



Ataques a redes de datos IPv4

Índice



1 MAC/CAM	3
2 Vlan Hopping	8
3 Man in the Middle	10
3.1 ARP Spoofing	11
3.2 Network Packet Manipulation	13
3.3 Contramedidas	15
4 Ataques a DHCP	20
5 Otros tipos de Spoofing	23
5.1 IP Spoofing	23
5.2 DNS Spoofing	25

A estas alturas todos sabemos que un ataque informático es un mecanismo por el que un atacante, mediante un sistema informático, tiene el objetivo de controlar, monitorizar, dañar o desestabilizar otro sistema distinto (ordenador, red privada,...). A continuación, pasamos a estudiar los diferentes ataques en IPv4.

1. MAC/CAM

La tabla CAM, que también conocemos como MAC Address Table guarda las direcciones MAC que un switch determinado aprende a través de las tramas que recibe (almacena el campo MAC Address Source). Por defecto, una MAC permanece en la tabla CAM un máximo de 300 segundos tras la última acción registrada. Este tiempo se denomina aging timer.

Podemos modificar el aging timer de la siguiente forma:

```
Switch (config) # mac address-table aging-time seconds
```

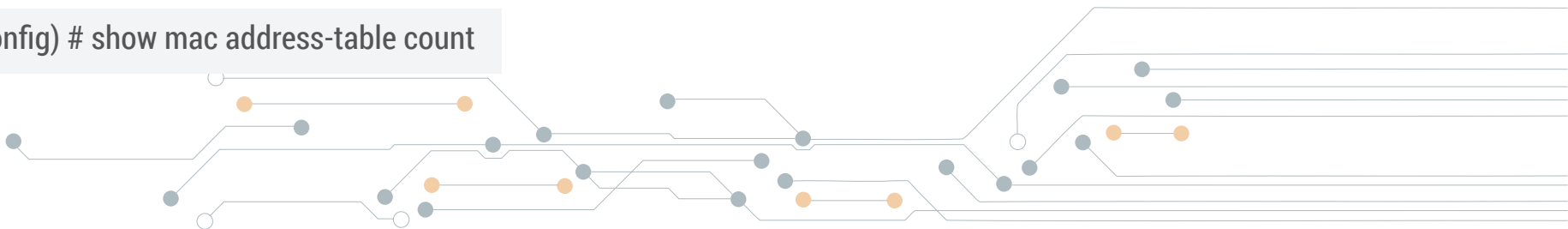
Además, podemos insertar entradas estáticas en la CAM:

```
Switch (config) # mac address-table static mac-address vlan vlan-id interface type  
mode/num
```

La tabla CAM tiene una longitud máxima determinada y es almacenada en la RAM del equipo para que la consulta tarde muy poco en realizarse.

El comando para saber el espacio disponible de la tabla es el siguiente:

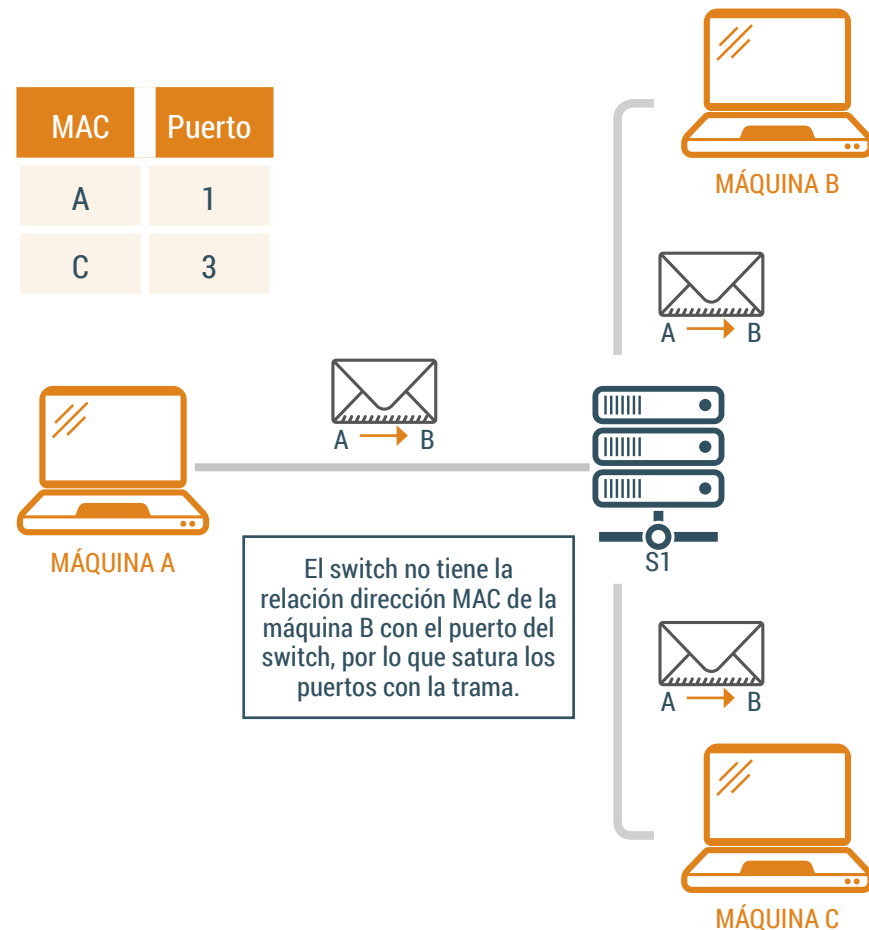
```
Switch (config) # show mac address-table count
```



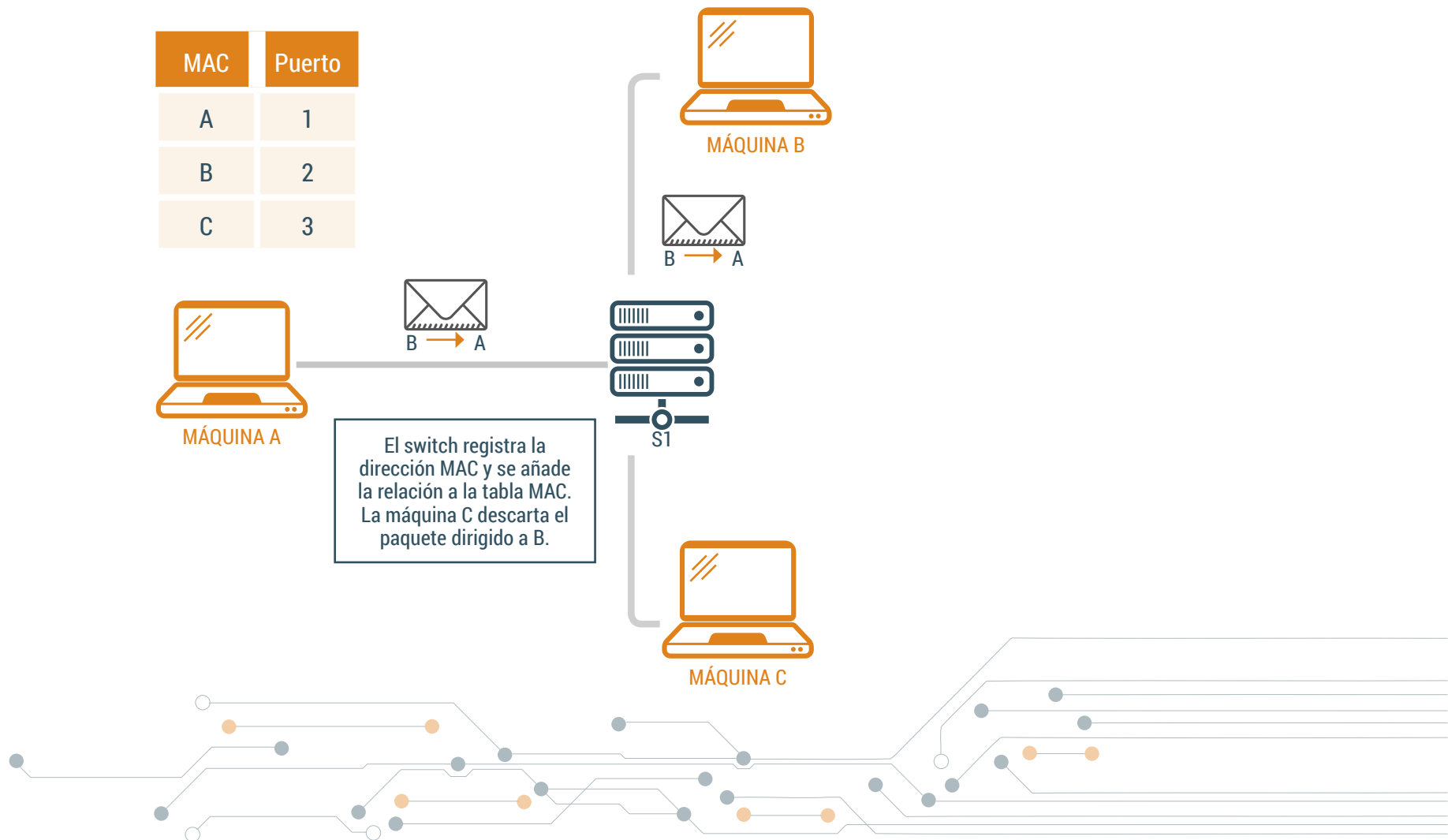
La tabla de direcciones MAC de un determinado switch almacena las direcciones MAC, así como su relación con cada puerto y la VLAN específica. En esta tabla se basa el funcionamiento de un switch, ya que cuando recibe una trama, busca en la tabla de direcciones el destino. Cuando un switch recibe una trama, la dirección MAC de origen queda registrada en la tabla, por tanto, si esa dirección ya está almacenada en la tabla, el switch reenvía la trama al puerto almacenado. Si por el contrario no existe, el switch satura todos los puertos (excepto el de origen) con la trama.

Este comportamiento puede utilizarse para realizar ataques de desbordamiento a tablas de direcciones MAC, también conocidos como "Ataques de desbordamiento de la tabla CAM". A continuación se explica el funcionamiento de este tipo de ataque paso a paso mediante ilustraciones.

En la primera imagen se puede observar el comportamiento normal de un switch. Como se acaba de comentar, dicho switch recibe la trama, busca la dirección de B de destino en su tabla y como no tiene la dirección del destino almacenada, difunde la trama enviándola por todos los puertos del switch exceptuando el de origen.



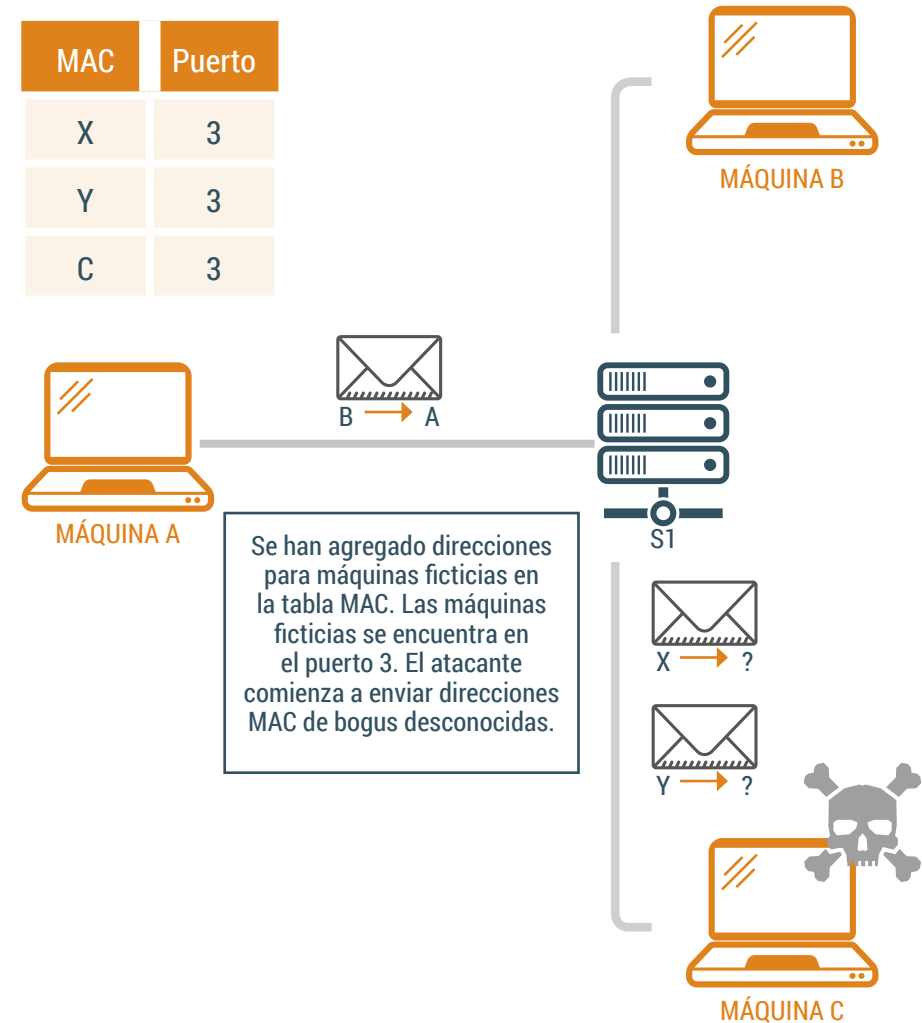
Una vez que el switch ha difundido la trama, el destino, en este caso B, al haber recibido la trama, envía una respuesta a la máquina B. De esta forma el switch descubre que la MAC de B se encuentra en el puerto 2 y almacena estos datos en la tabla de MAC. La máquina C también recibe la trama pero como identifica que no es el destino, la desecha. A partir de ese momento, el switch tiene e conocimiento del puerto por el que debe reenviar las tramas cuando el destino es la máquina B.



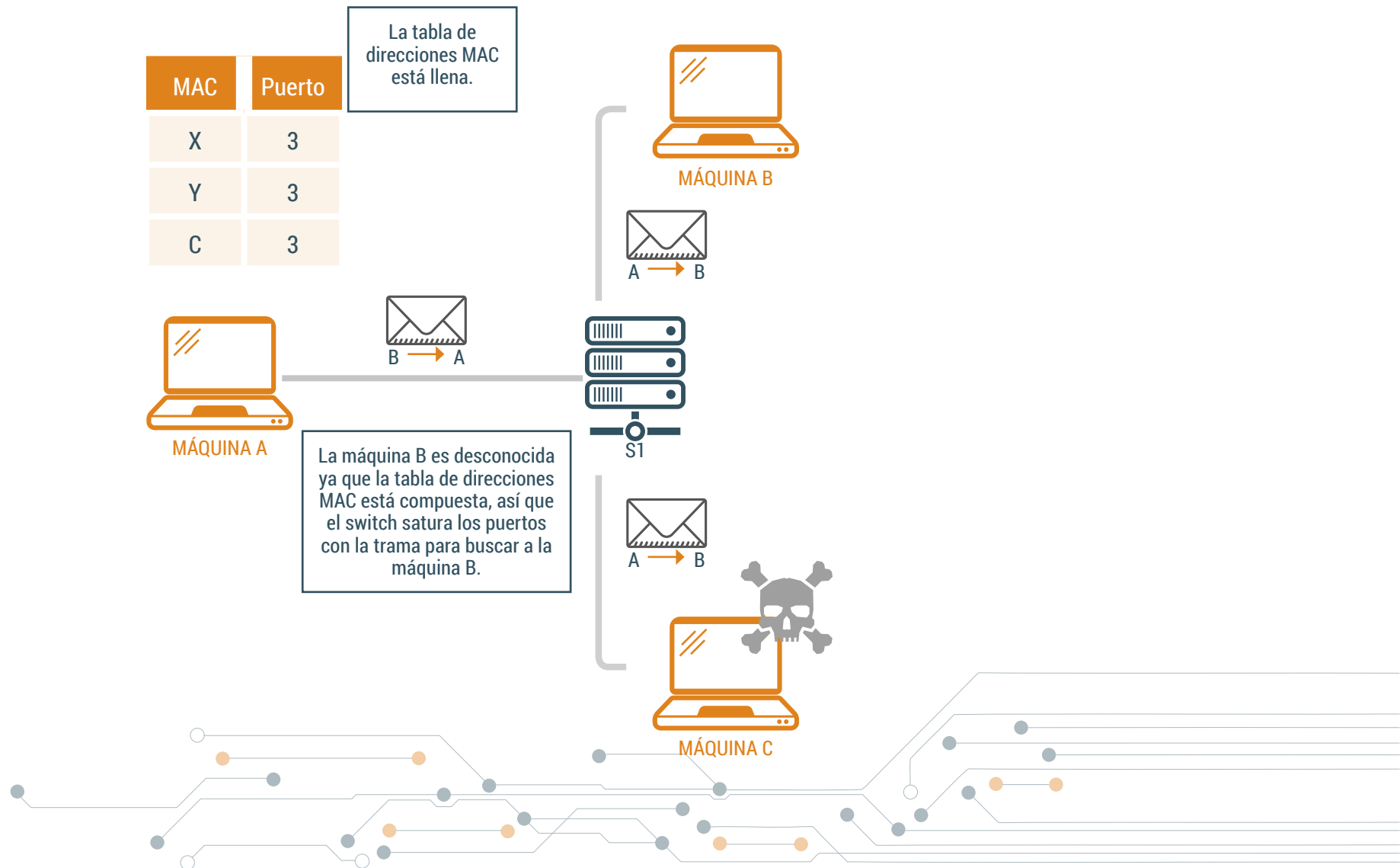
Con el proceso de almacenamiento de relaciones en la máquina de direcciones MAC surge el riesgo de llenar la tabla, ya que estas tienen limitación de registros.

Un ataque de desbordamiento por CAM se aprovecha de esta vulnerabilidad e intentará sobrecargar al switch con direcciones MAC que no existen para así poder completar la tabla y desbordarla. Existen herramientas que pueden generar miles de entradas por minuto.

En la siguiente figura un atacante en la máquina C está generando direcciones MAC falsas, con el consecuente almacenamiento por parte del switch de dichas direcciones. La máquina C continúa con este comportamiento hasta conseguir que la tabla se complete y al llegar a este punto, el switch entra en "fail-open". Al entrar en este modo, el switch reenvía las tramas a todas las máquinas existentes en la red, dando visibilidad al atacante de todas las tramas enviadas.



Por tanto, mientras la tabla de direcciones MAC esté completa, el switch permanecerá en ese estado y difundirá por cada puerto todas las tramas recibidas. En la siguiente figura se puede apreciar cómo el switch reenvía la trama dirigida a la máquina B por todos los puertos, siendo accesible por tanto a la máquina B.

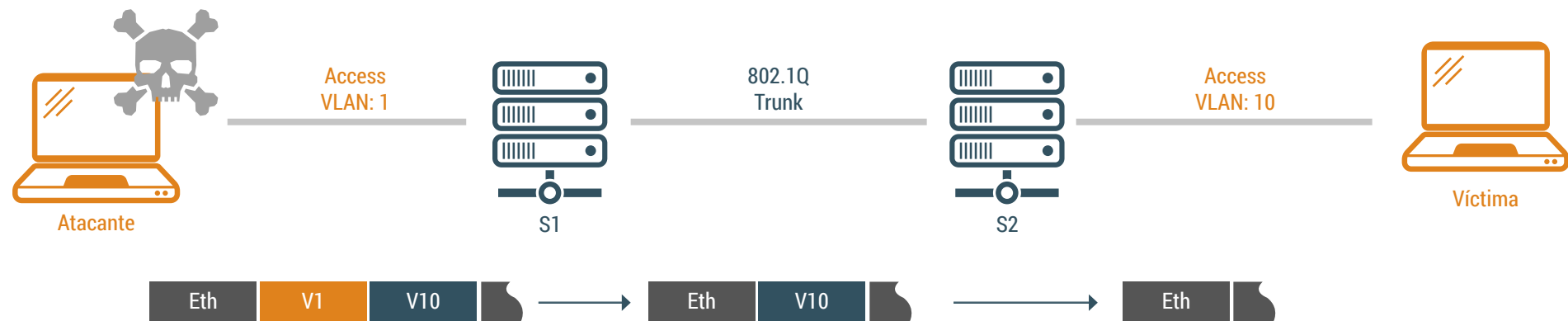


2. Vlan Hopping

Para que una red de VLANs sea administrada por un switch es necesaria la creación de un puerto trunk que tiene acceso a todas las VLANs y se utiliza para transmitir paquetes de varias VLAN en el mismo enlace físico. Para su administración se utiliza DTP (Dynamic Trunk Protocol) y es donde se realizan principalmente estos ataques. DTP se utiliza para la negociación de un enlace entre dos dispositivos y para negociar el tipo de encapsulación que se utilizará (802.1Q).

VLAN Hopping aprovecha la vulnerabilidad que se da en entornos VLAN, en los que hay una conexión por puertos troncales en los Switch. Realizando este ataque podremos mandar y recibir paquetes desde una VLAN a la que el sistema final no debería poder acceder.

El modo de operación consiste en que la máquina atacante aspira a conseguir acceso a una VLAN en la que no es autorizado mediante el anexo de dos etiquetas en los paquetes que salen del cliente. Dichas etiquetas se añaden a los paquetes que establecen a qué VLAN corresponden (VLAN ID). El método se denomina doble etiquetado. El ataque comienza cuando el atacante envía un paquete conectado a un switch añadiendo dos etiquetas VLAN en la cabecera del paquete. Si el primer atacante está conectado al switch, la primera etiqueta coincide. Si el atacante está conectado a un 802.1Q Trunk, la primera etiqueta coincide con la VLAN nativa (generalmente 1). La segunda etiqueta identifica la VLAN a la que el atacante le gustaría reenviar el paquete.



Cuando el switch recibe el paquete del atacante, elimina la primera etiqueta. A continuación, reenvía el paquete a todos los switches vecinos (ya que también utilizan la misma VLAN nativa). Debido a que la segunda etiqueta nunca se eliminó después de entrar en el primer switch, los siguientes switches que reciben el paquete ven la etiqueta restante como el destino de la VLAN y reenvían el paquete al puerto de destino en esa VLAN.

A continuación, se detalla un ejemplo para la mejor comprensión de lo recientemente estudiado.

Se considera un servidor web seguro en una VLAN denominada VLAN2. Los hosts en VLAN2 tienen acceso permitido al servidor web; las máquinas de fuera de VLAN2 son bloqueados por filtros de capa 3.

Una máquina atacante en una VLAN separada, llamada VLAN1 (Nativa), crea un paquete especialmente diseñado para atacar al servidor web, colocando una cabecera etiquetando al paquete como si perteneciera a VLAN2 detrás de la etiqueta de VLAN1. Cuando se envía el paquete, el switch analiza el encabezado VLAN1 y al ver que es el predeterminado, elimina la etiqueta y reenvía el paquete. El siguiente switch ve el encabezado VLAN2 coloca el paquete en VLAN2. El paquete llega así al servidor destino como si fuera enviado desde otra máquina en VLAN2, ignorando cualquier filtro de capa 3 que pudiera estar en su lugar.

Contramedidas

La característica clave de un ataque de este tipo es la explotación de la VLAN nativa. Dado que VLAN1 es la VLAN predeterminada para los puertos de acceso y la VLAN nativa predeterminada en puertos troncales, es un objetivo fácil.

La primera contramedida es eliminar los puertos de acceso de la VLAN1 predeterminada, ya que el puerto del atacante debe coincidir con el de la VLAN nativa del switch.

```
Switch(config-if)# switchport access vlan 10
```

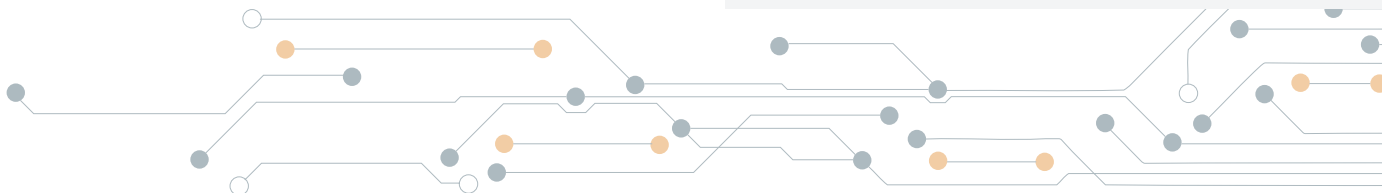
```
Switch(config-if)# description access_port
```

La segunda contramedida es asignar la VLAN native a todos los puertos troncales del switch a una VLAN no utilizada.

```
Switch(config-if)# switchport trunk native vlan 99
```

Las medidas anteriores evitaban el ataque de salto de VLAN, pero debemos tener en cuenta que existe una tercera opción. Podremos etiquetar alternativamente la VLAN en todos los puertos troncales, desactivando todo el tráfico sin etiquetar sobre la interfaz.

```
Switch(config-if)# switchport trunk native vlan tag
```



3. Man in the Middle

Man in the Middle (MITM) es un conocido sistema de ataque en el que se vulnera un canal entre dos máquinas, y sin que ninguno de los dos extremos sea consciente, la información que se envía entre las dos máquinas puede ser leída o modificada. Es decir, el atacante se sitúa entre la máquina A y la máquina B, y cuando la máquina A envíe información a la máquina B, dicha información será susceptible ya que llegará primero al equipo de la víctima que reenviará o modificará, en función del tipo de ataque, la información hacia la máquina B.

A continuación, se muestra el funcionamiento básico de este tipo de ataque:

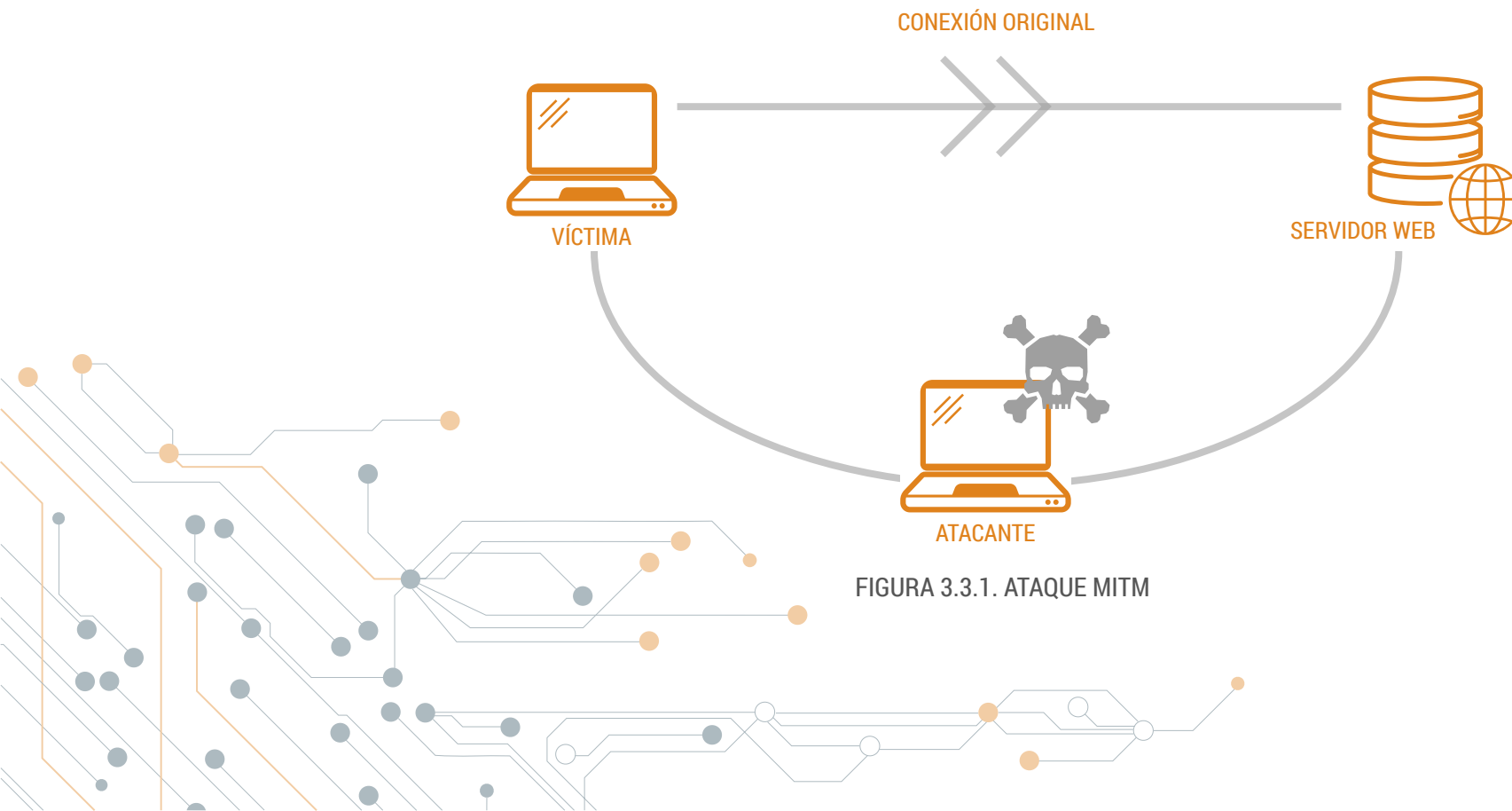


FIGURA 3.3.1. ATAQUE MITM

3.1 | ARP Spoofing

En este tipo de ataque el atacante envenena la tabla ARP de la víctima de forma que envía falsos mensajes ARP a dicha víctima. ARP Spoofing no se realiza en redes con hubs, sino en redes switcheadas. A continuación, se detalla un ejemplo teórico que nos ayudará a comprender el funcionamiento de ARP Spoofing, también conocido como ARP Poisoning.

En primer lugar, se considera que el atacante dispone de la herramienta necesaria (esta herramienta puede ser caín, ettercap, nemesiis, ...).

Imaginemos el siguiente escenario:

TABLA ARP VÍCTIMA 1, máquina con IP 10.0.0.2, dirección IP 10.0.0.1 (router), MAC AA:AA:AA:AA:AA:AA

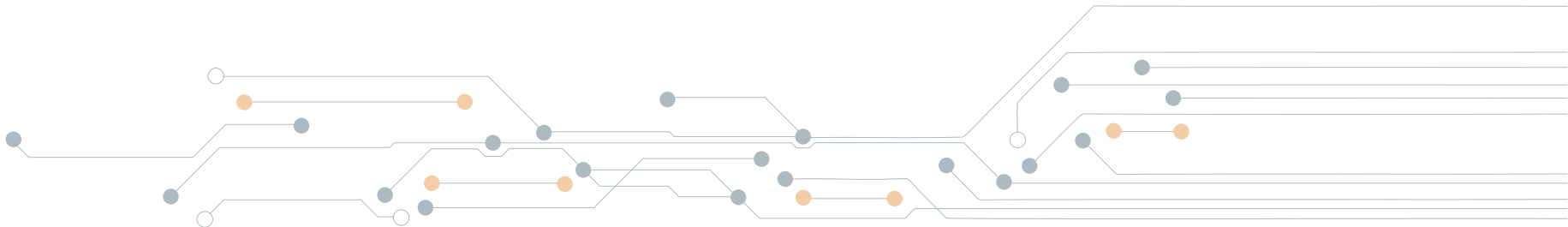
TABLA ARP VÍCTIMA 2, máquina con IP 10.0.0.3, dirección IP 10.0.0.2, MAC AA:AA:AA:AA:AA:AA

MAC ATACANTE: AA:BB:AA:BB:AA:BB

El atacante realizará un ARP Spoofing modificando el valor de las tablas ARP de las dos máquinas, siendo el resultado el siguiente:

TABLA ARP VÍCTIMA 1, máquina con IP 10.0.0.2, dirección IP 10.0.0.1 (se cree el router), MAC AA:BB:AA:BB:AA:BB

TABLA ARP VÍCTIMA 2, máquina con IP 10.0.0.3, dirección IP 10.0.0.2, MAC AA:BB:AA:BB:AA:BB



¿Qué ha conseguido el atacante?

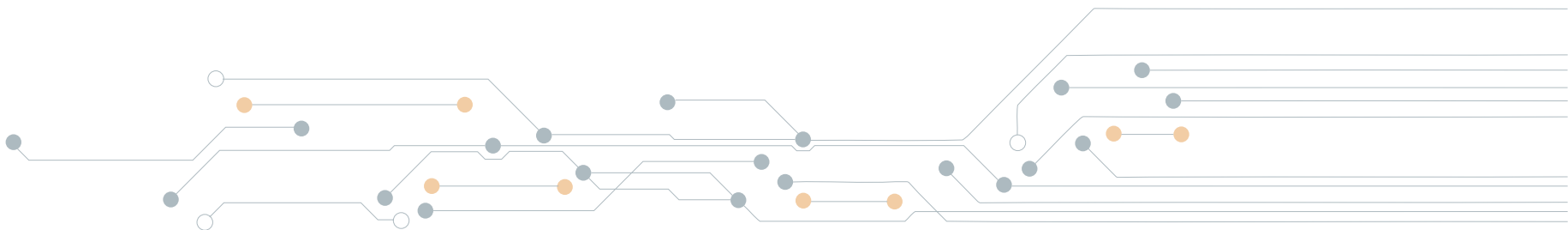
El atacante ha conseguido que todos los envíos de información de la máquina VÍCTIMA 1 hacia cualquier destino pasarán por la máquina del atacante. Del mismo modo, la información que envíe la VÍCTIMA 2 (por ejemplo, el router) a la VÍCTIMA 1 también pasarán por la máquina del atacante.

Información en texto plano

Después de entender en qué consiste el ataque, podemos comprender que, si la autenticación de un determinado servidor es en texto plano, con un ARP SPOOFING podremos obtener la información de forma muy sencilla, ya que toda la información pasará por la máquina del atacante.

¿Estamos a salvo de este tipo de ataques si utilizamos HTTPS?

El cifrado de las comunicaciones dificulta que se pueda obtener la información tan fácilmente ante un ataque básico de ARP Spoofing, pero existen métodos más elaborados en lo que se puede obtener dicha información.



3.2 | Network Packet Manipulation

La técnica Network Packet Manipulation permite modificar el contenido de un paquete de red. Un atacante que se encuentre en una situación privilegiada, como puede ser en medio de una comunicación, gracias entre otras posibilidades a un ARP Spoofing en una red de área local, puede observar el tráfico que circula a través de él. Además de observar, imagine que puede modificar algún dato de un paquete que circula por la tarjeta de red. En esto se basa la técnica Network Packet Manipulation, por lo que, a priori se necesita realizar un MiTM y poder colocarse en medio de la comunicación.

Para llevar a cabo un ejemplo de esta técnica se utilizará Ettercap. Esta herramienta permite realizar diferentes tipos de ataques de red, entre ellos el propio ARP Spoofing, estudiado anteriormente. Ettercap permite crear una serie de filtros, a través de su herramienta Etterfilter, para encontrar los bytes que deseamos

modificar y, a continuación, proporciona una forma de reemplazar fácilmente la información con lo que deseamos. Otros programas de manipulación de paquetes, como Airpwn, permiten hacer este tipo de manipulación también.

Crear un filtro ettercap es bastante sencillo. Sólo tenemos que decidir qué datos deseamos reemplazar y con qué. Un escenario divertido y común es reemplazar imágenes en webs con alguna imagen a nuestra elección.

A continuación, vamos a estudiar un ejemplo en el que vamos a reemplazar una imagen con nuestra propia imagen llamada OWNED.gif.

Lo primero de todo que tenemos que hacer, como acabamos de comentar, es crear el filtro. En un editor de texto cualquiera crearemos un nuevo fichero al que llamaremos owned.filter.

```
# owned.filter

if (ip.proto == TCP && tcp.src == 80) {
    replace("img src=", "img src=\"http://[redacted]/OWNED.gif \"");
    msg("image replaced\n");
}
```

Cuando tenemos creado el filtro, necesitamos compilarlo con el compilador de filtros de Ettercap, llamado Etterfilter:

```
etterfilter owned.filter -o owned.ef
```

Esto generará un fichero con extensión ef, el cual es el filtro compilado.

Después de compilar el filtro deberíamos ver la siguiente respuesta:

```
etterfilter NG-0.7.3 copyright 2001-2004 ALOR & NaGA
12 protocol tables loaded:
    DECODED DATA udp tcp gre icmp ip arp wifi fddi tr eth

11 constants loaded:
    VRRP OSPF GRE UDP TCP ICMP6 ICMP PPTP PPPoE IP ARP

Parsing source file 'owned.filter' done.
Unfolding the meta-tree done.
Converting labels to real offsets done.
Writing output to 'owned.ef' done.
-> Script encoded into 7 instructions.
```

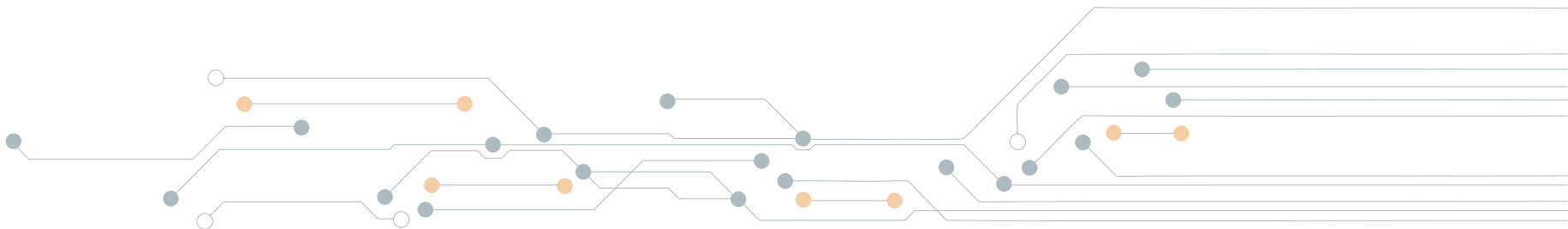
Una vez hecho esto, ejecutamos Ettercap y lanzamos el filtro:

```
ettercap -Tq -I eth0 -F owned.ef -M ARP /10.157.6.3/ //
```

El parámetro -T indica que usaremos Ettercap a través de una línea de comandos. Por otra parte, el módulo ARP permite indicar un target o toda la red, por lo que hemos indicado el host concreto que queremos envenenar.

A partir de ahora, cuando un paquete atraviese la máquina de sniffing, la trama que contiene la imagen será reescrita y la frase "IMAGE REPLACED" aparecerá como salida en la consola.

Debemos indicar que esto es bastante imperfecto, ya que requiere qué el desarrollador que hizo la máquina web que estamos intentando hackear escribiera o similares, y muchas personas escriben algo como indicando el tamaño deseado, o frases similares. Dado que protocolos como HTML permiten poner muchos elementos diferentes en orden diferente, el filtro no funcionará en el 100 por cien de los casos pero sirve de base didáctica para comprender en qué consiste el Network Packet Manipulation.



3.3 | Contramedidas

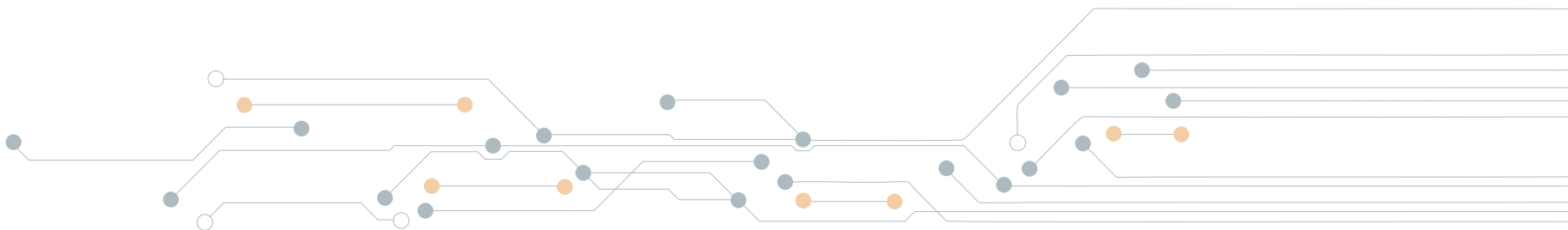
CONTRAMEDIDAS ARP SPOOFING

Existen diferentes formas y vías para protegerse contra el ataque de ARP Spoofing. Una vez se entiende en qué consiste dicho ataque, hay que tener en cuenta que el objetivo es lograr que las entradas que se encuentren en la tabla ARP o CAM, sean realmente las legítimas, y no haya ninguna entrada de un atacante que esté "spoofeando" la tabla.

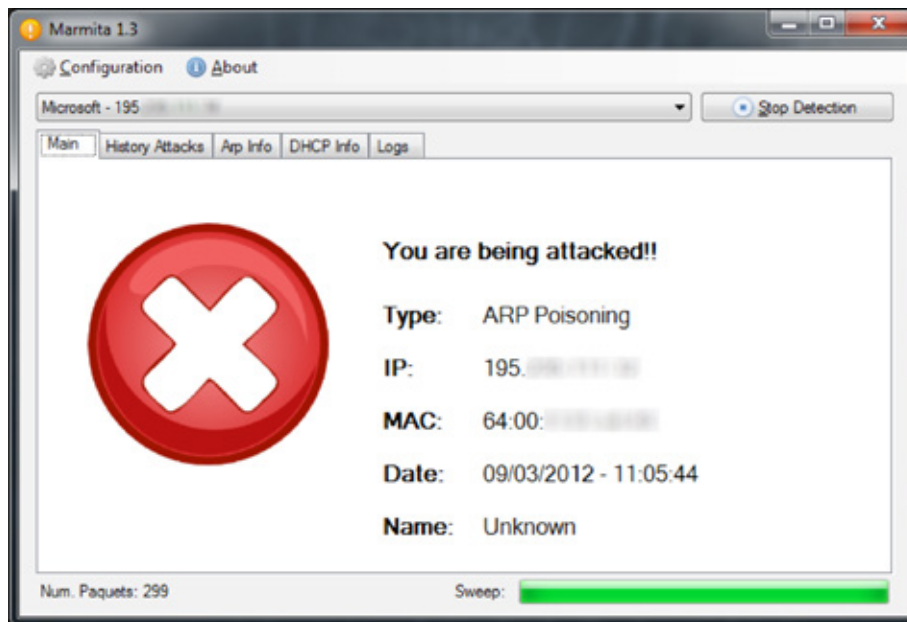
Simplemente, para recordar, el protocolo ARP consiste en preguntar a todos los equipos de la red de ámbito local, con el objetivo de encontrar al equipo buscado, partiendo del conocimiento de una dirección IP, para acabar conociendo su relación dirección IP con dirección física o MAC. El problema radica en que un atacante puede llevar a cabo ese tipo de respuestas falsificadas, con el objetivo de lograr cambiar la tabla ARP o CAM de un usuario. De esta forma, se puede conseguir que la víctima envíe su tráfico a otro equipo, pensando que éste es el router verdadero de la red, por lo que el atacante consigue situarse en medio de una comunicación, es decir, Man in the Middle.

Existen diferentes formas de protegerse de este tipo de ataques, algunas medidas son básicas, pero muy eficaces, mientras que otras, debido a la complejidad de infraestructura pueden ser más costosas de implantar, pero también son muy eficaces. A continuación, se enumera una serie de medidas por nivel de complejidad en la integración:

- Programa que monitorice el estado de la tabla ARP para evitar cambios no consentidos.
- Programa que monitoricen las peticiones de red que llegan a la máquina, con el objetivo de detectar cuando existen ARP Reply provenientes de una máquina, a la cual no se ha pedido un ARP Request.
- Estudio del volumen de datos o paquetes que han sido enviados y recibidos. De esta forma se puede detectar un comportamiento anómalo en el protocolo ARP por parte de un atacante.



En otras palabras, existen soluciones “caseras”, pero eficaces con scripting, por ejemplo, en Bash o en Powershell. Por otro lado, existen soluciones como ARP On o Marmita, las cuales son aplicaciones que permiten detectar ataques o intentos de ataques contra la tabla ARP.



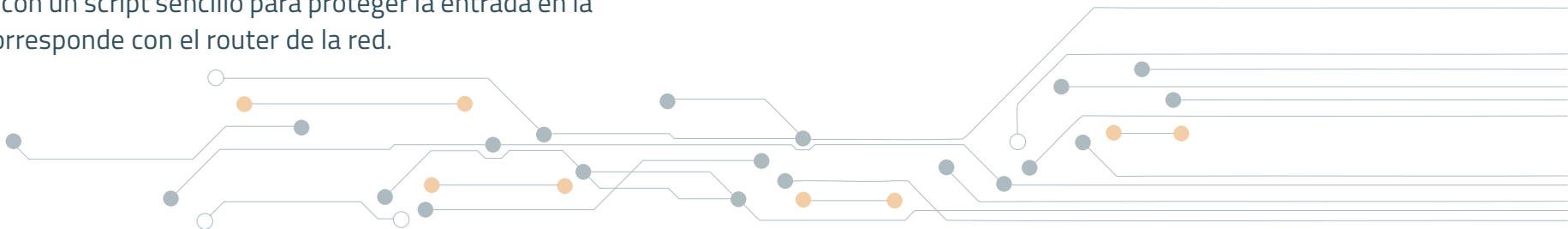
Por último, hay sistemas de detección de intrusiones que pueden detectar los intentos de ataques ARP Spoofing. En esta ocasión, se va a ejemplificar con un script sencillo para proteger la entrada en la tabla ARP que corresponde con el router de la red.

En este ejemplo, se supone el siguiente escenario:

- Un router.
- Un equipo de la víctima.
- Un equipo del atacante.

Las estradas de una tabla ARP son dinámicas, es decir, tienen un tiempo por defecto de vida, siempre y cuando no sean refrescadas. Es decir, cuando pase 'X' tiempo sin que llegue un ARP Reply de la máquina correspondiente con dicha entrada, ésta será eliminada. Incluso, si llega un ARP Reply que pertenezca a una dirección IP que existe en la tabla, pero su dirección física es distinta y la entrada está configurada en modo dinámico, ésta cambiará de valor por la nueva dirección física.

¿Cómo se crea una entrada estática en la tabla ARP? La instrucción para crear la entrada o modificar el comportamiento dinámico por estático es `arp -s <dirección IP> <dirección física>`. ¿Qué ventaja tiene la entrada estática? Aunque haya un equipo en la red que indique que la dirección física asociada a una dirección IP ha cambiado, no se atenderá dicha solicitud, ya que la entrada solo puede ser cambiada manualmente en local. A continuación, se propone, a modo de ejemplo, el siguiente script.



Este script, a modo de ejemplo, detecta que la entrada de la dirección IP 192.168.1.1 ha cambiado en la tabla ARP y lo notifica enviando un mensaje de alerta.

Hay que tener en cuenta que, cualquier persona que utilice un equipo portátil estará cada día en un lugar diferente, es decir, conectado a diferentes redes. Los equipos de sobremesa suelen estar mucho más “atados” al sitio, por lo que la utilización de este tipo de scripts puede ser válido, siempre y cuando puedan ser fácilmente actualizables, con el objetivo de poder variar las direcciones físicas válidas o legítimas.

```
#!/bin/bash
if [ $# -ne 1 ]
then
echo "Usage ./mitm.sh <dirección mac>"
exit
fi
mac=$1
while true
do
entradaRouter=$( arp -a | grep 192.168.1.1 | cut -d' ' -f4 )
if [ $mac != $entradaRouter ]
then
echo "Atento la Mac ha cambiado"
exit
fi
sleep 2
done
```

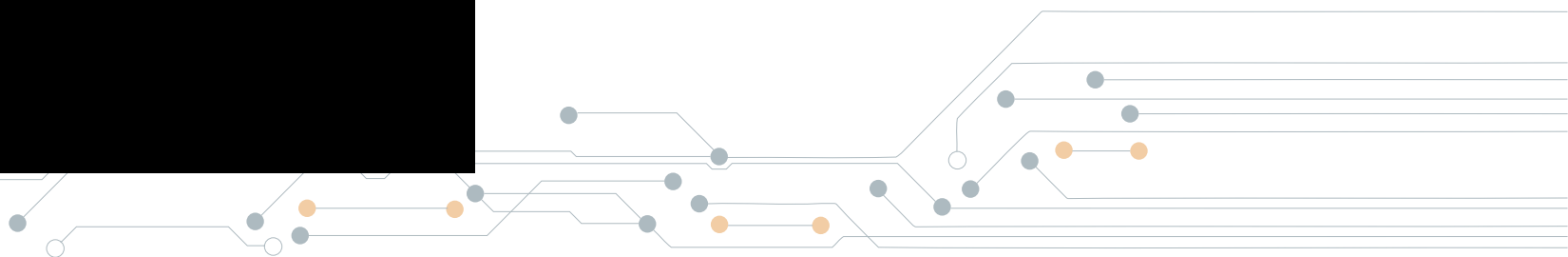
CONTRAMEDIDAS DE NETWORK PACKET MANIPULATION

La manipulación de paquetes permite a un atacante poder modificar el tráfico de una víctima para obtener una ventaja en la red. Para lograr esto, el atacante debe estar colocado en el medio de la comunicación. Generalmente, para protegerse de la manipulación de paquetes se debe optar por protocolos de cifrado que protejan la comunicación, y hagan realmente complejo que un atacante que esté situado en medio de la comunicación pueda analizar, comprender y modificar con sentido el tráfico.

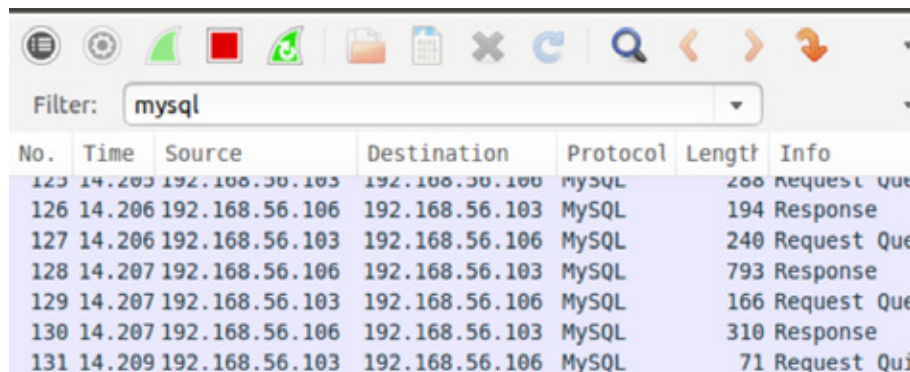
La manipulación de paquetes es crítica en entornos como las bases de datos, ya que, por defecto, los motores de bases de datos no cifran las comunicaciones entre las aplicaciones web o clientes y servidores dónde se encuentran la base de datos. Alguien que pueda situarse en medio de la comunicación podría manipular las peticiones o queries que se realizan contra la base de datos, pudiendo obtener un privilegio en el sistema.

Ejemplo: Proteger comunicación entre Wordpress y MySQL

La idea de este ejemplo o prueba de concepto es mostrar al alumno como la comunicación entre un cliente de base de datos o una aplicación web contra un servidor de base de datos debe estar protegida, y en la mayoría de las ocasiones no lo está.



En primer lugar, si se visualiza el tráfico entre Wordpress y MySQL, se puede ver como éste se encuentra sin cifrar.



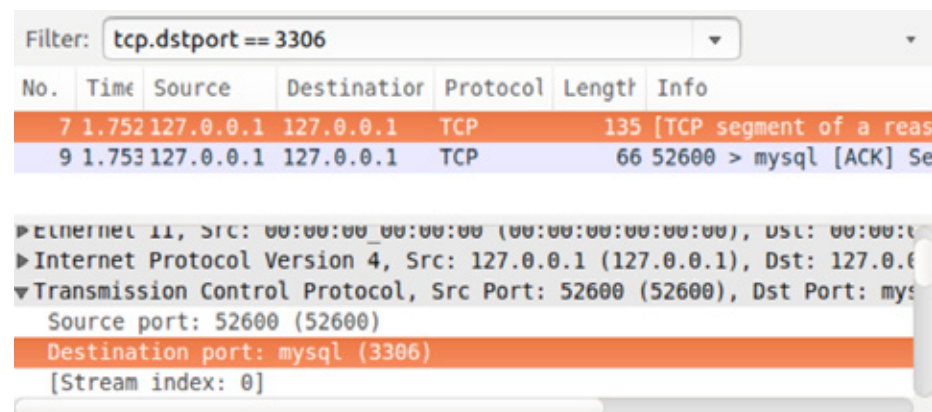
No.	Time	Source	Destination	Protocol	Length	Info
125	14.205	192.168.56.103	192.168.56.106	MySQL	200	Request Que
126	14.206	192.168.56.106	192.168.56.103	MySQL	194	Response
127	14.206	192.168.56.103	192.168.56.106	MySQL	240	Request Que
128	14.207	192.168.56.106	192.168.56.103	MySQL	793	Response
129	14.207	192.168.56.103	192.168.56.106	MySQL	166	Request Que
130	14.207	192.168.56.106	192.168.56.103	MySQL	310	Response
131	14.209	192.168.56.103	192.168.56.106	MySQL	71	Request Qui

Ahora, hay que cambiar la configuración del motor de base de datos para que permita cifrar las comunicaciones. Esto sería algo similar a que alguien se conectara contra un servidor web, éste debería estar preparado para poder cifrar la comunicación, si se maneja información sensible.

En el servidor de base de datos se comprueban las variables para ver si openssl se encuentra disponible y puede ser utilizado. Esto se lleva a cabo a través de la ejecución de `show variables like '%ssl';`. Las variables `have_openssl` y `have_ssl` deben encontrarse en MySQL. Ahora, es momento, de configurar los certificados necesarios para crear las comunicaciones cifradas: `ssl_ca`, `ssl_cert` y `ssl_key`, que indican la ruta de la CA, la ruta del certificado del servidor y la clave del certificado del servidor.

La generación de la CA, del certificado y de la clave se realiza a través de los siguientes comandos:

- `Openssl genrsa 2048 > ca-key.pem - openssl req -sha1 -new -x509 -nodes -days 3600 -key ca-key.pem > ca-cert.pem`. Con esto se tiene certificado y clave privada de la CA.
- `Openssl req -sha1 -newkey rsa:2048 -days 730 -nodes -keyout server-key.pem > server-req.pem`. Con esto se genera clave privada para el servidor.
- `Openssl rsa -in server-key.pem -out server-key.pem`. Se exporta clave privada, tipo RSA.
- `Openssl x509 -sha1 -req -in server-req.pem -days 730 -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 > server-cert.pem`. Se genera un certificado de servidor utilizando el certificado de la CA.



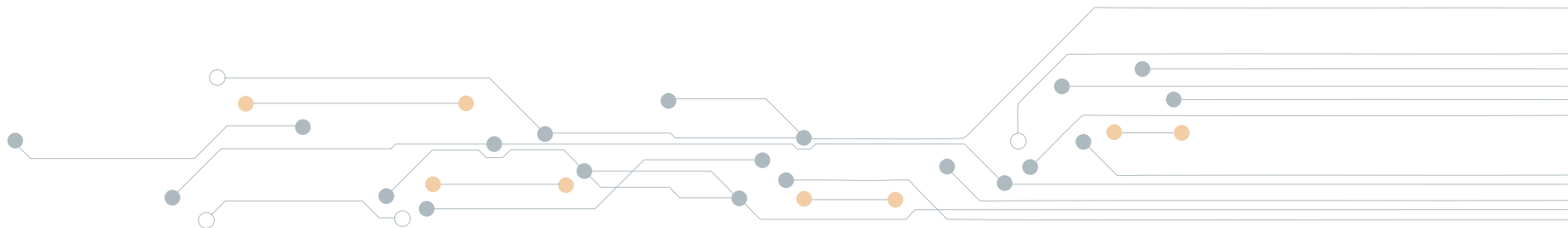
No.	Time	Source	Destination	Protocol	Length	Info
7	1.752	127.0.0.1	127.0.0.1	TCP	135	[TCP segment of a reas
9	1.753	127.0.0.1	127.0.0.1	TCP	66	52600 > mysql [ACK] Se

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)	
Internet Protocol Version 4, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)	
Transmission Control Protocol, Src Port: 52600 (52600), Dst Port: mysql (3306)	
Source port: 52600 (52600)	
Destination port: mysql (3306)	
[Stream index: 0]	

Para este ejemplo, dónde se protege una conexión entre Wordpress y una base de datos MySQL, hay que tener en cuenta el fichero wp-config.php, el cual tiene una serie de flags para la conexión con la base de datos. Se debe añadir, simplemente, la instrucción "define('MYSQL_CLIENT_FLAGS', MYSQL_CLIENT_SSL);". En la siguiente imagen se puede visualizar.

```
/*SSL*/  
define('MYSQL_CLIENT_FLAGS', MYSQL_CLIENT_SSL);  
  
/** MySQL hostname */  
define('DB_HOST', '192.168.56.106');  
  
/** Database Charset to use in creating database tables. */  
define('DB_CHARSET', 'utf8');
```

En este instante, las peticiones entre el sitio web de Wordpress y las consultas a MySQL van por un canal cifrado, por lo que la técnica de Network Packet Manipulation no puede funcionar, ya que no se puede observar que es lo que se está enviando. Ya no es válida.



4. Ataques a DHCP

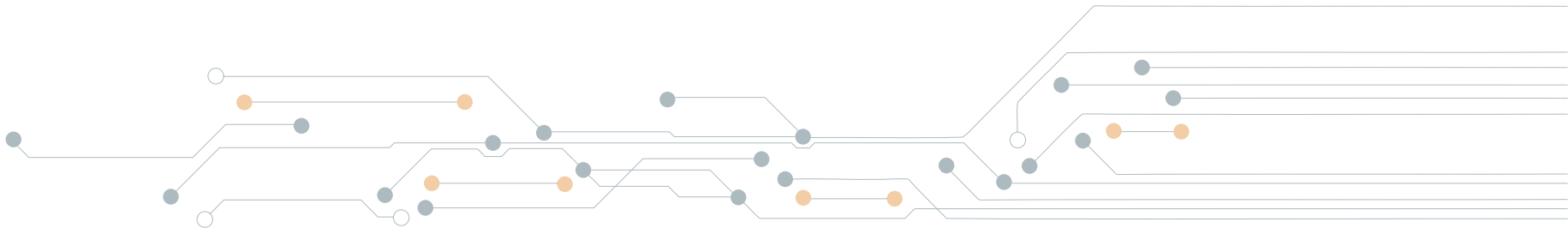
Existe una variante del ataque MITM que es bastante utilizada y se basa en la implementación de un servidor DHCP falseado. Debemos tener en cuenta que la funcionalidad del mismo es anárquica, de tal forma que la existencia en una red de dos servidores DHCP puede ocasionar, o bien la configuración adquirida por una máquina, o bien la facilitada por uno o por el otro. Básicamente el primero que responda a una petición formulada. El funcionamiento de dicho servicio en lo que respecta al proceso de petición es el siguiente:

- Envío por parte del cliente de un paquete DISCOVERY para que el servidor DHCP de dicha red de dispositivos le asigne una dirección IP y otros parámetros como la máscara de red o el DND.
- Respuesta por parte del servidor DHCP con un OFFER en el que detalla una serie de parámetros al cliente: IP, puertos, DNS,...
- Selección por parte del cliente de los parámetros que le interesan y con un REQUEST solicita estos parámetros al servidor.
- Reconocimiento por parte del servidor de que se ha reservado correctamente los parámetros solicitados con un DHCP ACK y se los envía al cliente.

El ataque simple se basa en implementar un servidor DHCP falso en la red, de forma que cuando el cliente envía una trama tipo DISCOVERY, responden con un OFFER tanto el DHCP real como el falso. El cliente atenderá al que antes envíe la respuesta DHCP OFFER.

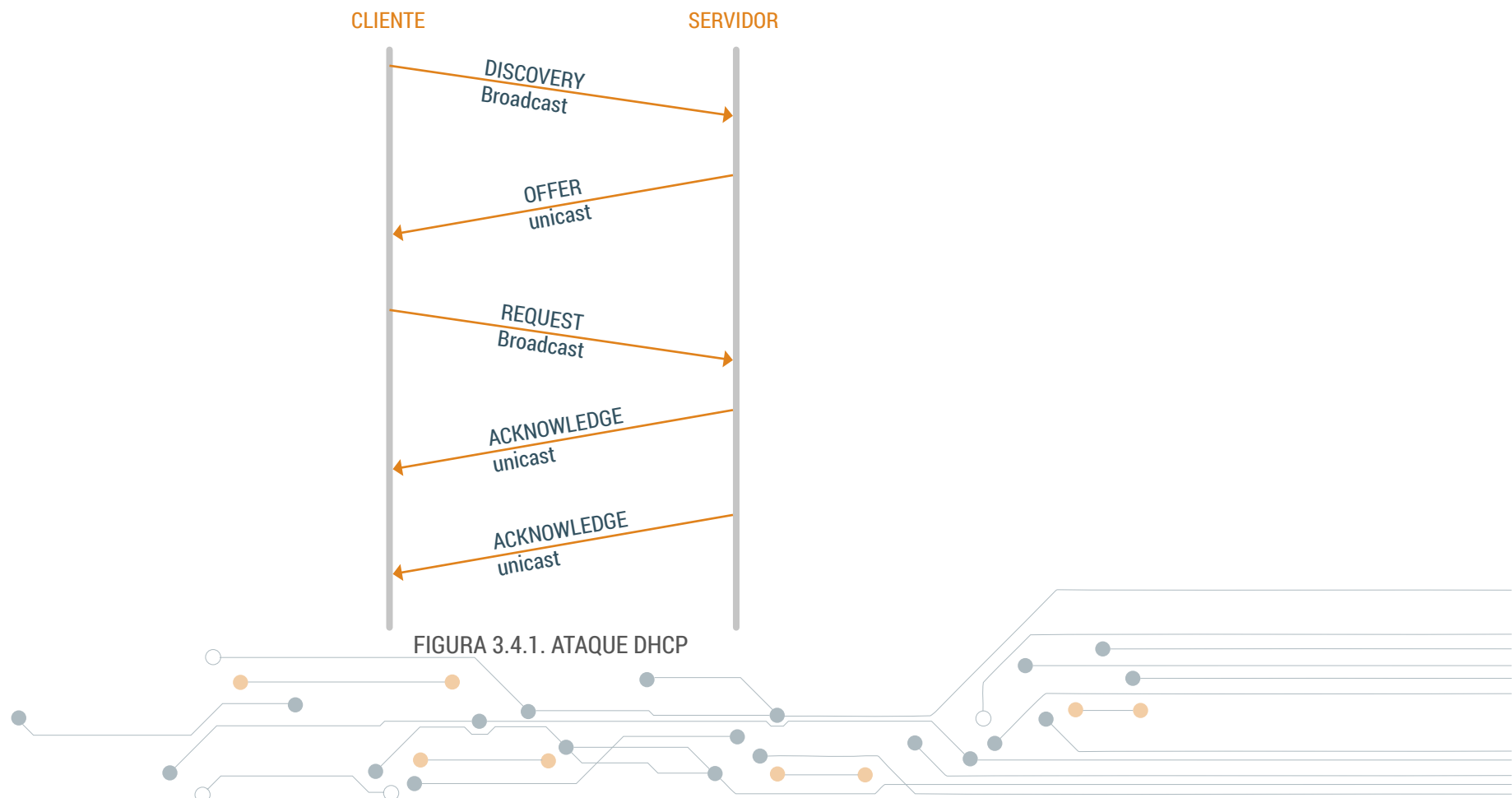
Un problema para el dispositivo del atacante es que no conoce en el inicio el rango de direcciones IP que se conceden ni las ya asignadas por el servidor DHCP real. Así, podría haber un conflicto entre Direcciones IP que el falso servidor crea, con las del servidor real. Para evitar que esto ocurra hay una posibilidad de ofrecer sólo información determinada y limitada del host: el ataque DHCP ACK injection.

Puesto que la comunicación DHCP se hace mandando los paquetes a la dirección MAC de broadcast FF:FF:FF:FF:FF:FF todos los clientes de la LAN reciben los paquetes DHCP. De esta forma existe la posibilidad de que un atacante monitorice los intercambios DHCP y en un punto de la comunicación envíe un paquete formado específicamente para cambiar su comportamiento.



Uno de los puntos donde interesaría intervenir es cuando el servidor reconoce con un DHCP ACK la configuración del cliente. Primero se tiene que escuchar toda la comunicación poniendo atención en el paquete REQUEST donde el cliente solicita la IP, DNS y Gateway entre otros de aquellos datos que anteriormente le ha ofrecido el servidor DHCP. Una vez recibido el REQUEST podría responderse con un ACK como lo haría el servidor DHCP real pero estableciendo la configuración a criterio del atacante.

La siguiente figura muestra cómo se produciría la transición de tramas para hacer efectivo el ataque:

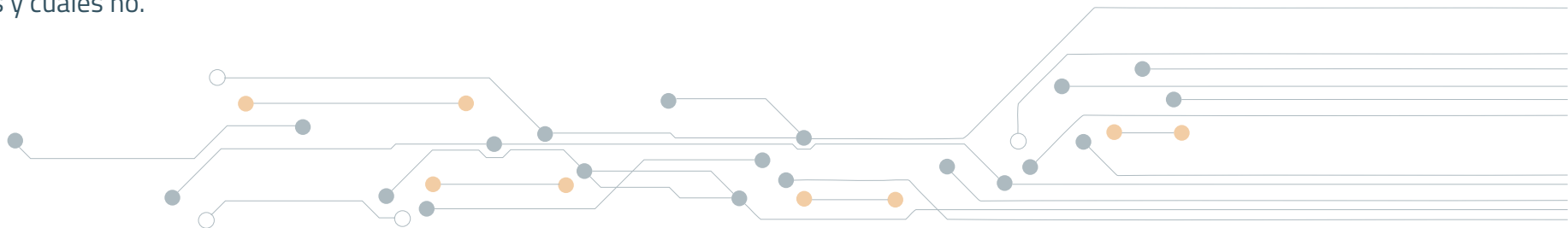


La ventaja de este ataque es que no necesita conocer el rango de direcciones IP válidas ni qué direcciones están libres y cuáles ocupadas. Se deja en manos del servidor DHCP real el que ofrezca toda esa información y sólo se interviene en la fase final, en el reconocimiento que da el servidor sobre la configuración seleccionada. También es más difícil de detectar. Sólo se envía un paquete y puede ser enviado con la IP suplantada del servidor DHCP.

Sin embargo, como en el anterior escenario existe la posibilidad de que la respuesta proceda tanto del atacante como del servidor DHCP real y el cliente sólo hará caso al primero de ellos que responda. Algunas veces será más rápido el servidor DHCP real, otras el atacante.

El atacante no tendrá que tener conocimiento ni de las IP dadas ni de las que pueden ofrecerse, sólo se facilitarán los parámetros necesarios para interceptar por ejemplo los paquetes dirigidos al router o plantear la base para un ataque de DNS Spoofing.

Como mecanismo de defensa pueden encontrarse aplicaciones tales como DHCP Probe, que cotejan en una base de datos los servidores DHCP legales que se habrán introducido con el tráfico generado por un falso servidor DHCP. Dicha aplicación lanza peticiones de DHCP evaluando la respuesta obtenida, indicando para ello qué servidores son los legítimos y cuáles no.



5. Otros tipos de Spoofing

5.1 | IP Spoofing

La técnica de IP *Spoofing*, como todos los Spoofing, consiste en engañar al que recibe la petición haciéndole creer que la dirección de origen es una, cuando realmente es otra. En otras palabras, si se enviara una carta a la casa del señor García, y en el remitente se pusiera señor Fernández, el señor García pensaría que la carta viene del señor Fernández, cuando en realidad es el señor González quien envió la carta. Esto es un símil válido para la técnica IP Spoofing.

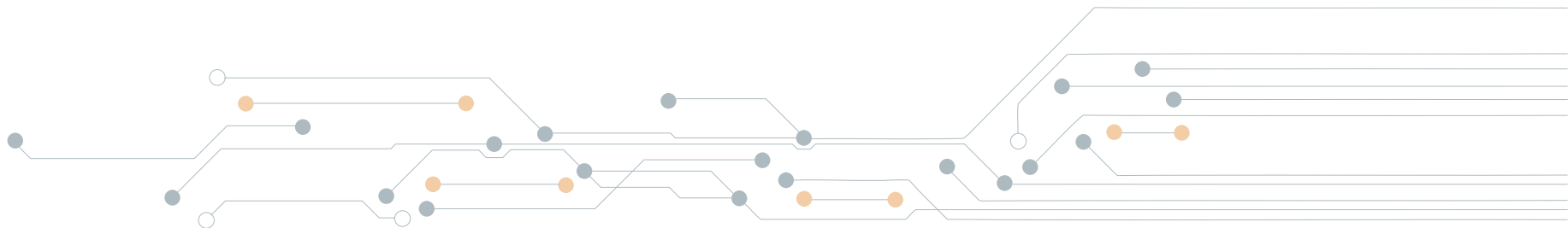
En otras palabras, la técnica de IP Spoofing permite falsificar la dirección IP origen de un paquete. Generalmente, está muy ligado al protocolo de transporte UDP, ya que dicho protocolo no está orientado a conexión, por lo que el equipo que recibe un paquete por UDP puede contestar directamente sin más. Cuando se trata del protocolo TCP, es más raro ver este tipo de técnicas, ya que, al estar orientado a conexión, no tendría demasiado sentido que se envíe un SYN con dirección IP falsa, ya que el equipo que recibe el flag de SYN contestará SYN+ACK o RST, según sea, a una máquina que no lo espera y que descartará el paquete. Es cierto, que existen usos con TCP, como, por ejemplo, un escaneo *zombie* o *idle*.

En el caso del protocolo UDP, es muy común ver la técnica de IP Spoofing orientada a ataques de denegación de servicio con ataques de amplificación. Un ataque de amplificación consiste en que, si la máquina A envía un paquete de 10 bytes, la máquina B conteste con un paquete de $n \times 10$ bytes, es decir, amplificado.

Como casos reales se pueden encontrar varios, pero se hablará de:

- El protocolo DNS.
- El protocolo NTP.

Un ataque de *DNS Amplification* consiste en que una serie de máquinas realizan consultas contra servidores DNS a través del protocolo de transporte UDP. El protocolo UDP no valida las direcciones IP de origen, de ello debería encargarse el protocolo de la capa de aplicación, por lo que si éste no lo realiza se tendrá un problema. Es realmente fácil falsificar un datagrama IP para configurar una dirección IP de origen falsa. Cuando muchos paquetes UDP tienen como dirección IP de origen una o una serie de máquinas en concreto, los servidores contestarán a esa dirección y se realizará



una acción de amplificación enorme contra ese o esos servidores. Esto se denomina una negación reflejada.

El ataque de *NTP Amplification* es exactamente igual que el anterior, solo que se utiliza el protocolo NTP en vez del protocolo DNS. Además, con la aparición de una vulnerabilidad en el comando *monlist* que permitía obtener un gran número de direcciones IP de los servidores o máquinas que habían consultado el presente servidor, se obtenía una gran amplificación.

- | | |
|-----------|------------|
| ■ DNS | ■ 28 to 54 |
| ■ NTP | ■ 556.9 |
| ■ SNMPv2 | ■ 6.3 |
| ■ NetBIOS | ■ 3.8 |
| ■ SSDP | ■ 30.8 |

En la siguiente relación, se tiene una serie de protocolos basados en UDP y el factor que se puede llegar a amplificar, es decir, en el caso de

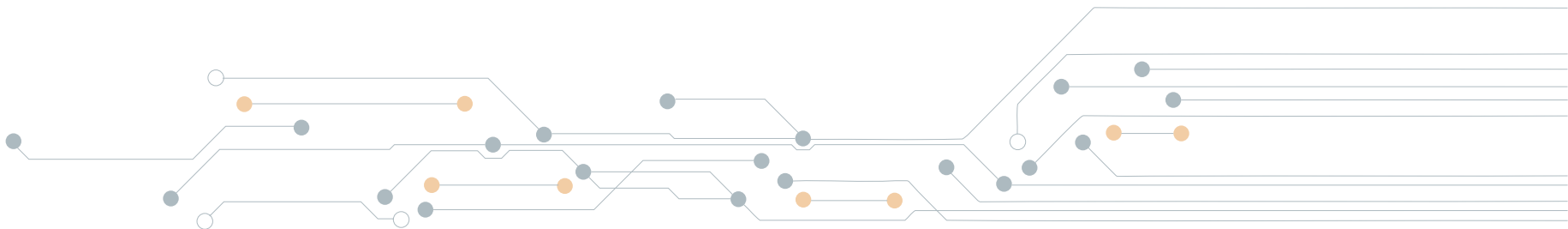
NTP, por cada *byte* que se envía se puede llegar a devolver hasta 556 bytes, lo cual es una potente arma para la denegación de servicio.

En la siguiente imagen, se puede visualizar una configuración de la herramienta *hping3*, con la que generar un ejemplo sencillo de IP Spoofing. Con *hping3*, se está indicando que la petición se enviará a una dirección IP, en este ejemplo a la dirección IP 127.0.0.1.

Con el parámetro *-a* se indica la dirección IP que se configurará realmente, es decir, la que se estará "*spoofeando*". El parámetro *-S* configura el flag de *SYN*, ya que en este ejemplo se está utilizando TCP como protocolo de transporte. El paquete es enviado al puerto 9000, esto se especifica con el parámetro *-p*, y con el parámetro *-c* se indica que solo se enviará un solo paquete.

```
root@kali:~# hping3 127.0.0.1 -a 8.8.8.8 -S -p 9000 -c 1
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes

--- 127.0.0.1 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~#
```



5.2 | DNS Spoofing

DNS Spoofing es un tipo de ataque con el cual, un atacante puede redirigirnos a un sitio web diferente al que nos queremos conecta y con intenciones diferentes a las nuestras. Conceptualmente estamos hablando de una suplantación de identidad por nombre de dominio, es decir, se trata de la manipulación de una relación Dominio-IP, resolviendo un cierto nombre DNS con una dirección IP falsa, o al contrario. El atacante consigue este objetivo falseando dicha relación aprovechando una vulnerabilidad del servidor o su confianza hacia servidores poco fiables. Además, dichas entradas son sensibles de envenenar la caché DNS de otro servidor DNS.

A continuación, se muestra un esquema gráfico de DNS Spoofing.

1. El atacante está en medio de la comunicación entre la víctima y el servidor DNS. Cuando la víctima envía una petición DNS al servidor legítimo, ésta es interceptada.
2. El atacante reenvía la respuesta DNS falsificada, es decir, la dirección IP que se devuelve a la víctima no corresponde con el dominio legítimo solicitado.
3. La víctima se conecta con un sitio malicioso, en vez de conectarse con el sitio web legítimo. La víctima se estará conectado a un sitio web dónde le podrían robar, entre otras cosas, las credenciales.

Para una mejor comprensión, se ejemplifica un ataque a una víctima con DNS Spoofing, en el que se la redirige a la dirección IP que el atacante quiere.

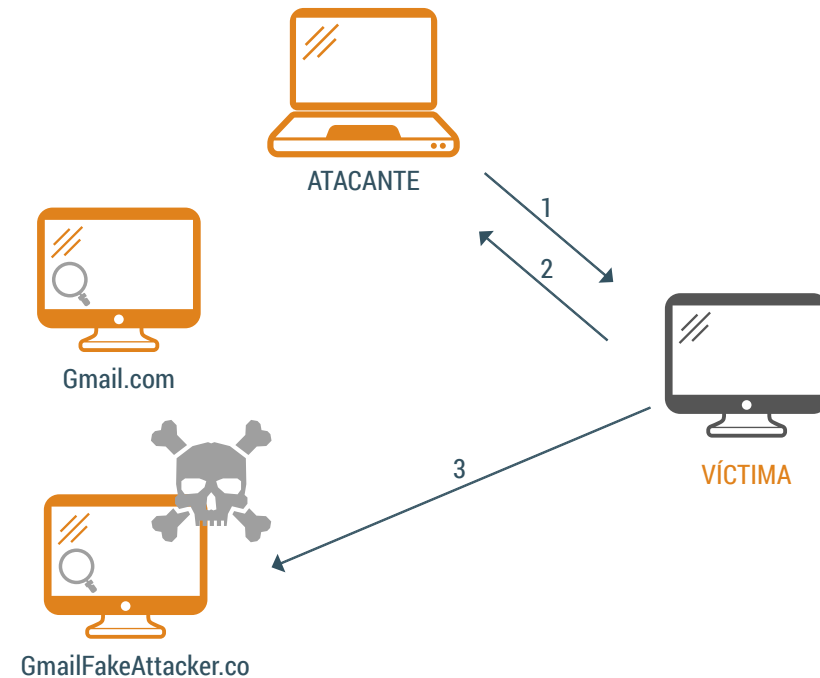
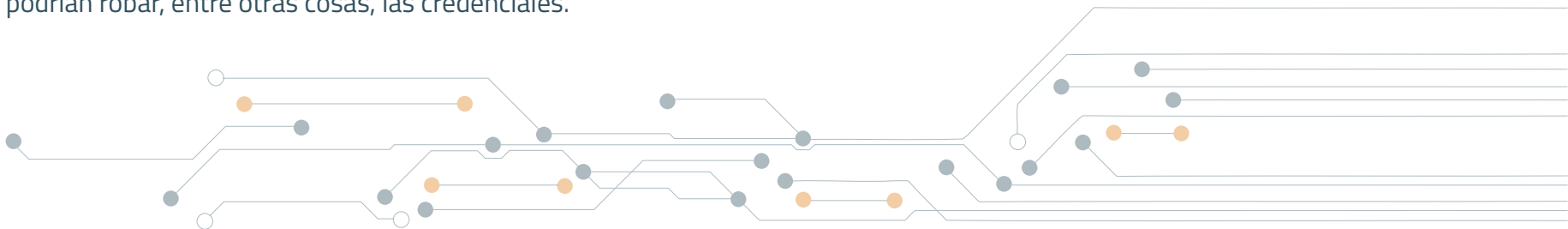


FIGURA 3.5.2.1. ESQUEMA DNS SPOOFING



ESCENARIO

- Una víctima navegando desde un ordenador Windows, utilizando un servidor DNS externo a la red local, el 8.8.8.8 (DNS de Google).
- Un atacante con Kali Linux.
- Ambos se encuentran en la misma LAN.

ATAQUE

El objetivo del atacante será interceptar las peticiones y respuestas DNS, construyendo un fichero hosts con la nueva IP deseada por el atacante. Por tanto, en primer lugar el atacante realizará un ataque Man In the Middle para interceptar dicho tráfico, mediante ARP Spoofing. Una vez que el atacante consigue acceder al tráfico construye un fichero host que se pasará a dnsspoof, con las IP acordes a los nombres que desea spoofear.

Editará el siguiente fragmento:

```
192.168.0.57 *.tuenti.com 192.168.0.57 tuenti.com
```

Cuando el atacante haya terminado de editar el fichero, deberá ejecutar la herramienta dnsspoof con el siguiente texto:

```
Dnsspoof -i -et0 -f <fichero>
```

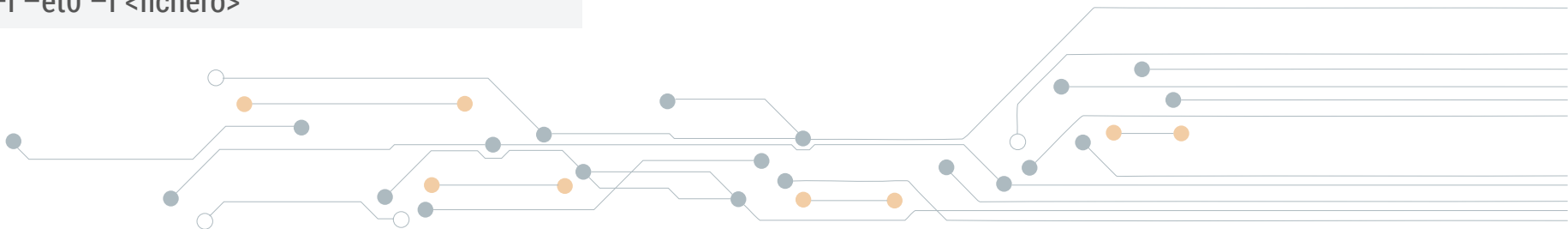
De una manera sencilla el atacante ha realizado un phishing casi perfecto. La víctima podrá detectar esto comprobando su caché. En ella descubriría que la resolución de nombres nueva se encuentra en una dirección IP local:

```
Nombre de registro . . : secure.tuenti.com
Tipo de registro . . . : 1
Tiempo de vida . . . . : 47
Longitud de datos . . . : 4
Sección . . . . . : respuesta
Un registro (host) . . : 192.168.0.57

static.tuenti.com
-----
Nombre de registro . . : static.tuenti.com
Tipo de registro . . . : 1
Tiempo de vida . . . . : 1
Longitud de datos . . . : 4
Sección . . . . . : respuesta
Un registro (host) . . : 192.168.0.57

www.stopbadware.org
-----
Nombre de registro . . : www.stopbadware.org
Tipo de registro . . . : 5
Tiempo de vida . . . . : 285
Longitud de datos . . . : 4
Sección . . . . . : respuesta
Registro CNAME . . . . : cf-ssl18624-protected-www.stopbadware.org

tuenti.com
-----
Nombre de registro . . : tuenti.com
Tipo de registro . . . : 1
Tiempo de vida . . . . : 1
Longitud de datos . . . : 4
Sección . . . . . : respuesta
Un registro (host) . . : 192.168.0.57
```



Telefonica EDUCACIÓN DIGITAL