Seguridad en Redes de Ordenadores
Práctica 6: Seguridad Perimetral: snort, nmap, iptables
versión 3.0

Eva M. Castro (eva.castro at urjc . es)
GSyC
Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Abril de 2018

Resumen

Esta práctica se va a realizar con el uso de 2 raspberry pi en las que hay que instalar una distribución kali y utilizar herramientas de sondeo de equipos y detección de intrusos.

1. Configuración previa

Se desea configurar un escenario como el que se muestra en la siguiente figura 1.

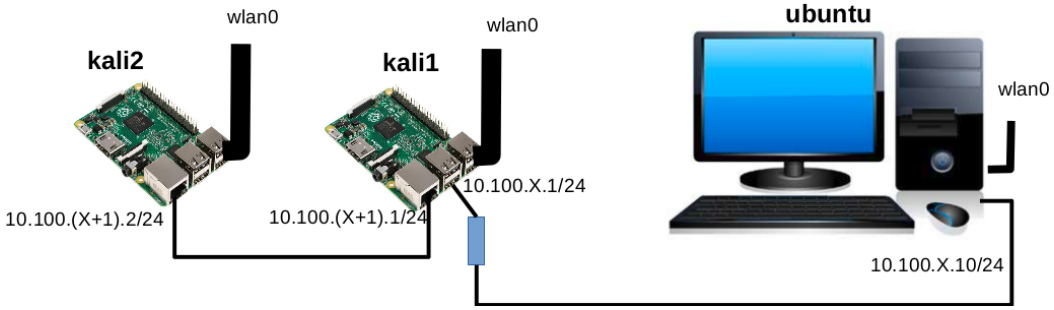


Figura 1: Configuración de red para realizar la práctica.

Donde tendremos 3 máquinas:

- Tu ordenador con una distribución **ubuntu** que debe estar conectado a 2 interfaces de red. La interfaz cableada estará directamente conectada a la raspberry pi `kali1` dentro de la subred 10.100.X.0/24. Además deberá tener conexión a Internet a través de la tarjeta inalámbrica. El ordenador tendrá configurada una ruta a la subred 10.100.(X+1).0/24 a través de 10.100.X.1 (`kali1`).
- La raspberry pi **kali1** que tendrá 3 interfaces de red. Una de ellas conectada a tu ordenador dentro de la subred 10.100.X.0/24 , otra de ellas conectada a la raspberry pi `kali2` dentro de la subred 100.100.(X+1).0/24 y la interfaz inalámbrica que deberá tener conexión a Internet. `kali1` estará configurada como router entre las subredes 10.100.X.0/24 y 10.100.(X+1).0/24.
- La raspberry pi **kali2** que tendrá 2 interfaces de red. Una de ellas conectada a la raspberry pi `kali1` dentro de la subred 100.100.(X+1).0/24 y la interfaz inalámbrica que deberá tener conexión a Internet. El `kali2` tendrá configurada una ruta a la subred 10.100.X.0/24 a través de 10.100.(X+1).1 (`kali1`).

Todas las pruebas de detección y sondeo en la red que realizaremos en esta práctica serán sobre las redes privadas 10.100.(X+1).0/24 y 10.100.X.0/24.

1.1. Copia la distribución kali en 2 tarjetas de memoria

Vamos a copiar la imagen de la distribución `kali` que se llama `kali-linux-2018.1a-rpi3-nexmon.img.xz` en tu portátil, se encuentra en las máquinas del laboratorio en la carpeta `/var/lib/vms` o también la puedes descargar de:

`https://images.offensive-security.com/arm-images/kali-linux-2018.1a-rpi3-nexmon.img.xz`

Una vez copiada en tu portátil, descomprime esta imagen con el siguiente comando:

```
xz -d kali-linux-2018.1a-rpi3-nexmon.img.xz
```

Mete la tarjeta microSD en tu portátil. Para saber el nombre del dispositivo que se corresponde con la tarjeta miniSD en tu ordenador ejecuta el siguiente comando:

```
df -h
```

Este comando mostrará todas las particiones que hay montadas en la máquina, hay que localizar la que se corresponde con la memoria miniSD. En particular, en Linux será algo como: `/dev/sdX1` montada en `/media/nombreUsuario`. Si la tarjeta ya tenía grabado algo previamente, pueden aparecer varios puntos de montaje: `/dev/sdX1`, `/dev/sdX2`, etc. Primero es necesario desmontar todas ellas, por ejemplo:

```
umount /dev/sdX1
```

Para copiar esta distribución ejecuta en tu portátil el siguiente comando, teniendo en cuenta que `<DIR>` es el nombre de la carpeta donde está almacenada la imagen de kali y `<DEV>` es el nombre del dispositivo `sdX` en tu ordenador **sin el número**. Es importante que seas especialmente cuidadoso con estas instrucciones para no borrar alguna de las particiones de tu ordenador:

```
sudo dd bs=4M if=<DIR>/kali-linux-2018.1a-rpi3-nexmon.img of=/dev/<DEV>
```

Mete la segunda tarjeta microSD y repite los pasos anteriores para copiarla: desmonta los puntos de montaje de la tarjeta microSD y realiza la copia.

1.2. Configuración inicial kali1

Arranca la primera raspberry, a la que vamos a llamar `kali1`. La raspberry tiene conectado un cable TTL serial que permite tener una consola a través de su puerto serie conectado a un puerto USB de tu ordenador, véase la figura 1.2.

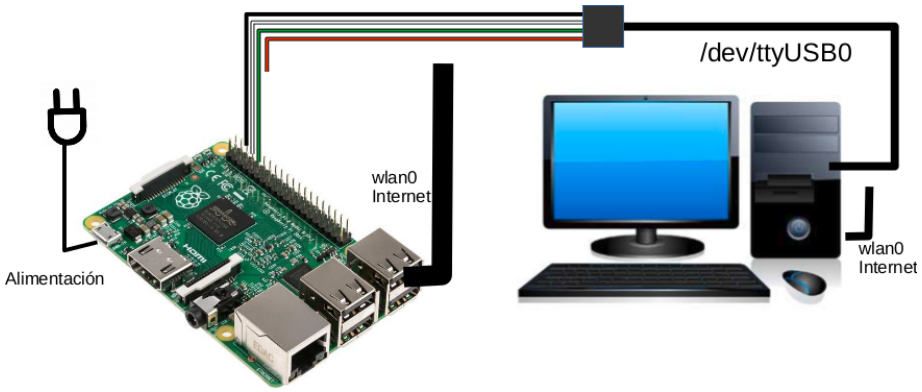


Figura 2: Conexión de la raspberry pi con un ordenador a través del cable TTL serial al puerto USB.

Para tener una consola serie necesitas usar el programa `screen` en tu portátil (instálalo si no lo tienes) y ejecuta el siguiente comando:

```
sudo screen  /dev/ttyUSB0 115200
```

Este programa abrirá una consola serie con la raspberry pi, entra con nombre de usuario `root`, contraseña `toor`.

Es necesario configurar las direcciones IP de `kali1`: puertos `eth0` y `eth1` en el fichero `/etc/network/interfaces` variando los valores de X por los que se te asignaron en las prácticas anteriores:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 10.100.X.1
   netmask 255.255.255.0

auto eth1
iface eth1 inet static
   address 10.100.X+1.1
   netmask 255.255.255.0
```

```
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Además, se desea que la raspberry pi tenga acceso a través de la configuración de la red inalámbrica de la universidad y para ello hay que configurar el fichero `/etc/wpa_supplicant/wpa_supplicant.conf`:

```
network={
    ssid="eduroam"
    scan_ssid=1
    key_mgmt=WPA-EAP
    pairwise=CCMP
    group=CCMP TKIP
    eap=PEAP
    ca_cert="/etc/ssl/certs/ca.pem"
    identity="alumno@alumnos.urjc.es"
    domain_suffix_match="urjc.es"
    phase1="peaplabel=0"
    phase2="auth=MSCHAPV2"
    password="PASS"
}
```

Ten en cuenta que hay que dejar el fichero `ca.pem` con el certificado de la autoridad de certificación en la carpeta `/etc/ssl/certs/`. Descarga el fichero en tu ordenador desde el aula virtual y con el ratón copia exactamente el contenido en el fichero `/etc/ssl/certs/ca.pem`.

Para usar la red inalámbrica de tu casa, dependerá de la configuración que tengas, pero es habitual usar WPA-PSK en cuyo caso el fichero de configuración `/etc/wpa_supplicant/wpa_supplicant.conf` debería contener:

```
network={
    ssid="nombreDeLaRedEnTuCasa"
    psk="contraseña"
    key_mgmt=WPA-PSK
}
```

Cambia el nombre a la raspberry para que la indentifiquemos como `kali1`, edita el fichero `/etc/hostname` y modifícalo para que tenga el siguiente contenido[1]:

```
kali1
```

Además como en el caso de `kali1` se desea que funcione como router es necesario activar el reenvío, para ello edita el fichero `/etc/sysctl` y añade la siguiente línea[2]:

```
net.ipv4.ip_forward=1
```

Cambia el passwd de `root`, ya que tiene el valor por defecto a `toor` y lo vas a conectar a Internet.

Hay un problema con la hora, la distribución kali tiene configurada la hora en la que se creó la distribución, diciembre de 2017 y el certificado para conectarse a la red inalámbrica aún no es válido en esa fecha, por tanto hay que cambiar la hora a `kali1`. Para ello vamos a configurar que cada vez que se reinicie la máquina se configure una hora más actual:

```
crontab -e

@reboot date --set "04/17/2018 11:00"
```

Ejecuta `reboot` para que la configuración tenga efecto y vuelve a entrar a través del puerto serie. La raspberry se habrá conectado a la red inalámbrica eduroam, en su interfaz `wlan0`. Apunta la dirección IP que te han asignado por DHCP. Esta dirección IP te permitirá conectarte a la raspberry pi, de forma más cómoda y con tantos terminales como necesites, a través de `ssh` desde tu portátil que también deberá estará conectado a la red eduroam. Ten en cuenta que cada vez que inicies la raspberry le podrán asignar una dirección IP diferente a su interfaz inalámbrica.

Ahora que ya tienes conectada `kali1` a Internet, ejecuta:

```
apt-get update
apt-get install tcpdump
apt-get install snort
apt-get install netcat
apt-get install nmap
```

[1] Cuando reinicies la raspberry observarás que ha cambiado el nombre de la máquina

[2] Una vez que reinicies la raspberry podrás comprobar que la configuración se ha aplicado correctamente ejecutando el siguiente comando cuyo resultado debe ser 1: cat /proc/sys/net/ipv4/ip_forward

1.3. Configuración inicial kali2

Arranca la segunda raspberry pi y conéctate también utilizando el cable TTL serial. Edita su fichero `/etc/network/interfaces` con la siguiente configuración, para que tenga una dirección IP en la misma subred de `kali1` y además utilice a `kali1` para alcanzar la subred 10.100.X.0/24:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
   address 10.100.X+1.2
   netmask 255.255.255.0

allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

up route add -net 10.100.X.0/24 gw 10.100.X+1.1
```

Realiza la misma configuración en `/etc/wpa_supplicant/wpa_supplicant.conf` que en `kali1`.

Cambia el nombre a esta segunda raspberry pi para identificarla como `kali2`. Cambia el passwd de `root` y la hora.

Ejecuta `reboot` para que la configuración tenga efecto y vuelve a entrar a través del puerto serie. La raspberry se habrá conectado a la red inalámbrica eduroam, en su interfaz `wlan0`. Apunta la dirección IP que te han asignado por DHCP. Esta dirección IP te permitirá conectarte a la raspberry pi, de forma más cómoda y con tantos terminales como necesites, a través de `ssh` desde tu portátil que también deberá estará conectado a la red eduroam. Ten en cuenta que cada vez que inicies la raspberry le podrán asignar una dirección IP diferente a su interfaz inalámbrica.

1.4. Configuración en tu ordenador

Configura la dirección IP 10.100.X.10/24 en tu ordenador a través de la interfaz cableada que tendrás conectada a `kali1`. También debes configurar una ruta a la subred 10.100.(X+1).0/24 a través del router 10.100.X.1 (`kali1`), esta ruta te permitirá alcanzar la máquina `kali2`.

Para ello puedes usar el botón "Configuración del sistema" que hay en la barra de aplicaciones de ubuntu y seleccionar "Red", a continuación "Cableada". Pulsa sobre el botón "Opciones" y "Ajustes de IPv4".Una vez allí pulsa sobre "Añadir" para añadir una dirección IP y rellena los campos:

- Dirección: 10.100.X.10
- Máscara de red: 255.255.255.0
- Puerta de enlace: deja este campo vacío[3].

Pulsa sobre el botón "Rutas" y añade una ruta para alcanzar la red 10.100.(X+1).0/24 a través de `kali1`. Rellena con los siguientes campos:

- Dirección: 10.100.(X+1).0
- Máscara de red: 255.255.255.0
- Puerta de enlace: 10.100.X.1.

Comprueba que puedes hacer un ping a la máquina `kali2` desde tu ordenador.

2. Snort

La máquina `kali1` va a ejecutar una herramienta IDS, `snort`, que detecta accesos potencialmente maliciosos y los registra en un fichero de log. Cuando `snort` descubra tráfico potencialmente malicioso escribirá una alerta en un fichero de logs y almacenará el tráfico malicioso en un fichero de captura. Estos ficheros se encontrarán en la carpeta `/var/log/snort`.

[3]No queremos rellenar ruta por defecto porque nuestro ordenador se conectará a Internet a través de la interfaz inalámbrica y no a través de la red cableada

2.1. Reglas Snort

En la carpeta `/etc/snort/rules/` se describen reglas predefinidas en `snort` para la inspección de tráfico. Dependiendo de la configuración del fichero `/etc/snort/snort.conf` se podrán cargar las reglas que se desean aplicar al tráfico que el IDS examine.

Debido a que no vamos a utilizar algunas de las reglas que se cargan por defecto en `snort`, vamos a comentar el uso de estas reglas para que el IDS no esté comprobándolas. Edita el fichero `/etc/snort/snort.conf` y comenta las siguientes líneas añadiendo el carácter # al principio de cada una de ellas:

```
#include $RULE_PATH/web-activex.rules
#include $RULE_PATH/web-attacks.rules
#include $RULE_PATH/web-cgi.rules
#include $RULE_PATH/web-client.rules
#include $RULE_PATH/web-coldfusion.rules
#include $RULE_PATH/web-frontpage.rules
#include $RULE_PATH/web-iis.rules
#include $RULE_PATH/web-misc.rules
#include $RULE_PATH/web-php.rules
#include $RULE_PATH/x11.rules
#include $RULE_PATH/community-sql-injection.rules
#include $RULE_PATH/community-web-client.rules
#include $RULE_PATH/community-web-dos.rules
#include $RULE_PATH/community-web-iis.rules
#include $RULE_PATH/community-web-misc.rules
#include $RULE_PATH/community-web-php.rules
#include $RULE_PATH/community-sql-injection.rules
#include $RULE_PATH/community-web-client.rules
#include $RULE_PATH/community-web-dos.rules
#include $RULE_PATH/community-web-iis.rules
#include $RULE_PATH/community-web-misc.rules
#include $RULE_PATH/community-web-php.rules
```

A continuación responde a las siguientes preguntas en la memoria de la práctica:

1. El fichero `/etc/snort/snort.conf` es el que contiene la configuración de snort. Este fichero está dividido en varias partes, las líneas que comienzan por # son comentarios. Nosotros nos vamos a fijar en la parte de la configuración de variables (parte 1 ó *Step #1*) y en la parte de configuración de reglas (parte 7 ó *Step #7*). Escribe en la memoria el contenido de las variable `HOME_NET` y `EXTERNAL_NET` y explica qué crees que significa ese valor.
2. En la sección *Step #7* se incluyen la reglas escritas en diversos ficheros que se encuentran en la carpeta `/etc/snort/rules`, en particular comprueba que se inlcuye el fichero `/etc/snort/rules/icmp.rules`. Abre este fichero. Las líneas que comienzan por # son comentarios, las reglas están escritas cada una en una única línea. Incluye la última regla de ese fichero en la memoria y explica el contenido [4].
3. Busca en el fichero `icmp.rules` la regla que es una alerta que escribe el mensaje `"ICMP PING NMAP"`[5], inclúyela en la memoria y explica todo lo que puedes saber de su contenido.
4. Busca en el fichero `icmp-info.rules` la regla que es una alerta que escribe el mensaje `ICMP PING *NIX"`, inclúyela en la memoria y explica todo lo que puedes saber de su contenido.
5. Explica cuál es la diferencia entre ambas reglas y el nivel de prioridad de cada una de ellas. ¿Por qué una tiene mayor prioridad que otra?
6. Lanza `snort` en la máquina `kali1` (`snort -A console -c /etc/snort/snort.conf -i eth1`) para que comience a detectar tráfico potencialmente peligroso y déjalo lanzado para que te vaya mostrando las alertas que detecte en los apartados sucesivos. Una vez arrancado informará de que la inicialización se ha completado e indicará el número de proceso (pid), apúntalo para que, en caso de que sea necesario, puedas matar el proceso.

2.2. Alertas Snort

Vamos a realizar algunas pruebas sencillas para ver cómo se activan las notificaciones en snort.

1. Desde `kali2` ejecuta un `ping a 10.100.X.10` con el envío de un único paquete. Observa las alertas que muestra snort y ve al fichero de definición de esa/s regla/s, copia la/s regla/s y explica qué condiciones se han cumplido para que se activen.
2. Desde `kali2` vuelve a ejecutar el mismo ping pero con tamaño de paquete 1000 bytes (`-s 1000`). Observa las alertas que muestra snort y ve al fichero de definición de esa/s regla/s, copia la/s regla/s y explica qué condiciones se han cumplido para que se activen.

[4] Puedes consultar la información sobre classtype en la sección 3.4.6 del manual: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node31.html

[5] Nmap es una aplicación que realiza escaneo de redes, aplicaciones y servicios. Se utiliza para Pentesting.

3. Desde `kali2` vuelve a ejecutar el mismo ping pero con tamaño de paquete 0 bytes (`-s 0`). Observa las alertas que muestra snort y ve al fichero de definición de esa/s regla/s, copia la/s regla/s y explica qué condiciones se han cumplido para que se activen.
4. En la carpeta `/var/log/snort` se quedan almacenados ficheros `snort.log.*`. Estos ficheros contienen los paquetes que han generado las alertas que se han mostrado en snort. Interpreta estos ficheros cargándolos con tcpdump y la opción `-r <nombreFichero>`.

3. Pentesting con nmap

Si has interrumpido la ejecución de `snort` en la máquina `kali1` vuelve a lanzarlo:

```
snort -A console -c /etc/snort/snort.conf -i eth1
```

para que comience a detectar tráfico potencialmente peligroso y déjalo lanzado para que te vaya mostrando las alertas que detecte en los apartados sucesivos.

Existen diversas técnicas para el descubrimiento de equipos utilizando `nmap`. El objetivo es mostrar si la máquina se encuentra activa o no. A continuación se muestran algunas formas de sondeo utilizando `nmap`:

3.1. Sondeo de equipos

1. Desde `kali2` ejecuta `nmap` para sondear qué equipos están activos en la subredes 10.100.(X+1).0/24. Realiza una captura en `kali2(eth0)` con la opción `-n`[6] y guarda el contenido en un fichero `nmap-01.cap` y después arranca el sondeo en `kali2`.
   a) Explica la salida que muestra nmap.
   b) Interrumpe la captura y explica qué paquetes se han enviado y cuáles tienen respuesta.
   c) ¿Se muestra alguna alerta en snort? Explica tu respuesta.
2. Desde `kali2` ejecuta `nmap` para sondear un único equipo de su misma subred, `kali1`. Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-02.cap` y después arranca el sondeo en `kali2`.
   a) Explica la salida que muestra nmap.
   b) Interrumpe la captura y explica qué paquetes se han enviado y cuáles tienen respuesta.
   c) ¿Se muestra alguna alerta en snort? Explica tu respuesta.
3. Desde `kali2` ejecuta `nmap` para sondear un único equipo de otra subred diferente, tu máquina. Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-03.cap` y después arranca el sondeo en `kali2`.
   a) Explica la salida que muestra nmap.
   b) Interrumpe la captura y explica qué paquetes se han enviado y cuáles tienen respuesta.
   c) ¿Se muestra alguna alerta en snort? Explica tu respuesta.

3.2. Sondeo TCP SYN

Realiza una captura en `lkali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-04.cap`. Utiliza `nmap` desde `kali2` de la siguiente forma para que se envíen segmentos TCP con el flag de SYN activo con el objetivo de determinar si hay un servicio esperando recibir paquetes entre los puertos 1 a 50 de `kali1`.

1. Explica la salida que muestra nmap.
2. Interrumpe la captura y explica los paquetes intercambiados. Explica las diferencias del sondeo del puerto 22 y el resto de puertos.
3. Explica si `snort` ha detectado alertas e indica cuáles y por qué.
4. Consulta los servicios TCP activos (los servidores que se encuentran esperando paquetes TCP) en la máquina `kali1` a la que estabas realizando el sondeo, utilizando el comando `netstat -nt4l`. Relaciona el resultado de la ejecución de este comando con el resultado del sondeo.
5. Una vez encontrado un puerto abierto, puede ser útil obtener información de la versión del servicio que se está ejecutando. Prueba a realizar el sondeo anterior únicamente en el puerto que has encontrado abierto y añadiendo la opción `-sV`. Previamente a realizar el sondeo realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-05.cap`. Estudia el resultado de nmap y el contenido de la captura y explícalos.

[6] Esta opción se utiliza para que tcpdump no envíe paquetes de DNS para solicitar la resolución de direcciones IP a nombres y mostrar la información de forma más legible. Como en el escenario de pruebas no tenemos servidor de DNS para las máquinas involucradas, es mejor utilizar esta opción.

6. Vamos a arrancar un servidor de web en la máquina `kali1` escuchando peticiones HTTP en el puerto 80. Para que este servidor sólo use IPv4 hay que modificar el siguiente fichero `/etc/apache2/ports.conf` y cambiar la línea `Listen 80` por `Listen 0.0.0.0:80`. Inicia el servidor con el siguiente comando:
   `/etc/init.d/apache2 start`
   Ejecuta `netstat` igual que antes para comprobar que se encuentra arrancado este servicio en el puerto 80. ¿Qué crees que ocurrirá si se sondea nuevamente la máquina con el rango de puertos 20-100?
7. Prueba a realizar el sondeo anterior en el puerto 80 añadiendo la opción `-sV`. Previamente a realizar el sondeo realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-06.cap`. Estudia el resultado de nmap y el contenido de la captura y explícalos.

3.3. Sondeo UDP

Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-07.cap`. Utiliza `nmap` desde `kali2` de la siguiente forma para que se envíen paquetes UDP con el objetivo de determinar si hay un servicio esperando recibir paquetes entre los puertos 60 a 70 de `kali1`.

1. Explica la salida que muestra nmap.
2. Interrumpe la captura y explica los paquetes intercambiados. Explica las diferencias del sondeo del puerto 68 y el resto de puertos.
3. Explica si `snort` ha detectado alertas e indica cuáles y por qué.
4. Consulta los servicios UDP activos (los servidores que se encuentran esperando paquetes UDP) en la máquina `kali1` a la que estabas realizando el sondeo, utilizando el comando `netstat -nu4l`. Relaciona el resultado de la ejecución de este comando con el resultado del sondeo.

3.4. Sondeo TCP FIN, Xmas

Este sondeo consiste en enviar diferentes segmentos TCP que tengan activos determinados flags:

- FIN: únicamente flag FIN (`-sF` en vez de `-sS`)
- Xmas: activa FIN, PSH y URG (`-sX` en vez de `-sS`)

La RFC de TCP (RFC-793) no está totalmente definida para ciertas ocasiones inesperadas, como por ejemplo la activación de flags inesperados en determinados momentos. Dependiendo de los SO se pueden responder diferentes tipos de paquetes.

1. Desde `kali2` ejecuta `nmap` para realizar un ataque FIN TCP. Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-08.cap` y después arranca el sondeo en `kali2` para los puertos 20-25.
   a) Explica la salida que muestra nmap.
   b) Interrumpe la captura y explica qué paquetes se han enviado y cuáles tienen respuesta.
   c) ¿Se muestra alguna alerta en snort? Explica tu respuesta.
2. Desde `kali2` ejecuta `nmap` para realizar un ataque XMAS TCP. Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-09.cap` y después arranca el sondeo en `kali2` para los puertos 20-25.
   a) Explica la salida que muestra nmap.
   b) Interrumpe la captura y explica qué paquetes se han enviado y cuáles tienen respuesta.
   c) ¿Se muestra alguna alerta en snort? Explica tu respuesta.

3.5. Sondeos exhaustivos

Los sondeos anteriores aportan información concreta sobre un determinado aspecto de la máquina. Sin embargo, nmap permite realizar sondeos más agresivos que aportan información exhaustiva de una máquina. Esto requiere enviar muchos mensajes más para detectar toda la información posible. Generalmente llevan más tiempo y pueden alertar a los sistemas IDS. Para activar este tipo de sondeo se arranca `nmap -A -n <dirIP>`.

1. Desde `kali2` vamos a realizar un análisis exhaustivo de `kali1`. Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-10.cap` y después arranca el sondeo en `kali2` hacia la máquina `kali1`. Explica el resultado que muestra nmap. Carga la captura y comenta algún aspecto relevante que veas.
2. Desde `kali2` vamos a realizar un análisis exhaustivo de tu ordenador en la interfaz `10.100.X.10`. Realiza una captura en `kali2(eth0)` con la opción `-n` y guarda el contenido en un fichero `nmap-11.cap` y después arranca el sondeo en `kali2` hacia tu máquina. Explica el resultado que muestra nmap. Carga la captura y comenta algún aspecto relevante que veas.

4. firewall: impedir sondeos y ataques

Vamos a configurar las reglas de iptables en `kali1` para tratar de evitar que una prueba de pentesting o un escaneo intrusivo a nuestro ordenador ubuntu pueda mostrar información de los servicios que tenemos activos.

4.1. Sondeo FIN

1. Si no tienes un servidor de ssh en tu ordenador, lanza uno utilizando el comando `/etc/init.d/ssh start`[7]. Comprueba con `netstat` que en tu máquina hay una aplicación servidor TCP esperando recibir mensajes en el puerto 22 (el puerto 22 es el puerto de ssh). Comprueba también que no tienes aplicaciones servidor esperando recibir mensajes en los puertos de FTP (20 es el de ftp-data y 21 es el de control de ftp).
2. Desde `kali2` usando nmap realiza un sondeo TCP FIN a tu ordenador en el puerto 22. Añade la opción `--reason` para que informe del motivo por el cuál califica un determinado puerto con ese estado. Explica cómo califica nmap a estos puertos, teniendo en cuenta lo que has aprendido en los apartados anteriores.
3. Observa la configuración por defecto de iptables en `kali1`: `iptables -L -n -v`.
4. Vamos a añadir una regla de iptables en `kali1` que se cumpla cuando el sistema de seguimiento de conexiones detecte en la cadena FORWARD una conexión inválida (condición `-m conntrack --ctstate INVALID`) y cuya acción sea escribir un mensaje en el fichero de log. El mensaje de log contendrá los campos más importantes del paquete que ha satisfecho la regla, en este caso de un paquete inválido.

   iptables genera por defecto los mensajes de log en `kali` dentro del fichero `/var/log/syslog`. Para diferenciar los mensajes del sistema de los que estamos generando con iptables podemos utilizar un prefijo que se antepone a un mensaje generado por una determinada acción de LOG, para ello hay que usar la opción `--log-prefix`: `-j LOG --log-prefix "IPTables:: paquete INVALIDO "`

   Configura la regla en iptables para que cualquier paquete en estado inválido ejecute la acción de escribir un mensaje en el fichero de log del sistema. Observa que has configurado correctamente la regla en `iptables` y lanza el mismo sondeo TCP FIN anterior.

   Explica los mensajes que se han generado en el fichero de log del sistema como consecuencia de haber configurado esa regla.

5. Borra la configuración de iptables anterior y genera un script para la configuración de iptables en `kali1`. Normalemente un script de este tipo primero tiene la configuración del borrado de reglas que hubiera previamente y el reinicio de contadores de paquetes e inicialización de las políticas por defecto en las cadenas predefinidas. Lo habitual es ser restrictivo y tener políticas por defecto DROP, permitiendo sólo aquellos paquetes que cumplan las reglas definidas en el firewall, sin embargo, el objetivo de nuestra práctica sólo va a ser detectar paquetes que pueden ser potenciales sospechosos y por ello, dejaremos las políticas predefinidas con la acción ACCEPT, centrándonos únicamente en la definición de reglas que actúen sobre esta detección de paquetes. A continuación tiene la configuración que se desea realizar. El script deberá crear una cadena nueva de usuario en `kali1` de forma que los paquetes de conexiones TCP ejecuten esta nueva cadena. La nueva cadena detectará conexiones inválidas y quederán registradas en el fichero de log, a continuación borrará esos paquetes para que no se reenvíen. Incluye el script en la memoria.
6. Aplica la configuración anterior y captura el tráfico en la interfaz 10.100.X.10 de tu máquina `fw-01.cap`. Realiza el mismo ataque TCP FIN. Explica el resultado de nmap y los paquetes capturados. Explica también el contenido del fichero log.
7. ¿Qué crees que ocurriría si con esa configuración de iptables se realizase un ataque similar pero con segmentos SYN?

4.2. Sondeo ACK

Con la misma configuración de firewall en `kali1` que has realizado para el apartado anterior vamos a realizar un sondeo utilizando segmentos con el flag ACK activado. Para ello hay que usar en nmap la opción `-sA` en vez de `-sF`, como se estaba usando en el sondeo de FIN.

1. Ejecuta nmap hacia la dirección IP 10.100.X.10 (tu ordenador) desde `kali2` y explica el resultado [8]

[7] Si no tienes un servidor de ssh instalado, puedes instalar el paquete: openssh-server

[8] Los segmentos con ACK activado que no pertenecen a conexiones previas no se consideran inválidos porque pueden pertenecer a retardos en la red o rearranque de máquinas, etc

2. Para poder ser más restrictivo con respecto a los segmentos TCP que se permiten a través del firewall, se puede configurar una regla que compruebe la siguiente condición: aquellos paquetes que no lleven el flag SYN activo y pertenezcan a una conexión nueva dentro del sistema de seguimiento de conexiones son sospechos de ser sondeos o ataques y por tanto los vamos a registrar en el log y los vamos a descartar (`! --syn -m conntrack --ctstate NEW`). Configura esta nueva regla en tu script dentro de la cadena de usuario que has creado. Incluye este script en la memoria.
3. Vuelve a realizar la misma prueba de sondeo ACK y explica el resultado de nmap y los mensajes del log que son resultado de la ejecución de esta prueba.

4.3. SYN Flood

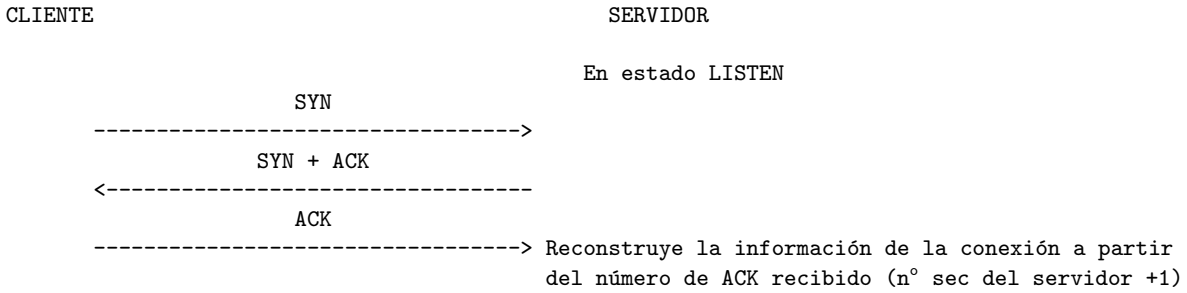
Es difícil poder saber si un segmento SYN es de una conexión legítima o pertenece a un sondeo. Sin embargo si lo que se está produciendo es un intento de ataque DoS (denegación de servicio) se puede intentar mitigar su efecto. Se pueden usar varios mecanismos, entre ellos vamos a ver dos que se pueden utilizar en Linux.

4.3.1. SYN cookies

Cuando un servidor recibe un segmento de SYN para el establecimiento de la conexión, el servidor responde con SYN+ACK y guarda información de dicha conexión (por ejemplo el número de secuencia con el que responde al cliente) en una cola de SYN pendientes hasta que se recibe el ACK de establecimiento completo de la conexión. Este comportamiento puede ser aprovechado por un atacante que comienza a enviar segmentos SYN al servidor para llenar la cola de conexiones pendientes, de forma que nunca llega a terminar el proceso de establecimiento de la conexión y el servidor deja de proporcionar servicio a clientes legítimos.

El mecanismo SYN cookies funciona de la siguiente forma, el servidor cuando recibe SYN responde con SYN+ACK y elimina la información de dicha conexión de la cola SYN pendientes de asentimiento. El servidor a partir del ACK que envía el cliente para completar el establecimiento de la conexión es capaz de reconstruir la información de dicha conexión. Para ello el servidor habrá elegido astutamente el número de secuencia que va a usar en esa conexión con el cliente, donde los 32 bits del número de secuencia codifican más información:

- 5 bits más significativos: es el timestamp modulo 32
- 3 bits siguientes: un valor que representa a MSS.
- 24 bits siguientes: número de secuencia incial.



Para ver como funciona, deshabilita en `kali1` el funcionamiento SYN cookies y configura el valor máximo de la cola de conexiones pendientes a 16:

```
sysctl -w net.ipv4.tcp_syncookies=0
sysctl -w net.ipv4.tcp_max_syn_backlog=16
```

Realiza una captura de tráfico en `kali1` en la interfaz que está conectada a `kali2`. En un terminal de `kali1` vamos a observar como las conexiones se van a mantener abiertas durante el ataque SYN flood pero sólo aquellas que ocupan el tamaño máximo de la cola (`tcp_max_syn_backlog`). Para ello usa el siguiente comando que muestra el estado de las conexiones TCP de una máquina cada 0,2 segundos:

```
kali2:~# watch -n 0,2 netstat -nt4
```

Realiza un ataque de SYN Flood desde `kali2` a `kali1` al puerto 22 puedes utilizar la herramienta `hping3`[9]:

```
kali2:~# hping3 -S -p 22 --flood 10.100.(X+1).1
```

Interrumpe a los 3 segundos el comando `hping3` para que no haya demasiados paquetes, se está ejecutando una inundación.

[9]Si no está instalado el paquete hping3 en kali2, deberás instalarlo. En realidad este ataque se suele realizar enviando paquetes IP desde direcciones IP aleatorias, de tal forma que más difícil detectarlo. No lo vamos a hacer así, porque estaríamos enviando paquetes IP a otras máquinas aleatorias en la red

1. ¿Qué observas en el comando `netstat`?
2. ¿Qué observas en la captura de tráfico?
3. ¿Cómo relacionas el tamaño máximo de la cola de conexiones pendientes con lo observado en la captura?

Ahora vuelve a activar el mecanismo SYN cookies:

```
sysctl -w net.ipv4.tcp_syn_cookies=1
```

Vuelve a iniciar una nueva captura `fw-03.cap` en las mismas condiciones que antes y muestra nuevamente Realiza un ataque de SYN Flood desde `kali2` a `kali1` al puerto 22:

```
kali2:~# hping3 -S -p 22 --flood 10.100.(X+1).1
```

Interrumpe a los 3 segundos el comando `hping3` para que no haya demasiados paquetes, se está ejecutando una inundación.

1. ¿Qué observas en el comando `netstat`?
2. ¿Qué observas en la captura de tráfico?
3. ¿Por qué crees que esta configuración soporta mejor un ataque SYN flood?

4.3.2. iptables

Las siguientes reglas limitan el número de conexiones desde la misma dirección IP origen para que no haya más de 20 SYN en el último segundo:

```
-p tcp -m state --state NEW -m recent --set --name sattack
-p tcp -m state --state NEW -m recent --rcheck --name sattack --seconds 1 --hitcount 20 -j DROP
```

1. Crea una cadena de usuario nueva (diferente de la que ya tenías en el script) dentro del script que ya has realizado para los apartados anteriore, de forma que contenga estas reglas y que añada un mensaje al fichero de LOG antes de tirar un paquete que sobrepase esta limitación: 20 SYN en el último segundo. Estas reglas deberán ser aplicadas en la cadena INPUT en `kali1`. Incluye el script en la memoria.
2. Vuelve a ejecutar el ataque SYN FLOOD, déjalo ejecutando 3 segundos para que no genere demasiado tráfico. Explica qué es lo que ocurre observando el fichero de LOG.
3. Realiza un vídeo corto donde se muestre la conexión de las raspberrrys y expliques el sondeo FIN y el ataque SYN flood y la forma en la que iptables puede detectarlo y descartar paquetes. Sube este vídeo a youtube.

5. Normas de entrega

Deberás subir al `aulavirtual` un fichero `snort-nmap-iptables.tgz` que contenga los siguientes archivos:

- La memoria en formato pdf.
- Un archivo `nmap-fw-caps.tgz` que contenga los ficheros con las capturas de `nmap-01.cap` a `nmap-11.cap` y de `fw-01.cap` a `fw-3.cap`.
- Enlace al vídeo de youtube con tus explicaciones.