

Seguridad ante ataques MiM (Man in the Middle)



By Pablo E. Bullian
pablo [dot] bullian [at] gmail [dot] com

Finalidad de los ataques

- Estos ataques se utilizan para “sniffear” el tráfico de la red, logrando obtener información privada, como usuarios y contraseñas entre otras cosas.
- En una red con hubs, no hay necesidad, se podría sniffear paquetes sin problemas, pero en redes con switches se procede con estos ataques
- Funcionan algunos ataques sobre la base técnica de que cada paquete e al red se direcciona gracias a la dirección física o dirección MAC de cada PC

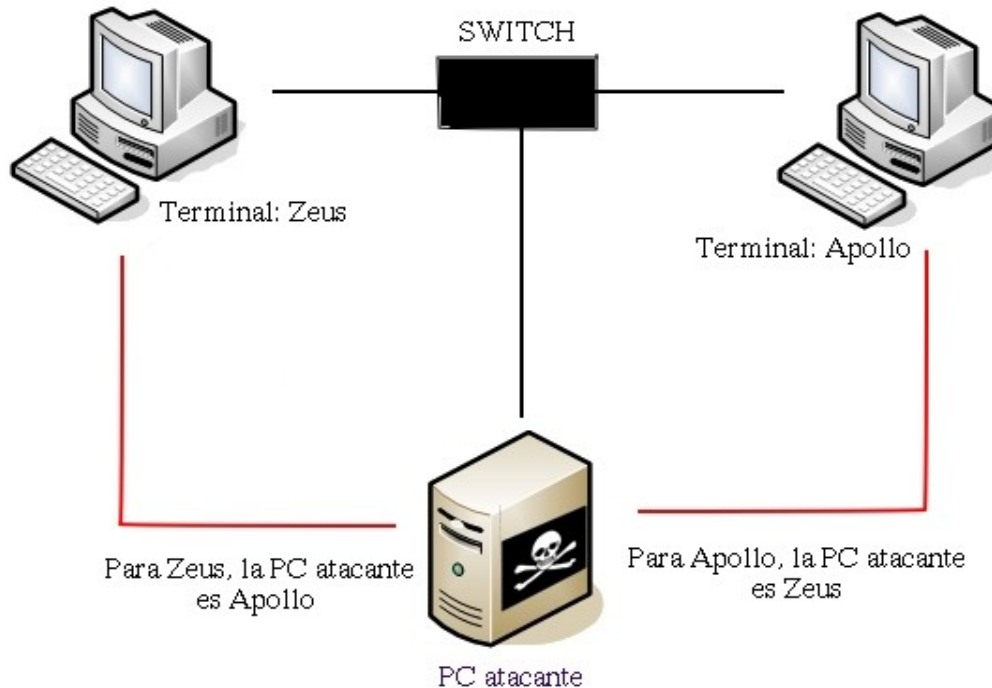
preambulo	S. O. F.	Mac del Destino	Mac del origen	tipo	Payload	C R C
-----------	----------------	-----------------	----------------	------	---------	-------------

(Esquema de paquete ethernet y 802.3)

Tipos de ataques MiM

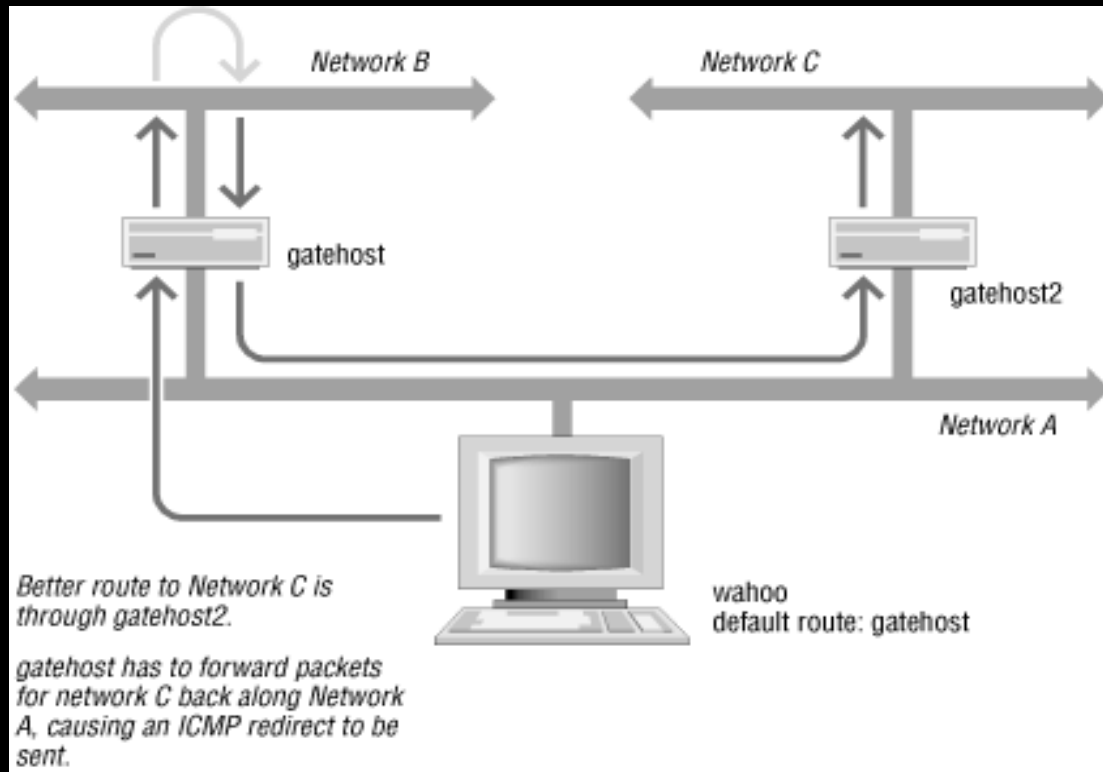
- **ARP Poisoning**
- **ICMP Redirection**
- **DHCP Spoofing**
- **Port Stealing**

ARP Poisoning



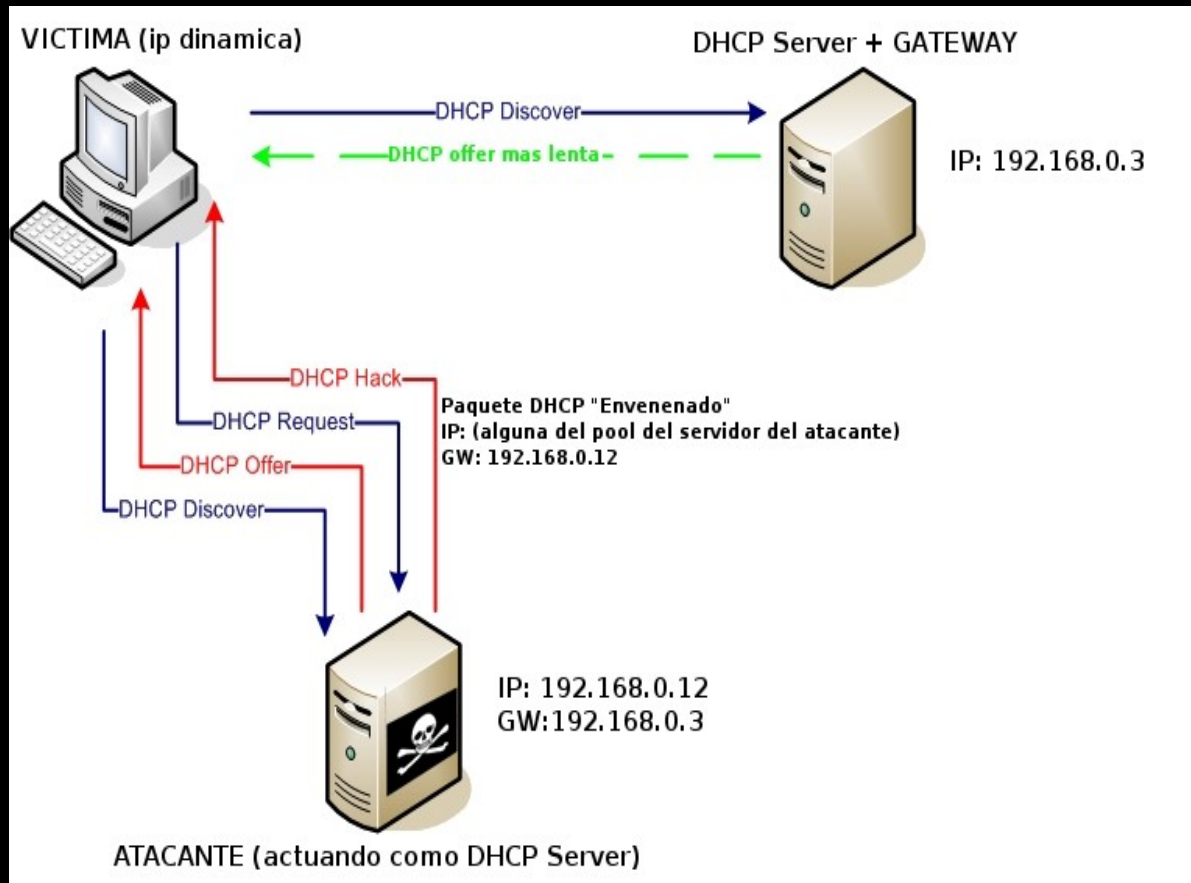
- La PC del atacante envia paquetes ARP request/replies para que las PCs reescriban sus tablas.
- Ahora el trafico entre las dos PCs lo “tramitamos” nosotros, con lo cual, podemos analizarlo y obtener datos sensibles.

ICMP Redirection



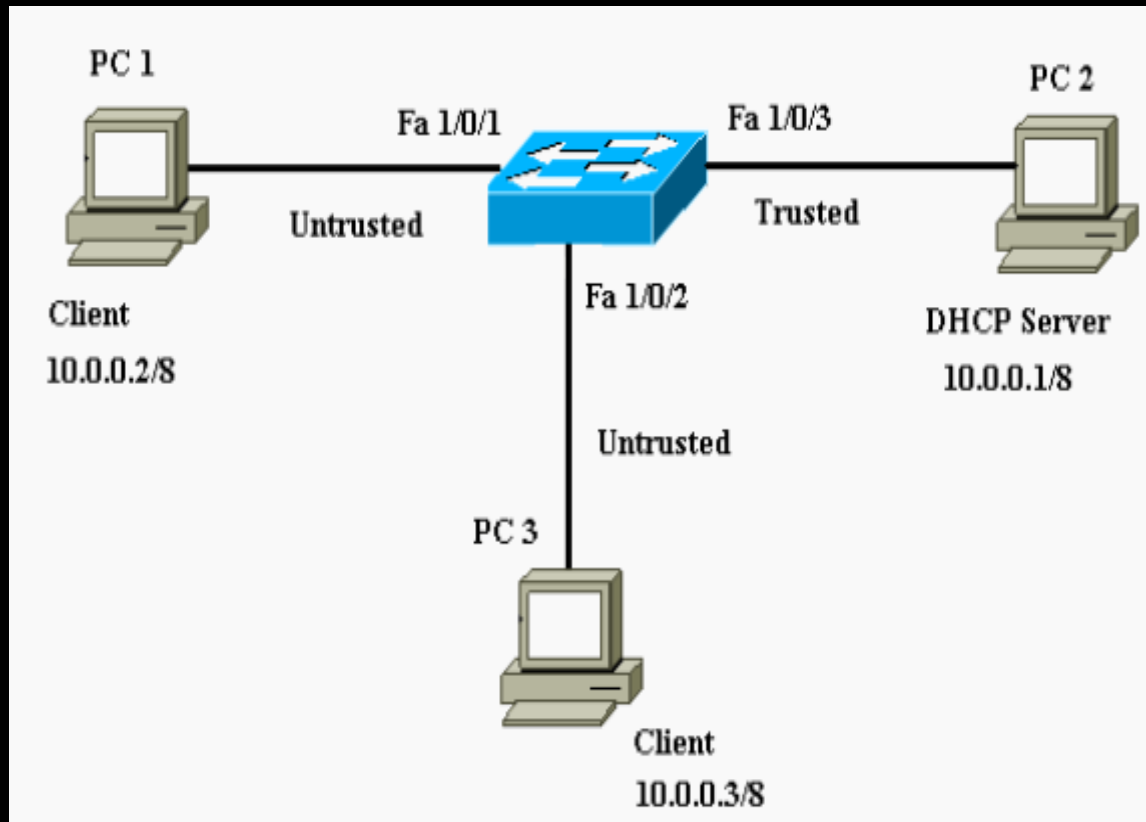
- Los paquetes ICMP redirection les dicen a las Pcs cual es la mejor salida.
- Todo el trafico hacemos que lo enruten a traves de nosotros, y nosotros lo enrutamos a donde debe ir
- Tambien existe en algunos OS que aceptan el ICMP Router Advertisements

DHCP Spoofing



- Las Pcs envian un DHCP DISCOVER
- EL atacante trata de ganar la carrera contra el servidor DHCP
- La respuesta del atacante hace que el quede como gateway por default de la pc, enrutando el trafico hacia el

Port Stealing



- La idea es llenar el puerto de paquetes bobos con el 'source address' de la pc victima
- El switch reescribe su CAM y los proximos paquetes a la victima, van a nuestro puerto, los capturamos en modo promiscuo
- Para devolver los paquetes debemos enviar un ARP request, que la victima conteste y tome su puerto nuevamente

Ettercap

Un programa sniffer y filtrador de contenido para ataques Man in the Middle.

Características:

- SSH1 support
- SSL support
- Packet filtering/dropping
- Remote traffic sniffing through tunnels and route mangling
- Plug-ins support
- Password collector
- Passive OS fingerprint
- hosts in the LAN
- Kill a connection

Ettercap

Tiene dos tipos de Sniffing:

*Unificado

Hace un sniff de todo paquete que pase por el cable, puede estar la placa en promiscua o no, y puede haber algun tipo de ataque corriendo o no.

Este tipo de Sniffing, desabilita el forwarding del kernel, para activar el propio, ademas como solo escucha en una interfaz, no puede usarse en un Gateway sin activar la funcion Unoffensive, ya que no re-routearia los paquetes a la otra interfaz.

*Bridged

Funciona con dos placas, se puede pensar como un enrutamiento de capa 1, somos una pc entre un cable, es totalmente "inocuo" ya que no hay forma de detectar este ataque. No lo podemos utilizar en Gateways.

Collected passive profiles:

```
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
192.168.0.6
192.168.0.7
192.168.0.8
192.168.0.12
192.168.0.13
192.168.0.15
192.168.0.15
192.168.0.22
192.168.0.60
192.168.0.62
192.168.0.99
192.168.0.99
192.168.0.103
192.168.0.106
192.168.0.107
192.168.0.108
192.168.0.109
192.168.0.110
192.168.0.111
192.168.0.112
192.168.0.115
192.168.0.136
192.168.0.150
192.168.0.151
192.168.0.152
192.168.0.153
192.168.0.154
192.168.0.155
192.168.0.156
```

User messages:

```
HTTP : 200.00.00.00:80 -> USER: admin PASS: admin INFO: 200.00.00.00/soporte/redled.gif
HTTP : 200.00.00.00:80 -> USER: admin PASS: admin INFO: 200.00.00.00/soporte/swdown.gif
HTTP : 200.00.00.00:80 -> USER: admin PASS: admin INFO: 200.00.00.00/soporte/swanillo2.php
POP : 200.00.00.00:110 -> USER: admin PASS: admin
POP : 200.00.00.00:110 -> USER: admin PASS: admin
```

Ettercap Logs

- m (message user) : Guarda todos los mensajes que salen al usuario, en el caso de la interfaz ncurses, es la tabla que aparece abajo de todo.
- L (log de paquetes) : Guarda todos los paquetes capturados en un formato binario que se puede abrir con etterlog.
- l (log de paquetes "utiles"): solo guarda paquetes en formato binario que contengan informacion de usuarios y passwords
- O / -o (solo remotos/solo locales) : guarda paquetes, solo de host remotos o solo de host locales

Ataque a SSH

El ataque (forma remota) funciona de la siguiente manera:

el cliente envia el paquete SYN al server de SSH

cliente ==GW==**atacante**==> server

el server responde, pero como es victima de algun ataque MiM, el trafico se da asi:

Server == **Atacante**==GW==>cliente

Llegamos al paso de que el server le envia los datos de la conexion, para poder obtener passwords y demas, se debe "hablar" en SSH v1, para eso el atacante modifica el paquete que envia el servidor, haciendo un downgrade de SSH v2 a ofrecerle v1 al cliente, si este es el caso.

Server [v2] == [v2] Atacante [v1] == [v1] cliente

Ataque a SSH

y vamos a obtener algo asi:

```
Sniffing (ARP based) : ANY:0 <--> 10.1.1.112 <--> 10.1.1.1:0

TCP + UDP packets... (default)

Collecting passwords...

19:49:45  10.1.1.111:3927 <--> 216.133.72.171:22          SSH decryp
t

USER: iamhacked
PASS: mypasswordislame
```

Prevencion y Deteccion

- **Armar rutas ARP estaticas PARA TODO en las PC**

-Se sabe por pruebas que algunos OS de la familia Windows, responden cambiando sus tablas a los broadcast ARP por mas que esten fijas

-No es la solucion unica, ya que en el caso de SSH remoto podemos fijarnos la MAC en la pc a nuestro gateway, pero el atacante puede estar en la red del servidor.

-Podemos sufrir port stealing

IP address	HW type	Flags	HW address	Mask	Device
192.168.0.1	0x1	0x2	00:17:9A:59:5B:85	*	eth1

Prevencion y Deteccion

En el caso puntual de SSH:

- No debemos aceptar una nueva key fingerprint si ya habiamos guardado la anterior, ante un mensaje de estos debemos cancelar toda accion y revisar.

```
pblaptop:/home/pablo# ssh 200.69.1.2
The authenticity of host '200.69.1.2 (200.69.1.2)' can't be established.
RSA key fingerprint is 90:f5:a4:83:7e:70:ee:68:a1:e1:94:a3:5:5:5:5.
Are you sure you want to continue connecting (yes/no)? █
```

- Anexado a esto, debemos validar de forma local las fingerprint del servidor.
- Si estamos en una red local, debemos mantener rutas estaticas (aunque como vimos no es la solucion, pero nos ayuda)
- Ante alertas acerca de cambios en certificados SSL, debemos cancelar la accion y revisar.

Prevencion y Deteccion

- **Mac binding en los Switches**

- Solo los switchs de alta gama tienen esta capacidad.

- Tiene que estar asociado con la anterior medida de prevencion.

- Obviamente, debemos tener toda la red conectada por switchs.

Prevencion y Deteccion

- **Tener herramientas para la deteccion en caso de cambios en las tablas ARP**

- ARPCWATCH

- Envia mail y utiliza el syslog, para alertar en cambios, o paquetes broadcast ARP enviados en la red.

- ARPCALERT

- Previene conexiones a la red de MACs que no esten autorizadas y puede correr scripts al detectarlas

- SNORT

- El tan famoso “programa de deteccion de intrusos”. Basicamente este programa, analiza el trafico de la red y se basa en reglas configuradas para analizarlo.

Prevencion y Deteccion

- **Usar ciertos plugins del ETTERCAP para deteccion**

- find_ettercap

- Trata de identificar “paquetes ettercap” en la red. Se basa en numeros de identificacion o secuencia, por lo tanto no es muy fiable.

- search_promisc

- Envia dos tipos de paquetes ARP request malformados a cada uno de los host, si un host responde, es mas o menos posible que su placa este en promiscua, no es seguro pero nos da una idea de quienes podrian estarlo.

Prevencion y Deteccion

- **Usar ciertos plugins del ETTERCAP para deteccion**

- scan_poisoner

- De una lista de hosts chequea si dos pc tiene la misma MAC, luego envia paquetes icmp echo, si la MAC de la respuesta difiere de la IP que tenemos en nuestra lista podemos estar frente a una pc que esta haciendo un forwarding de ese paquete que va a la pc infectada devuelta hacia nosotros.

Consultas

Fragmento de las man pages del Ettercap:

"Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning." - Rich Cook

By Pablo E. Bullian

pablo [dot] bullian [at] gmail [dot] com

www.coffeeattack.com.ar