-ping.

```
nmap -p22 192.168.1.103 --packet-trace --disable-arp-ping
```

Aquí puede observar los siguientes paquetes ENVIADOS desde el origen 192.168.1.105 al destino 192.168.1.103.

• ENVIÓ solicitud de eco ICMP •
ENVIÓ TCP SYN al puerto 443 •
ENVIÓ TCP ACK al puerto 80 •
ENVIÓ solicitud de marca de tiempo ICMP
• Luego paquete RCVD Respuesta de eco ICMP desde el destino 192.168.1.103 • Luego
usó bibliotecas NSOCK para indicar el viaje real de los paquetes de solicitud y respuesta entre fuentes
al enrutador de destino.
• Solicitud TCP-SYN ENVIADA en el puerto 22
• RECV TCP-SYN, respuesta ACK desde el puerto 22.

```
root@kali:~# nmap -p22 192.168.1.103 --packet-trace --disable-arp-ping
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 16:33 IST
SENT (0.0744s) ICMP [192.168.1.105 > 192.168.1.103 Echo request (type=8/code=0) id=
58786 seq=0] IP [ttl=53 id=2137 iplen=28 ]
SENT (0.0744s) TCP 192.168.1.105:37388 > 192.168.1.103:443 S ttl=51 id=15499 iplen=
44  seq=3691325058 win=1024 <mss 1460>
SENT (0.0745s) TCP 192.168.1.105:37388 > 192.168.1.103:80 A ttl=51 id=61797 iplen=4
0  seq=0 win=1024
SENT (0.0746s) ICMP [192.168.1.105 > 192.168.1.103 Timestamp request (type=13/code=
0) id=55380 seq=0 orig=0 recv=0 trans=0] IP [ttl=50 id=24463 iplen=40 ]
RCVD (0.0746s) ICMP [192.168.1.103 > 192.168.1.105 Echo reply (type=0/code=0) id=58
786 seq=0] IP [ttl=64 id=33588 iplen=28 ]
NSOCK INFO [0.1230s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1230s] nsock_connect_udp(): UDP connection requested to 192.168.1.1:5
3 (IOD #1) EID 8
NSOCK INFO [0.1230s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeo
ut: -1ms) EID 18
NSOCK INFO [0.1230s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [19
2.168.1.1:53]
NSOCK INFO [0.1230s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for
EID 8 [192.168.1.1:53]
NSOCK INFO [0.1230s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EI
D 27 [192.168.1.1:53]
NSOCK INFO [0.1370s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID
 18 [192.168.1.1:53] (121 bytes)
NSOCK INFO [0.1370s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (timeo
ut: -1ms) EID 34
NSOCK INFO [0.1370s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.1370s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.1770s) TCP 192.168.1.105:37644 > 192.168.1.103:22 S ttl=45 id=45820 iplen=4
4  seq=259118263 win=1024 <mss 1460>
RCVD (0.1774s) TCP 192.168.1.103:22 > 192.168.1.105:37644 SA ttl=64 id=0 iplen=44
seq=3066528596 win=29200 <mss 1460>
Nmap scan report for 192.168.1.103
Host is up (0.00030s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:37:8D:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

Por otro lado, también vimos el mismo patrón de tráfico de red en Wireshark.

ip.addr == 192.168.1.103

o.	Tin	Source	Destination	Protocol	Len	Info
…		192.168.1.105	192.168.1.103	ICMP	42	Echo (ping) request id=0xe5a2, seq=0/0, ttl=53 (reply in 11)
…		192.168.1.105	192.168.1.103	TCP	58	37388 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
…		192.168.1.105	192.168.1.103	TCP	54	37388 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
…		192.168.1.105	192.168.1.103	ICMP	54	Timestamp request id=0xd854, seq=0/0, ttl=50
…		192.168.1.103	192.168.1.105	ICMP	60	Echo (ping) reply id=0xe5a2, seq=0/0, ttl=64 (request in 7)
…		192.168.1.103	192.168.1.105	TCP	60	443 → 37388 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
…		192.168.1.103	192.168.1.105	TCP	60	80 → 37388 [RST] Seq=1 Win=0 Len=0
…		192.168.1.103	192.168.1.105	ICMP	60	Timestamp reply id=0xd854, seq=0/0, ttl=64
…		192.168.1.105	192.168.1.103	TCP	58	37644 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
…		192.168.1.103	192.168.1.105	TCP	60	22 → 37644 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
…		192.168.1.105	192.168.1.103	TCP	54	37644 → 22 [RST] Seq=1 Win=0 Len=0

Análisis de escaneo TCP de Nmap A partir

de nuestro conocimiento básico de comunicación de red, sabemos que un escaneo TCP realiza un protocolo de enlace de tres vías. Aquí se realiza un escaneo TCP de nmap:

```
nmap -sT -p22 192.168.1.103 --rastreo de paquetes
```

Entonces puede observar la siguiente información que obtuvimos de nmap, que es similar a TCP-SYN Ping.

Solicitud y respuesta ARP ENVIADA/RECD respectivamente.

Detalles de las bibliotecas de Nsock

La conexión de TCP Localhost desde el host de destino 192.168.1.103:22 está en curso.

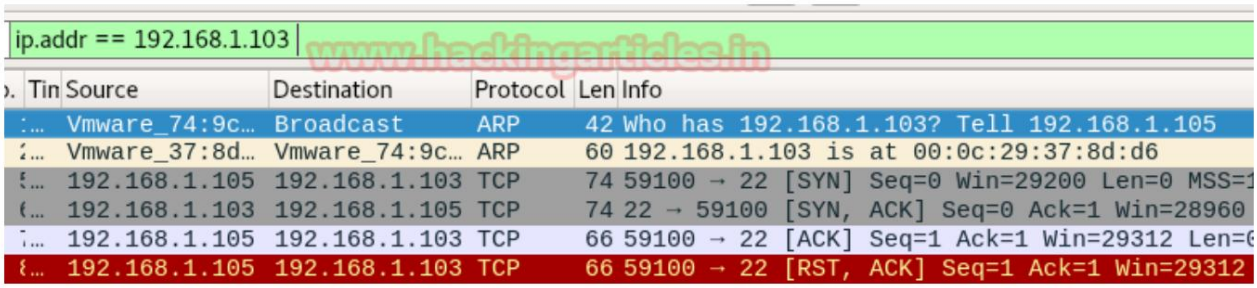
Se conectó TCP Localhost desde el host de destino 192.168.1.103:22 correctamente.

```
root@kali:~# nmap -sT -p22 192.168.1.103 --packet-trace
Starting Nmap 7.70 ( https://nmap.org ) at 2018-06-27 19:30 IST
SENT (0.0661s) ARP who-has 192.168.1.103 tell 192.168.1.105
RCVD (0.0663s) ARP reply 192.168.1.103 is-at 00:0C:29:37:8D:D6
NSOCK INFO [0.1140s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1140s] nsock_connect_udp(): UDP connection requested to 192.168.1.1
:53 (IOD #1) EID 8
NSOCK INFO [0.1140s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (tim
eout: -1ms) EID 18
NSOCK INFO [0.1140s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [
192.168.1.1:53]
NSOCK INFO [0.1140s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS fo
r EID 8 [192.168.1.1:53]
NSOCK INFO [0.1140s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for
EID 27 [192.168.1.1:53]
NSOCK INFO [0.1290s] nsock_trace_handler_callback(): Callback: READ SUCCESS for E
ID 18 [192.168.1.1:53] (121 bytes)
NSOCK INFO [0.1290s] nsock_read(): Read request from IOD #1 [192.168.1.1:53] (tim
eout: -1ms) EID 34
NSOCK INFO [0.1290s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.1290s] nevent_delete(): nevent_delete on event #34 (type READ)
CONN (0.1300s) TCP localhost > 192.168.1.103:22 => Operation now in progress
CONN (0.1306s) TCP localhost > 192.168.1.103:22 => Connected
Nmap scan report for 192.168.1.103
Host is up (0.00032s latency).

PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:37:8D:D6 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
```

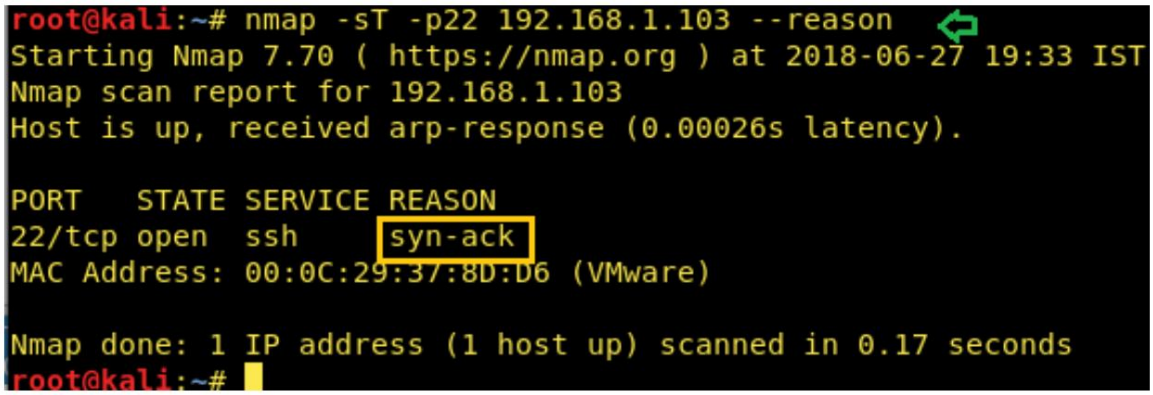
De manera similar, también vimos el mismo patrón de tráfico de red en Wireshark.



De manera similar, puede usar el comando nmap con la opción "-reason" para enumerar las respuestas de la red host.

```
nmap -sT -p22 192.168.1.103 --razón
```

Aquí puede observar que el puerto 22 está abierto y cuando se recibe el paquete SYN/ACK del host.



iGNITE Technologies

ÚNETE A NUESTRO PROGRAMAS DE ENTRENAMIENTO

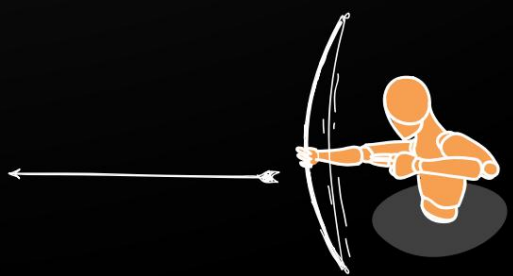
PRINCIPIANTE

- Hackeo ético
- Pentest de red
- Recompensa de errores
- Pentest inalámbrico
- Seguridad de la red Esenciales

AVANZADO

- Eructar Suite Pro
- Web Servicios-API
- Pro VAPT de infraestructura
- Computadora forense
- Pentest de Android
- Avanzado metasploit
- CTF

EXPERTO



- Operación del equipo rojo
  - APT's - Tácticas de ataque MITRE
  - Ataque de directorio activo
  - Evaluación de seguridad de MSSQL
- Escalada de privilegios
  - ventanas
  - linux

www.ignitetechnologies.in