

3. Protección de Datos

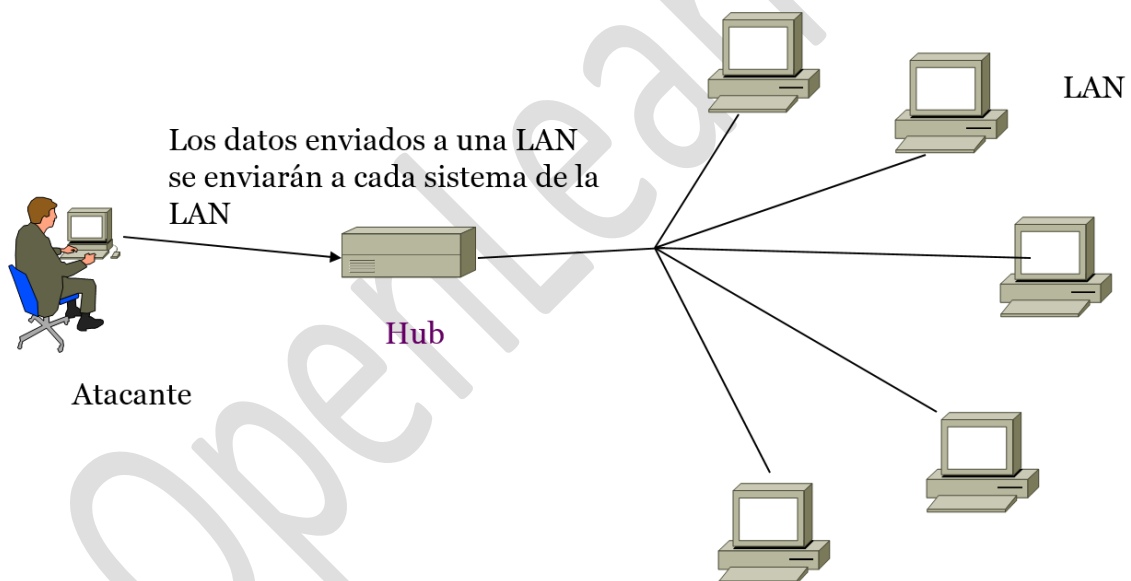
1. Sniffing de Red

1.1 Introducción

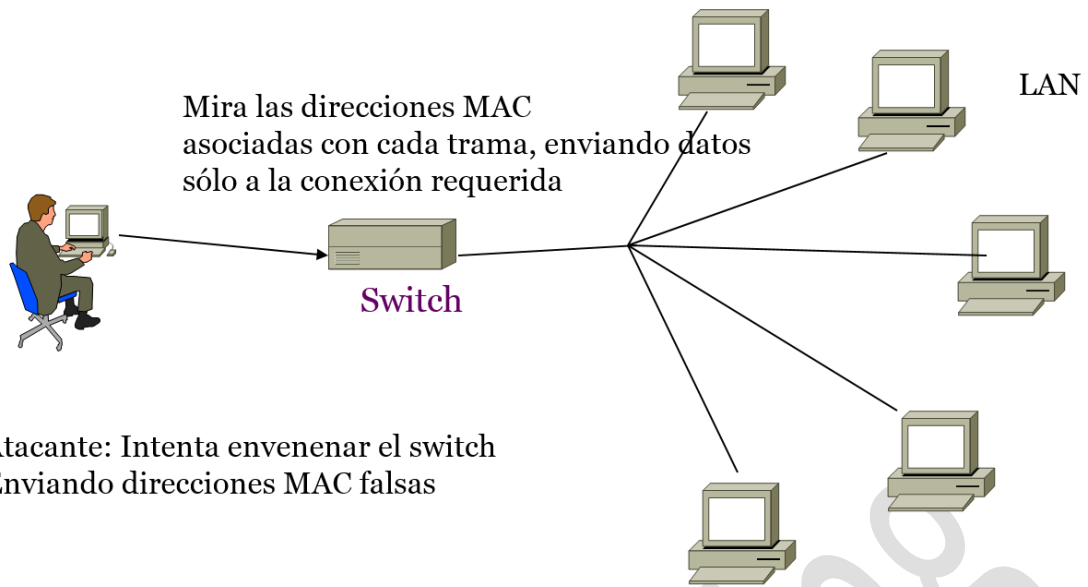
Un sniffer, o más concretamente, un sniffer de paquetes, se definen como una pieza de software o hardware que se conecta a una red informática y supervisa todo el tráfico que pasa por el cable. Al igual que los dispositivos de intervención de teléfonos que usan las autoridades para escuchar conversaciones de otras personas, un programa de sniffing permite a alguien escuchar las conversaciones entre ordenadores que fluyen por las redes.

Las conversaciones entre ordenadores consisten en, aparentemente, datos binarios aleatorios. Por lo tanto, los programas de intervención necesitan disponer de una característica denominada "análisis de protocolo", la cual permite decodificar el tráfico enviado y darle sentido para hacerlo de "alguna manera" legible.

En tecnologías compartidas (usando hubs) es muy sencillo capturar todo el tráfico que pasa por la red, ya todos los paquetes que se envían llegan a todos los ordenadores conectados a esa red.



Sin embargo, hoy en día es muy raro encontrar este tipo de redes, que hace tiempo que fueron sustituidas por tecnologías conmutadas (switches), donde la captura directa ya no es posible.



Aun así, existen técnicas de sniffing usadas en redes conmutadas que veremos en este mismo tema y que pueden servir para saber si nuestros datos viajan de una manera segura.

OpenLearning

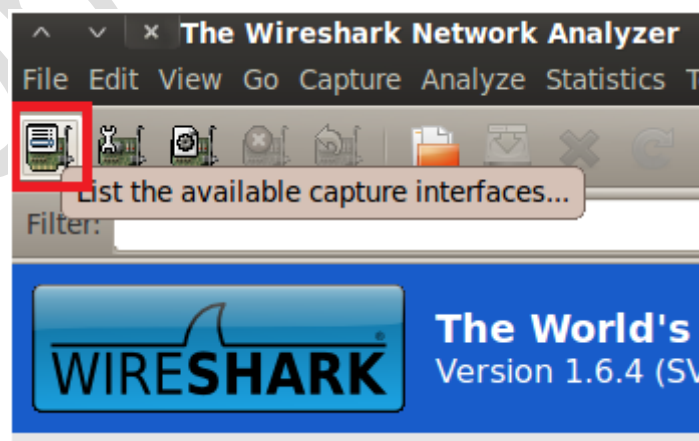
1.2 Sniffers

Tenemos a nuestra disposición una gran cantidad de sniffers:

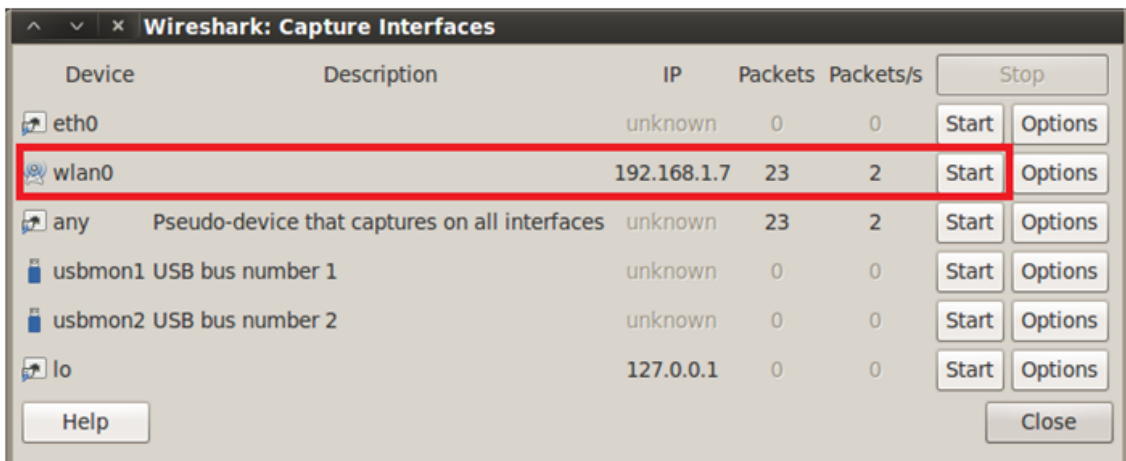
- Wireshark (Ethereal)
- Dsniff
- Sniffit
- Aldebaran
- Hunt
- NGSSniff
- Ntop
- pf
- IPTraf
- Etherape
- Netfilter
- Network Probe
- Maa Tec Network Analyzer
- Snort
- Macof, MailSnarf, URLSnarf, WebSpy
- Windump
- Etherpeek
- Ettercap
- SMAC
- Mac Changer
- Iris
- NetIntercept
- WinDNSSpoof

Y entre todas ellas destaca sin duda Wireshark. Se trata de una aplicación gratuita y multiplataforma. Veamos el funcionamiento básico de Wireshark:

Una vez instalado (en Kali Linux ya viene instalado), lo abrimos y presionamos sobre el icono marcado en rojo en la imagen:



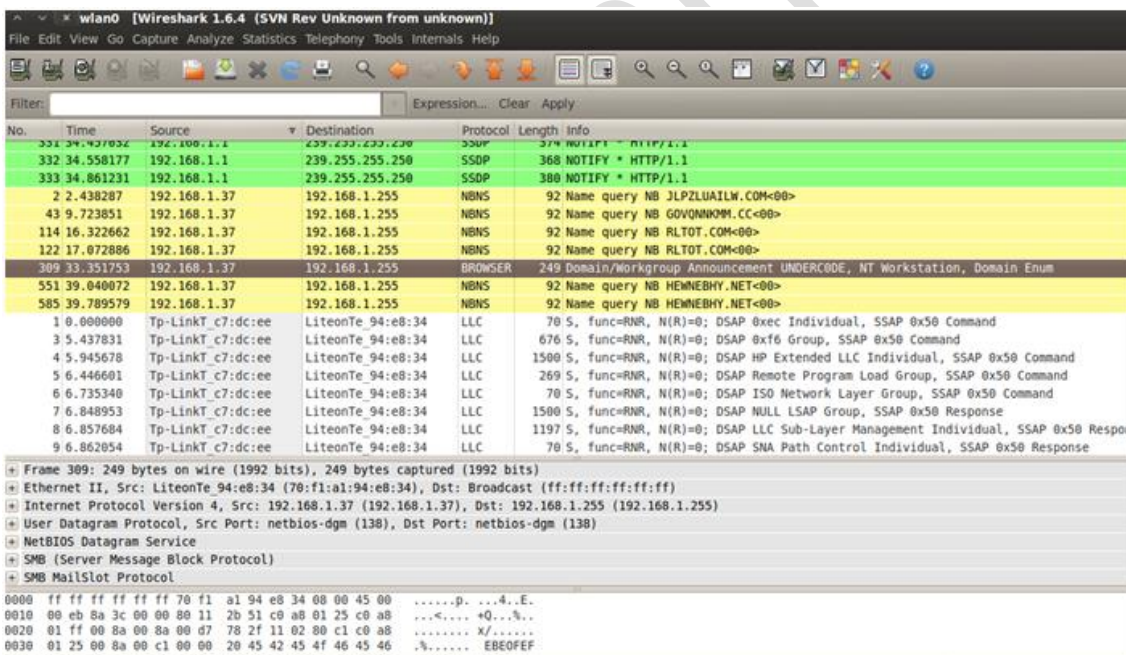
Esto nos permitirá seleccionar nuestra tarjeta de red que pondremos a la escucha de paquetes.



Para saber que tarjeta poner a la escucha, debemos observar cual es la que recibe paquetes.

Se puede observar en la imagen que en este caso es la wlan0. Una vez identificada, damos en Start para comenzar.

Automáticamente el programa comenzara a capturar paquetes de todos los hosts conectados a la red.



Donde podemos ver de la IP origen y destino entre las que se mueven los paquetes y el protocolo usado. Además de esto podemos ver el contenido del paquete.

Si observamos la imagen, hay una caja de texto llamada Filter.



Esa caja de texto, como bien dice su nombre, permite filtrar paquetes. Y ahora veremos algunos de los filtros que posee Wireshark para que podamos usar este sniffer de una forma más eficiente.

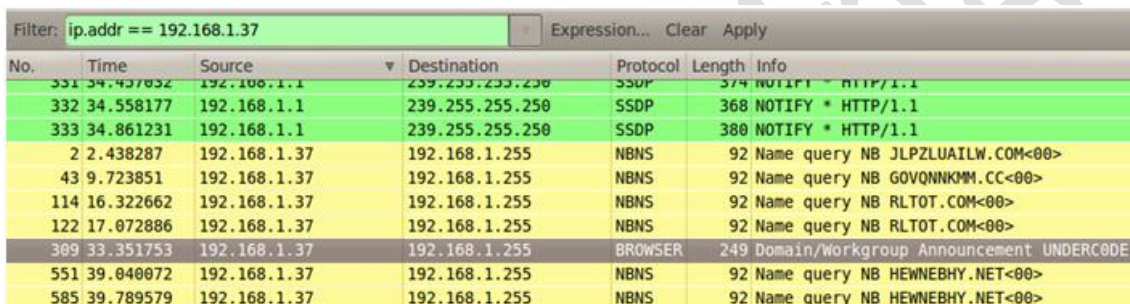
En el filtro se pueden usar operadores lógicos como los siguientes:

- == (Igual que)
- > (Mayor que)
- < (Menor que)
- != (Distinto que)
- >= (Mayor o igual que)
- <= (Menor o igual que)

Algunos filtros de ejemplo son (en lugar de 0.0.0.0 pondríamos la IP a filtrar):

- ip.addr == 0.0.0.0
- ip.addr == 0.0.0.0 && ip.addr == 0.0.0.0 (Para filtrar más de una IP)
- ip.addr == 0.0.0.0 || ip.addr == 0.0.0.0 (Para filtrar una IP de cualquiera de las dos)

Veremos un ejemplo con uno de los filtros: ip.addr == 192.168.1.37



No.	Time	Source	Destination	Protocol	Length	Info
331	34.437852	192.168.1.1	239.255.255.250	SSDP	374	NOTIFY * HTTP/1.1
332	34.558177	192.168.1.1	239.255.255.250	SSDP	368	NOTIFY * HTTP/1.1
333	34.861231	192.168.1.1	239.255.255.250	SSDP	380	NOTIFY * HTTP/1.1
2	2.438287	192.168.1.37	192.168.1.255	NBNS	92	Name query NB JLPZLUAILW.COM<00>
43	9.723851	192.168.1.37	192.168.1.255	NBNS	92	Name query NB GOVQNKMM.CC<00>
114	16.322662	192.168.1.37	192.168.1.255	NBNS	92	Name query NB RLTOT.COM<00>
122	17.072886	192.168.1.37	192.168.1.255	NBNS	92	Name query NB RLTOT.COM<00>
309	33.351753	192.168.1.37	192.168.1.255	BROWSER	249	Domain/Workgroup Announcement UNDERCODE
551	39.040072	192.168.1.37	192.168.1.255	NBNS	92	Name query NB HEWNEBHY.NET<00>
585	39.789579	192.168.1.37	192.168.1.255	NBNS	92	Name query NB HEWNEBHY.NET<00>

En este caso me debería mostrar los paquetes correspondientes a la ip 192.168.1.37

Este filtro es muy poderoso, y veremos su potencial cuando filtremos por protocolo.

Algunos de los filtros son estos: tcp, http, pop, dns, arp, ssl, etc.

Con un Sniffer podemos obtener datos muy importantes. Desde cookies hasta usuarios y contraseñas. A modo ejemplo, abrimos una sesión un FTP y veremos lo que hace nuestro Wireshark:

The image shows a Wireshark network traffic capture filtered for the FTP protocol. The main pane displays a list of packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Two packets are highlighted with red boxes: packet 6269 (Request: USER) and packet 6273 (Request: PASS). The packet details pane below shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and File Transfer Protocol (FTP) fields. The hex dump at the bottom shows the raw data of the selected packet, with the password '11111111' visible in hexadecimal.

No.	Time	Source	Destination	Protocol	Length	Info
6267	574.823962	190.196.69.216	192.168.1.7	FTP	386	Response: 220----- Welcome to P
6269	574.824219	192.168.1.7	190.196.69.216	FTP	81	Request: USER
6271	574.894166	190.196.69.216	192.168.1.7	FTP	107	Response: 331 User cristonu OK. Pass
6272	574.894296	192.168.1.7	190.196.69.216	FTP	83	Request: PASS
6273	574.995661	190.196.69.216	192.168.1.7	FTP	109	Response: 230 OK. Current restricted
6274	574.995825	192.168.1.7	190.196.69.216	FTP	72	Request: SYST
6275	575.084639	190.196.69.216	192.168.1.7	FTP	85	Response: 215 UNIX Type: L8
6276	575.084805	192.168.1.7	190.196.69.216	FTP	72	Request: FEAT
6277	575.230274	190.196.69.216	192.168.1.7	FTP	285	Response: 211-Extensions supported:
6278	575.246134	192.168.1.7	190.196.69.216	FTP	71	Request: PWD
6279	575.346222	190.196.69.216	192.168.1.7	FTP	100	Response: 257 "/" is your current lo
6280	575.347971	192.168.1.7	190.196.69.216	FTP	74	Request: TYPE I
6281	575.438497	190.196.69.216	192.168.1.7	FTP	96	Response: 200 TYPE is now 8-bit bina
6282	575.438668	192.168.1.7	190.196.69.216	FTP	72	Request: PASV
6283	575.704113	192.168.1.7	190.196.69.216	FTP	72	[TCP Retransmission] Request: PASV
6285	575.859590	190.196.69.216	192.168.1.7	FTP	118	[TCP Retransmission] Response: 227 E
6286	575.860609	192.168.1.7	190.196.69.216	FTP	72	Request: MLSD
6291	576.045100	190.196.69.216	192.168.1.7	FTP	96	Response: 150 Accepted data connecti

Frame 6272: 83 bytes on wire (664 bits), 83 bytes captured (664 bits)
Ethernet II, Src: Micro-St 9a:b3:1f (00:19:db:9a:b3:1f), Dst: Tp-LinkT c7:dc:ee (d8:5d:4c:c7:dc:ee)
Internet Protocol Version 4, Src: 192.168.1.7 (192.168.1.7), Dst: 190.196.69.216 (190.196.69.216)
Transmission Control Protocol, Src Port: 56672 (56672), Dst Port: ftp (21), Seq: 16, Ack: 362, Len: 17
File Transfer Protocol (FTP)

```
0000 d8 5d 4c c7 dc ee 00 19 db 9a b3 1f 08 00 45 00 .]L.....E.
0010 00 45 58 90 40 00 00 06 1b d7 c0 a8 01 07 be c4 .EX.@. ....
0020 45 d8 dd 60 00 15 2e e3 7c 69 36 a9 92 24 80 18 E..'....|i6..$.
0030 00 7b b9 00 00 00 01 01 08 0a 00 05 25 08 05 b6 .{.....%...
0040 1f 22 50 41 53 53 20 35 46 66 34 6d 70 51 38 6b ."PASS % #f#p#k
0050 6a 0d 0a j..
```

Como se puede ver, filtramos el protocolo FTP y Wireshark capturó el usuario y contraseña del FTP.

1.3 ARP Spoofing

ARP resuelve direcciones IP a direcciones MAC del interfaz para enviar datos.

Los paquetes ARP se pueden construir para enviar datos a la máquina del atacante.

Un atacante puede explotar ARP Poisoning para interceptar tráfico de red entre dos máquinas de la red.

Inundando la tabla ARP de un switch con respuestas falsas ARP, permite a un atacante sobrecargar los switches y después esnifar la red mientras el switch está en modo "hub".

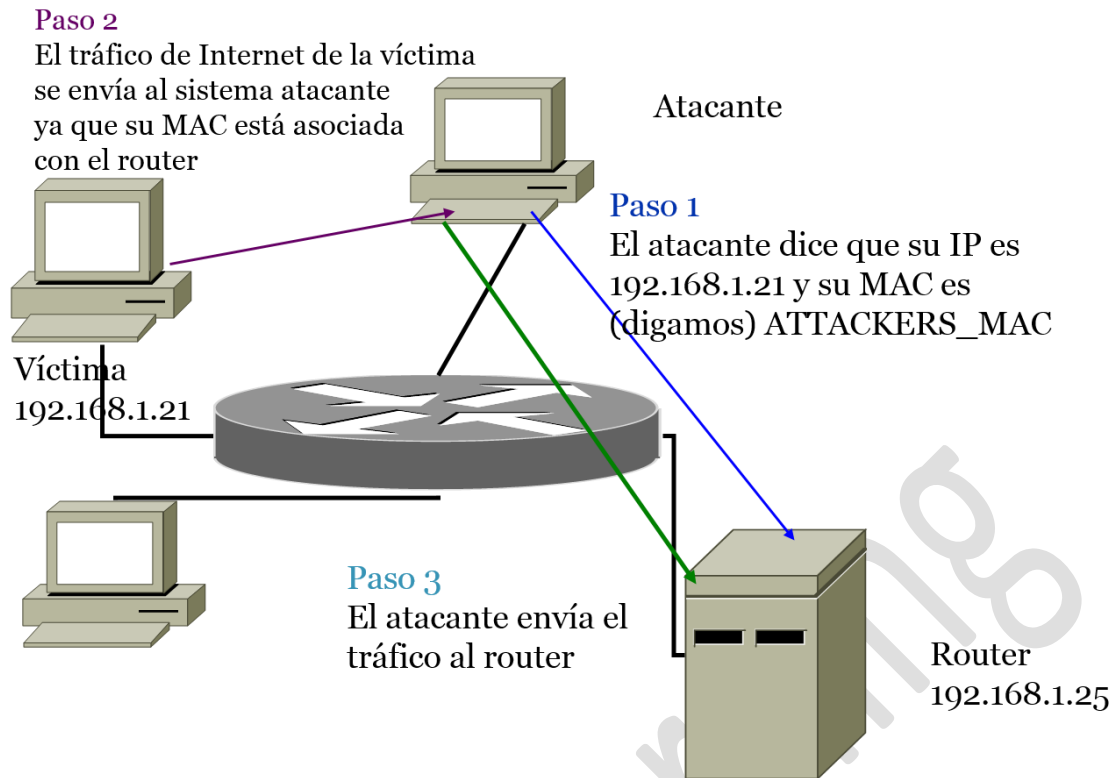
Los desarrolladores de ARP vivían en un mundo mucho más confiable que el de hoy día, por lo que hicieron simple este protocolo. El problema es que este diseño simple hace posible el ARP poisoning. Cuando se envía una petición ARP, el sistema simplemente confía en que la respuesta ARP viene del dispositivo correcto. ARP no proporciona ninguna forma de verificar que el dispositivo que responde es realmente quien dice ser. ARP es tan confiado que muchos sistemas operativos aceptan respuestas ARP, incluso si no han hecho ninguna petición. Para reducir la cantidad de tráfico ARP se implementa la llamada caché ARP. Podemos ver nuestra caché ARP con el comando arp -a.

```
Interfaz: 192.168.1.36 --- 0xa
Dirección de Internet      Dirección física      Tipo
192.168.1.1                d0-ae-ec-f9-b6-6c    dinámico
192.168.1.254             90-f6-52-b6-9c-d6    dinámico
192.168.1.255             ff-ff-ff-ff-ff-ff    estático
224.0.0.22                 01-00-5e-00-00-16    estático
224.0.0.252                01-00-5e-00-00-fc    estático
239.255.255.250           01-00-5e-7f-ff-fa    estático
255.255.255.255           ff-ff-ff-ff-ff-ff    estático

Interfaz: 192.168.127.1 --- 0x10
Dirección de Internet      Dirección física      Tipo
192.168.127.255           ff-ff-ff-ff-ff-ff    estático
224.0.0.22                 01-00-5e-00-00-16    estático
224.0.0.252                01-00-5e-00-00-fc    estático
239.255.255.250           01-00-5e-7f-ff-fa    estático

Interfaz: 192.168.236.1 --- 0x11
Dirección de Internet      Dirección física      Tipo
192.168.236.255           ff-ff-ff-ff-ff-ff    estático
224.0.0.22                 01-00-5e-00-00-16    estático
224.0.0.252                01-00-5e-00-00-fc    estático
239.255.255.250           01-00-5e-7f-ff-fa    estático
```

El método de ARP Poisoning implica el envío de peticiones o respuestas ARP falsas al switch y otros dispositivos para intentar redirigir el tráfico de los sistemas que queremos snifar. Los paquetes ARP falsos se almacenarán en el switch y en el resto de dispositivos que reciban los paquetes. La dirección MAC que suele falsearse es la del router de forma que el atacante pueda capturar todo el tráfico saliente.



Tenemos un vídeo demostrativo de esta técnica y otra más avanzada llamada Man In The Middle (MITM).