

Sniffers y escaneo de puertos

Comisión: 4k9

Profesores: Mario Centeno, Julio Cuenca

Grupo 8- Integrantes:

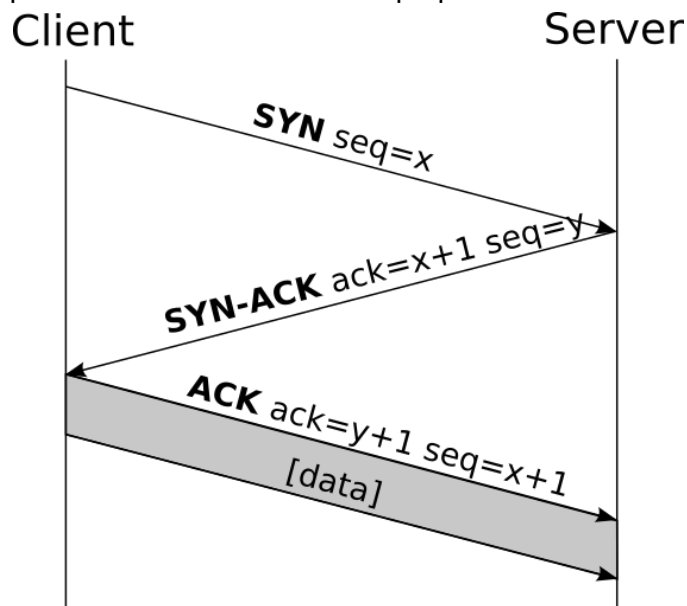
- Facundo Murello:
- Diego Paez: 46942
- Cristian Rosales: 47883
- Gonzalo Pozzoli: 49028

Índice

Índice	2
Introducción	3
Escaneo de puertos	3
Sniffers	6
Conclusión	9
Bibliografía	10

Introducción

El protocolo TCP/IP, que es orientado a la conexión, requiere que se realice un “handshake” entre los hosts que desean comunicarse. Cada uno de los hosts envía y recibe datos a través de sus puertos, que son interfaces de conexión a la red o a otros dispositivos. El handshake TCP/IP consiste en el envío de una señal SYN (synchronize) desde el host que inicia la conexión hacia el host con quien desea conectarse. El host que recibe la señal SYN puede responder con una señal SYN-ACK (synchronize-acknowledge) si desea establecer la conexión y además también enviar su número de secuencia inicial, y luego el host iniciador responde con una señal ACK para confirmar que la conexión fue establecida y que pueden comenzar a enviarse paquetes.



Esta habilidad de poder solicitar conexiones con cualquier host alcanzable puede presentar vulnerabilidades y exponer información sobre el setup de nuestros servidores y terminales si es que los descuidamos, como por ejemplo los firewalls que usamos (si es que usamos) hasta las aplicaciones que corren en cada puerto.

Además, dado que los paquetes comúnmente deben viajar a través de la red e Internet, implica que deben atravesar muchos puntos de enlace propensos a ser intervenidos por agentes maliciosos que podrían leer el contenido del paquete fácilmente.

Escaneo de puertos

El escaneo de puertos es una técnica que se utiliza para analizar por medio de un programa el estado de los puertos de una máquina que se encuentra conectada a una red, lo que permite averiguar qué puertos están escuchando y aceptando conexiones[\[1\]](#). Esta información se puede utilizar para determinar qué servicios se están ejecutando. Muchas veces esto lo hacen los “hackers” para robar información/explotar la vulnerabilidad o las propias empresas para poder descubrir puntos “débiles” y solucionar vulnerabilidades. [\[3\]](#)

Hay varias formas de escanear puertos en una red [2],[5].

- **Scans de ping:** Se utiliza el protocolo ICMP[4] para enviar solicitudes de eco (ping) desde una máquina emisora. Si el dispositivo objetivo está disponible, responde con un paquete de eco. Esta técnica se usa tanto para diagnosticar problemas de conectividad como para detectar dispositivos activos en la red, lo que puede ayudar a administradores de redes, pero también podría ser explotado por atacantes si no se implementan medidas de seguridad, como un firewall.
- **Stealth SYN Scan/Escaneo SYN:** Un escaneo SYN también se lo conoce como escaneo semiabierto debido a que no establece una conexión TCP completa. Esto significa que no completa el handshake, en su lugar, sólo se envía el paquete SYN inicial y se examina la respuesta:
 - Si se recibe un paquete SYN/ACK como respuesta, ese puerto debe estar aceptando conexiones.
 - Si se recibe un paquete RST significa que el puerto está cerrado.

En ambas situaciones se registra el resultado en la máquina emisora.

- **Escaneos XMAS, FIN y NULL:** En respuesta al escaneo SYN, se crearon nuevas herramientas para detectar y registrar conexiones semiabiertas. Así surgió otra colección de técnicas para el escaneo sigiloso de puertos: FIN, X-mas y Null. Todas ellas implican el envío de un paquete sin sentido a cada puerto del sistema objetivo basados en el RFC 793 (Estándar del protocolo TCP).
 - Escaneo FIN: En lugar de enviar un paquete SYN para iniciar una conexión, se envía un paquete con la bandera FIN activada, que normalmente se usa para finalizar una conexión.
 - Escaneo XMAS: envía un paquete con varias banderas activadas: FIN, URG, y PUSH. Esta combinación de banderas es poco común, lo que hace que el paquete parezca "ilógico" o "iluminado como un árbol de Navidad"
 - Escaneo NULL: Se envía un paquete sin ninguna bandera activada. Es un paquete "vacío", en el sentido de que no intenta iniciar, mantener ni finalizar una conexión.

Independientemente del que se utilice:

- Si el puerto está **abierto**, el sistema objetivo debe ignorar estos paquetes "sin sentido", ya que no tienen un propósito claro en una conexión TCP.
- Si el puerto está **cerrado**, el sistema debe responder con un paquete RST (Reset), lo que indica que no puede manejar esa "conexión inválida". Esta respuesta permite al atacante saber que el puerto está cerrado.
- **Suplantación de identidad con señuelos:** La máquina emisora falsifica (spoof) las direcciones IP de los paquetes que envía al sistema objetivo. Cada vez que envía un paquete legítimo desde su propia dirección IP (para escanear un puerto), también envía paquetes falsificados desde varias direcciones IP diferentes (los "señuelos"). Para que sea efectivo, las direcciones IP señuelo deben corresponder a hosts activos reales porque si se usan direcciones IP falsas, el sistema objetivo podría acabar enviando respuestas a direcciones que no existen.

De esta forma el sistema objetivo no puede diferenciar fácilmente cuál de los paquetes proviene de la dirección IP real del emisor y cuáles de los señuelos. Por lo que va a contestar a todas las direcciones.

Las direcciones IP señuelo no necesitan responder ya que solamente deben distraer al sistema objetivo.

- **Escaneo de vainilla:** El emisor envía un paquete con la bandera SYN. Este paquete se envía a TODOS los puertos de la máquina objetivo para ver si están abiertos. Si el puerto está abierto, el servidor responde con un paquete SYN-ACK, indicando que está dispuesto a completar la conexión. Cuando el emisor recibe el paquete SYN-ACK, envía de vuelta un paquete ACK, estableciendo una conexión TCP completa. Si el puerto está cerrado, la máquina objetivo responde con un paquete RST, que indica que la conexión no será permitida en ese puerto. Este escaneo es preciso pero fácilmente detectable porque los firewall siempre registran una conexión completa.
- **Escaneo UDP:** Es una técnica que usa el protocolo UDP (como DNS o SNMP) en un dispositivo objetivo. A diferencia de los escaneos que usan el protocolo TCP, este método no utiliza el proceso de handshake. En su lugar, se envían paquetes UDP (vacíos) a cada puerto del objetivo.
 - Si el puerto está **cerrado**, el sistema objetivo responde con un mensaje de "Puerto inalcanzable"/"Port unreachable".
 - Si el puerto está **abierto**, generalmente no hay respuesta, ya que UDP no tiene mecanismos de confirmación como TCP. La falta de respuesta suele indicar que el puerto está abierto, pero esto no siempre es definitivo.

A modo de resumen se deja el siguiente cuadro comparativo entre los diferentes métodos mencionados:

Tipo de Escaneo	Protocolo	Paquetes Enviados	Respuesta Esperada (Puerto Cerrado)	Respuesta Esperada (Puerto Abierto)	Observaciones
<i>Ping</i>	ICMP	Echo Request	Echo Reply	Ninguna (si no hay respuesta)	Básico, detecta dispositivos activos.
<i>SYN</i>	TCP	SYN	RST	SYN-ACK	Semi-abierto, no establece conexión completa.
<i>FIN, XMAS, NULL</i>	TCP	FIN, FIN+URG+PU SH, Ninguna	RST	Ninguna (generalmente)	Sigilosos, pero dependen de la implementación TCP.
<i>Suplantación con Señuelos</i>	TCP	SYN (con direcciones IP falsas)	RST a todas las direcciones	SYN-ACK a la dirección real	Dificulta identificar el escaneo, pero requiere direcciones IP válidas.
<i>Vainilla</i>	TCP	SYN	RST	SYN-ACK, ACK	Establece conexiones completas, fácil de detectar.
<i>UDP</i>	UDP	Paquete UDP	"Puerto inalcanzable"	Ninguna (generalmente)	No confiable, ya que UDP no garantiza la entrega.

Sniffers

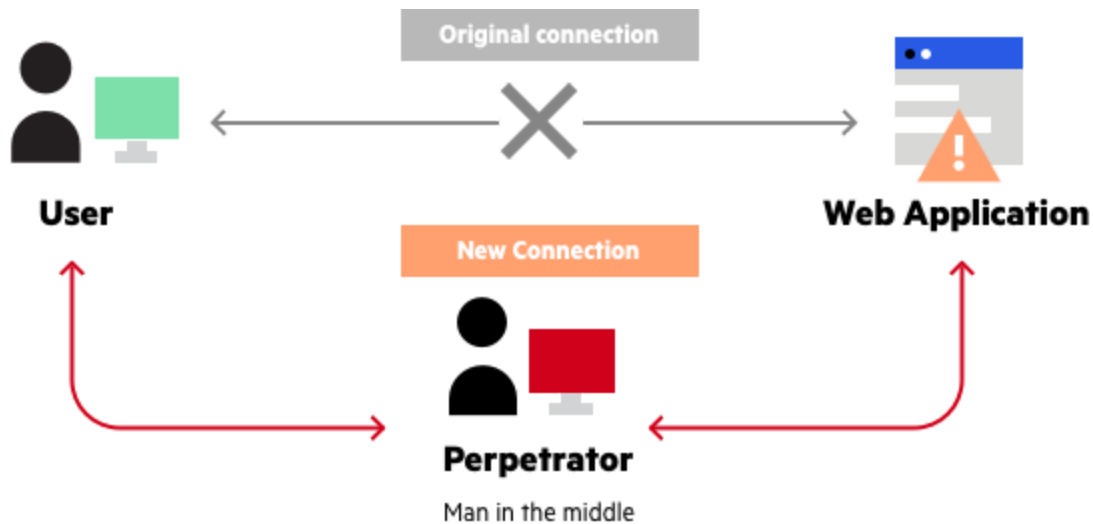
Un *sniffer*^[7] es un software que se acopla a la capa de enlace de datos y le da al usuario la capacidad de capturar y leer los paquetes que atraviesan algún host específico al que tiene acceso, como por ejemplo su computadora o el router al que está conectado. Si bien capturan las tramas de esa capa, son capaces de extraer la información encapsulada que traen de las capas superiores. Tienen la habilidad de filtrar paquetes y de esa forma solo capturar los de interés al usuario, como por ejemplo capturar todos los paquetes provenientes de cierta IP, o dirigidos a cierta IP.

Dependiendo de la red tenemos distintos tipos de sniffing: **activo** y **pasivo**. En una red principalmente formada por hubs, donde los paquetes se transmiten a todos los hosts conectados al hub, estos hosts tienen por defecto la orden de descartar todos los paquetes que no son dirigidos a ellos. En este caso se puede implementar un sniffer **pasivo** que simplemente ignora la orden de descartar los paquetes (colocando a la tarjeta de red en modo promiscuo, que hace que capture todos los paquetes), y los almacena en un archivo sin importar a qué host están realmente dirigidos.

El sniffing **activo** es el que se implementa en redes con routers y switches, ya que se debe intervenir la red para lograr la captura de los paquetes. En un subred de un switch, todos los paquetes entre un host y el switch solo viajan entre ellos, y no entre los demás hosts de la red. Tanto el switch como el host usan el protocolo **ARP** (Protocolo de Resolución de Direcciones) para saber a dónde dirigir cada paquete, disponiendo de una tabla que relaciona la IP de cada host con su dirección MAC, y que es usada cada vez que se envían paquetes en la subred.

El usuario malicioso puede enviar paquetes del protocolo ARP que le indican al switch que debe modificar la dirección MAC de la IP de la víctima por la MAC del hacker, e indicando también al host víctima que debe modificar la MAC del switch por la del hacker también. De esta forma, tanto el switch como la víctima apuntan al hacker, que podrá leer y redirigir los paquetes que se envían entre ellos.

El tipo de sniffing activo suele conocerse como un ataque **man in the middle** (hombre en el medio) debido a que el host hacker se posiciona entre medio en el camino de red de la víctima y el router, pudiendo pasar desapercibido.



Si bien hoy en día la mayor parte del tráfico común en internet manejan intercambio de datos de forma encriptada (por lo que el hacker no podría extraer información de la carga útil del paquete), hay casos en los que no es así y se podría inspeccionar en texto simple toda la información que se transmitió entre la víctima y un servidor. Por ello siempre se debe asegurar de no introducir información sensible en una página sin certificados de seguridad (sin HTTPS).

Podemos ver esto en un ejemplo utilizando el software WireShark el cual es un analizador de paquetes de red, una utilidad que captura todo tipo de información que pasa a través de una conexión. Wireshark es gratis y de código abierto, y se puede usar para diagnosticar problemas de red, efectuar auditorías de seguridad y aprender más sobre redes informáticas.

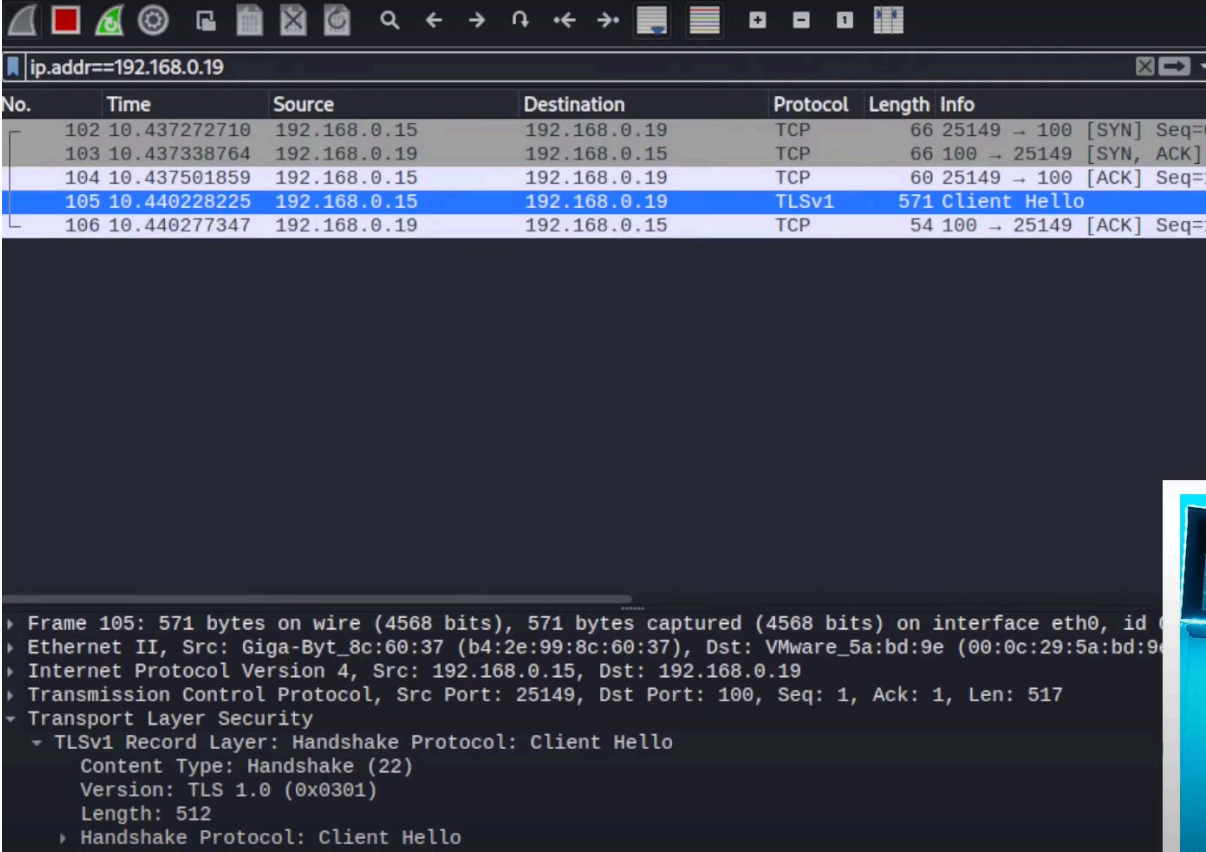
Podemos filtrar por una dirección IP para ver los paquetes que se envían a esta, en este caso un paquete que utiliza el protocolo HTTP

The image shows a Wireshark network traffic capture on the interface *eth0. The filter bar at the top is set to 'ip.addr==192.168.0.19'. The packet list shows several TCP and HTTP packets. Packet 337 is highlighted, showing an HTTP POST request to /login. The packet details pane below shows the structure of the captured frame, including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The HTTP body is HTML Form URL Encoded with the following data:

No.	Time	Source	Destination	Protocol	Length	Info
332	136.649730818	192.168.0.15	192.168.0.19	TCP	66	25059 → 100 [SYN] Seq: 100
333	136.649805488	192.168.0.19	192.168.0.15	TCP	66	100 → 25059 [SYN, ACK] Seq: 100
334	136.649997037	192.168.0.15	192.168.0.19	TCP	60	25059 → 100 [ACK] Seq: 100
335	136.650241644	192.168.0.15	192.168.0.19	TCP	198	25059 → 100 [PSH, ACK] Seq: 100
336	136.650261141	192.168.0.19	192.168.0.15	TCP	54	100 → 25059 [ACK] Seq: 100
337	136.650241774	192.168.0.15	192.168.0.19	HTTP	84	POST /login HTTP/1.1
338	136.650353669	192.168.0.19	192.168.0.15	TCP	54	100 → 25059 [ACK] Seq: 100

Frame 337: 84 bytes on wire (672 bits), 84 bytes captured (672 bits) on interface eth0, id 0
 Ethernet II, Src: Giga-Byt_8c:60:37 (b4:2e:99:8c:60:37), Dst: VMWare_5a:bd:9e (00:0c:29:5a:bd:9e)
 Internet Protocol Version 4, Src: 192.168.0.15, Dst: 192.168.0.19
 Transmission Control Protocol, Src Port: 25059, Dst Port: 100, Seq: 145, Ack: 1, Len: 30
 [2 Reassembled TCP Segments (174 bytes): #335(144), #337(30)]
 Hypertext Transfer Protocol
 HTML Form URL Encoded: application/x-www-form-urlencoded
 Form item: "username" = "mario"
 Form item: "password" = "123123"

Esto no es ningún tipo de ataque MITM (Man in the Middle), aquí solo estamos escuchando y viendo nuestra red. El protocolo HTTP envía los paquetes en texto plano, lo cual lo hace muy inseguro y fácil de interceptar, por lo cual recomendamos HTTPS. Ahora veamos el mismo paquete pero enviado con este otro protocolo.



The image shows a Wireshark packet capture window with the filter 'ip.addr==192.168.0.19'. The packet list shows five packets. Packet 105 is highlighted, showing a TLSv1 Client Hello. The packet details pane shows the structure of the Client Hello, including the Handshake Protocol, Content Type, Version (TLS 1.0), and Length (512).

No.	Time	Source	Destination	Protocol	Length	Info
102	10.437272710	192.168.0.15	192.168.0.19	TCP	66	25149 → 100 [SYN] Seq=
103	10.437338764	192.168.0.19	192.168.0.15	TCP	66	100 → 25149 [SYN, ACK] Seq=
104	10.437501859	192.168.0.15	192.168.0.19	TCP	60	25149 → 100 [ACK] Seq=
105	10.440228225	192.168.0.15	192.168.0.19	TLSv1	571	Client Hello
106	10.440277347	192.168.0.19	192.168.0.15	TCP	54	100 → 25149 [ACK] Seq=

```

Frame 105: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) on interface eth0, id 0
Ethernet II, Src: Giga-Byt_8c:60:37 (b4:2e:99:8c:60:37), Dst: VMware_5a:bd:9e (00:0c:29:5a:bd:9e)
Internet Protocol Version 4, Src: 192.168.0.15, Dst: 192.168.0.19
Transmission Control Protocol, Src Port: 25149, Dst Port: 100, Seq: 1, Ack: 1, Len: 517
Transport Layer Security
  TLSv1 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 512
    Handshake Protocol: Client Hello

```

Podemos ver que el paquete se recibe pero a simple vista no se puede acceder a lo que tiene dentro, ya que este se encuentra encriptado bajo el protocolo HTTPS.

Conclusión

Si bien las redes de computadoras nos han brindado y nos van a seguir brindando conectividad y facilidad en lo cotidiano de nuestras vidas, también pueden exponer datos sensibles sobre nosotros si no tomamos las precauciones necesarias. Si tomamos las medidas de seguridad necesarias, que suelen ser simples como una buena contraseña o verificar que un sitio web use HTTPS, vamos a impedir muchísimos de estos ataques de personas desconocidas.

Bibliografía

- [1] Wikipedia. (n.d.). **Escáner de puertos**. Wikipedia.
https://es.wikipedia.org/wiki/Esc%C3%A1ner_de_puertos
- [2] Erickson, J. (2008). **Hacking: The art of exploitation** (2nd ed.).
[https://repo.zenk-security.com/Magazine%20E-book/Hacking-%20The%20Art%20of%20Exploitation%20\(2nd%20ed.%202008\)%20-%20Erickson.pdf](https://repo.zenk-security.com/Magazine%20E-book/Hacking-%20The%20Art%20of%20Exploitation%20(2nd%20ed.%202008)%20-%20Erickson.pdf)
- [3] Avast. (n.d.). **What is port scanning?** Avast.
<https://www.avast.com/es-ar/business/resources/what-is-port-scanning#pc>
- [4] Fortinet. (n.d.). **Internet Control Message Protocol (ICMP)**. Fortinet.
<https://www.fortinet.com/lat/resources/cyberglossary/internet-control-message-protocol-icmp>
- [5] Ciphersafety. (n.d.). **Port scanning: Control de puertos**. Ciphersafety.
<https://ciphersafety.com/port-scanning-control-puertos/>
- [6] Fortinet. (n.d.). **What is port scanning?** Fortinet.
<https://www.fortinet.com/lat/resources/cyberglossary/what-is-port-scan#:~:text=Un%20escaneo%20de%20puertos%20es,est%C3%A1n%20recibiendo%20o%20enviando%20datos>
- [7] Avast (2020) ¿Que es un Sniffer?
<https://www.avast.com/es-es/c-sniffer>
- [8] Redes Zone. (n.d.). **Suplantación de ARP: qué es y cómo afecta a nuestra red:**
<https://www.redeszone.net/tutoriales/redes-cable/ataques-arp-spoofing-evitar/>