**Detección de Tarjetas Promiscuas en Red LAN**

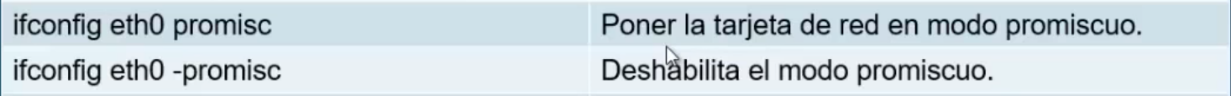
Ernesto Velasco Soler

## Funcionamiento general de una tarjeta en modo promiscuo

Para capturar el tráfico de la red a la que estamos conectados hemos de “situar” la tarjeta de red en modo promiscuo ¿Y qué es esto del modo promiscuo?, por definición todos los adaptadores de red reciben los paquetes que son para ellos, (filtran por IP), pero el colocar el adaptador de red en modo promiscuo hace que no filtre, y vea todo el tráfico que está en la red.

Los sniffers funcionan por una sencilla razón: muchos de los protocolos de acceso remoto a las máquinas se transmiten las claves de acceso como texto plano y, por lo tanto, capturando la información que se transmite por la red se puede obtener este tipo de información y el acceso ilegítimo a una determinada máquina.

## Activar modo promiscuo en una tarjeta de red (Linux)



## Capturar tráfico que circula por una interfaz

“tcpdump -i eth0 host 10.0.0.1” captura el tráfico proveniente de un host

“tcpdump -i eth0 net 192.168.1.0/24” captura el tráfico proveniente de una red

## Técnicas de detección de tarjetas promiscuas en sistemas

### • El test DNS

En este método, la herramienta de detección en sí misma está en modo promiscuo.

Creamos numerosas conexiones TCP falsas en nuestro segmento de red, esperando

un sniffer pobremente escrito para atrapar estas conexiones y resolver la dirección

IP de los inexistentes hosts.

Algunos sniffers realizan búsquedas inversas DNS en los paquetes que capturan.

Cuando se realiza una búsqueda inversa DNS, una utilidad de detección de sniffers

"huele" la petición de las operaciones de búsqueda para ver si el objetivo es aquel

que realiza la petición del host inexistente.

### • El Test ICMP

Ping de Latencia.

En este método, hacemos ping al blanco y anotamos el Round Trip Time (RTT,

retardo de ida y vuelta o tiempo de latencia).

Creamos centenares de falsas conexiones TCP en nuestro segmento de red en un

período de tiempo muy corto. Esperamos que el sniffer esté procesando estos

paquetes a razón de que el tiempo de latencia incremente.

Entonces hacemos ping otra vez, y comparamos el RTT esta vez con el de la primera vez. Después de una serie de tests y medias, podemos concluir o no si un sniffer está realmente funcionando en el objetivo o no.

### • El test ARP

Podemos enviar una petición ARP a nuestro objetivo con toda la información rápida

excepto con una dirección hardware de destino errónea.

Una máquina que no esté en modo promiscuo nunca verá este paquete, puesto que

no era destinado a ellos, por lo tanto, no contestará.

Si una máquina está en modo promiscuo, la petición ARP sería considerada y el

núcleo la procesaría y contestaría. Por la máquina que contesta, sabemos que

estamos en modo promiscuo.

### • El test Etherping

Enviamos un "ping echo" al host a testear con una IP de destino correcta y

dirección MAC falseada.

Si el host responde, es que su interfaz está en modo promiscuo, es decir, existe un

sniffer a la escucha y activo.

Una vez explicado como funciona una interfaz en modo promiscuo y los métodos que existen para detectarlas en una red LAN procederemos a la detección.

En mi caso realizaré la detección con dos de los métodos nombrados anteriormente:

* Test ARP
* Test EtherPing

## Infraestructura Previa

Contamos con 3 máquinas 🡪

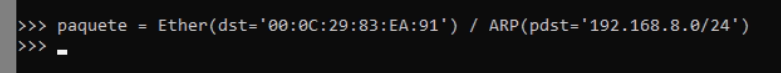
* Windows 10 (Scapy) IP:192.168.8.115 | Es la máquina de donde enviaremos los paquetes
* Windows 10 IP:192.168.8.111 Anfitrión de las máquinas virtualizadas 🡪
* Windows 7 (Tarjeta Promiscua) IP:192.168.8.117 | Es la máquina que está capturando el tráfico de la red, es la que deberíamos de detectar como maliciosa
* Debian (Equipo) IP:192.168.8.118 MAC:00-0C-29-83-EA-91 | Es la máquina que recibirá los paquetes, es benigna.

## Creación de paquetes con SCAPY

Utilizaré la herramienta SCAPY para la creación de paquetes y modificación de parámetros a mi gusto.

### Método (Test ARP)

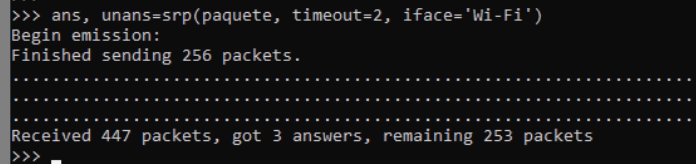
Crearemos un paquete en el que montaremos el protocolo “Ehter” y el de ARP



Como vemos en Ether indicaremos la MAC del dispositivo al cual va dirigido el paquete

Y en ARP podemos indicar un host en específico, pero yo pondré una red para que consulte a toda la red.

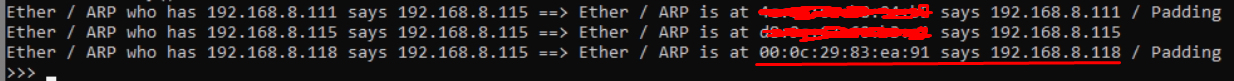
Ahora mandaremos el paquete con lo siguiente:



“ans” y “unans” se recogen los paquetes contestado y no contestado.

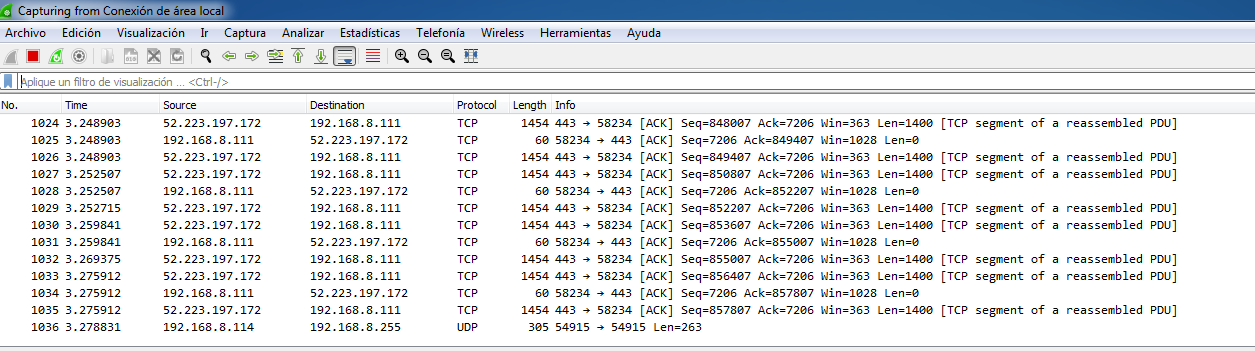
Con srp enviaremos y recibiremos respuesta para la capa 2 (IP,ARP…) del paquete que le indicamos dentro “paquete” y le indicamos a través de que interfaz queremos que lo haga (iface=’Wi-Fi’)

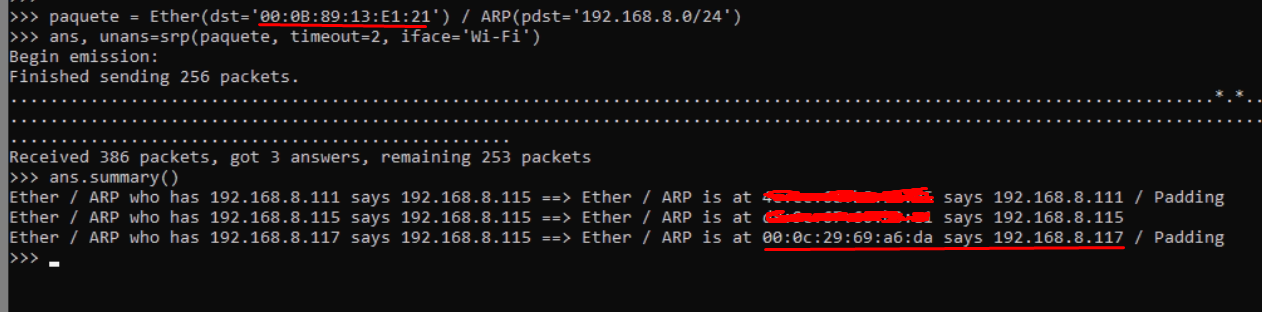
Vemos que hemos recibido 3 respuesta, pues veremos quien ha respondido con “ans.summary()”



Como vemos ha respondido solo Debian y nadie más ya que hasta ahora no se ha activado el modo promiscuo en Windows 7

Procedemos a activar el modo promiscuo en Windows 7, iniciando captura de paquetes con Wireshark.



Y volvemos a enviar el mismo paquete que construimos anteriormente, pero le ponemos una MAC falsa para ver si responde alguien que esté capturando el tráfico y no debería, y consultamos las respuestas:

Como podemos apreciar ha contestado alguien a una MAC inexistente por lo que podemos determinar que ese dispositivo con la IP:192.168.8.117 que es Windows 7 tiene activa una tarjeta promiscua.

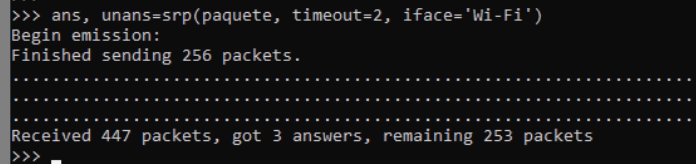
La tarjeta promiscua dice que esa dirección física falsa es suya para pueda obtener el paquete y leerlo, pero lo que no sabe es que ese dispositivo no existe y se está delatando.

### Método (Test EtherPing)

Crearemos un paquete en el que montaremos el protocolo “Ehter”, “IP” y “ICMP”

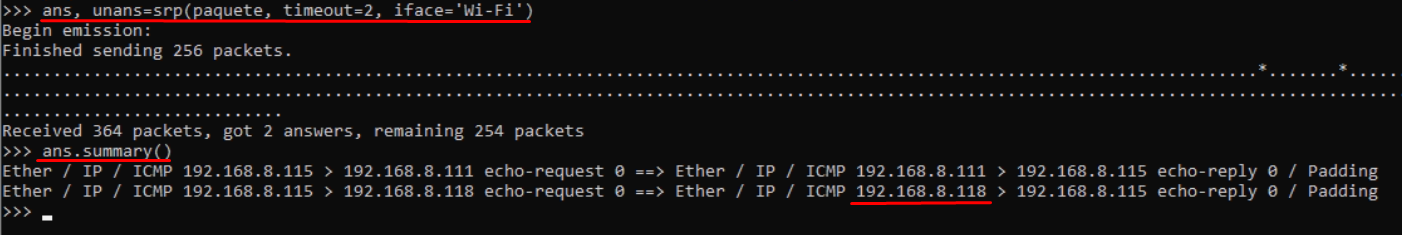


Ahora mandaremos el paquete con lo siguiente:



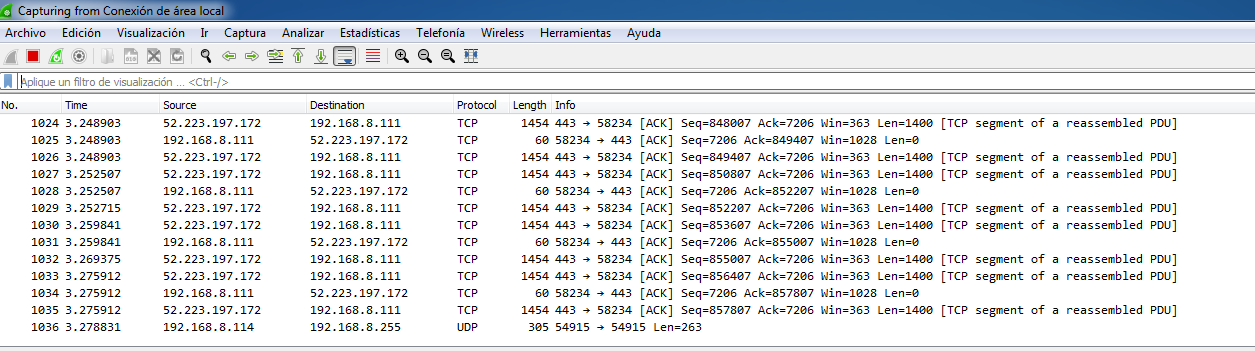
“ans” y “unans” se recogen los paquetes contestado y no contestado.

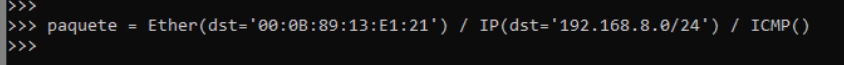
Con srp enviaremos y recibiremos respuesta para la capa 2 (IP,ARP…) del paquete que le indicamos dentro “paquete” y le indicamos a través de que interfaz queremos que lo haga (iface=’Wi-Fi’)

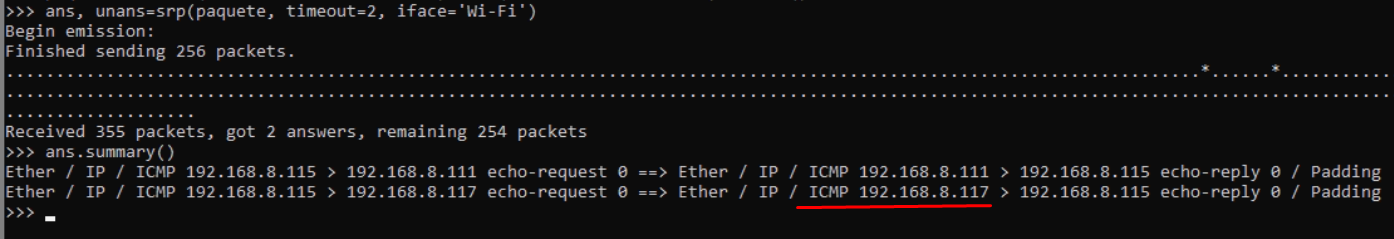
Vemos que hemos recibido 3 respuesta, pues veremos quien ha respondido con “ans.summary()”

Como vemos ha respondido solo Debian y nadie más ya que hasta ahora no se ha activado el modo promiscuo en Windows 7

Procedemos a activar el modo promiscuo en Windows 7, iniciando captura de paquetes con Wireshark.



Y volvemos a enviar el mismo paquete que construimos anteriormente, pero le ponemos una MAC falsa para ver si responde alguien que esté capturando el tráfico y no debería, y consultamos las respuestas:



Como podemos apreciar ha contestado alguien a una MAC inexistente por lo que podemos determinar que ese dispositivo con la IP:192.168.8.117 que es Windows 7 tiene activa una tarjeta promiscua.

La tarjeta promiscua dice que esa dirección física falsa es suya para pueda obtener el paquete y leerlo, pero lo que no sabe es que ese dispositivo no existe y se está delatando.