RETO 3: Implementación de un firewall de nivel 2

La tarea a llevar a cabo consiste en la implementación de un firewall de nivel 2 que se ejecuta en paralelo al “learning switch” en el controlador OpenFlow POX. Este firewall se configura con un conjunto de pares de MACs (lista de control de acceso, ACL) de forma que cuando se establece una conexión entre el controlador y el switch, la aplicación instala en la tabla OpenFlow las reglas necesarias para deshabilitar todas las comunicaciones entre cada par de MACs.

1 Topología de red

El firewall implementado debería ser independiente de la topología de red subyacente. Este firewall debería tomar como entrada la lista de pares de MACs e instalarla en los switches de la red.

2 Gestión de conflictos

POX permite la ejecución de múltiples aplicaciones de forma concurrente, por ejemplo, el “learning_switch” puede ejecutarse junto con el firewall, pero POX no gestiona de forma automática los conflictos entre reglas. Por lo tanto, el programador de la red ha de asegurarse de que no se instalan reglas que entran en conflicto entre sí, por ejemplo, que las dos aplicaciones traten de instalar una regla con la misma src/dst MAC y con el mismo nivel de prioridad pero con acciones diferentes. La forma más simple de evitar este problema es asignar diferentes niveles de prioridad a las diferentes aplicaciones.

3 Compresión del código

Para realizar esta tarea, es necesaria la utilización de los siguientes 2 ficheros:

1. firewall.py: una clase esqueleto que se actualizará con la lógica para instalar reglas de firewall.
2. firewall-policies.csv: una lista de pares de MACs (políticas) tomada como entrada por la aplicación de firewall.

El archivo firewall.py, que contiene el esqueleto del código, consiste en una clase de firewall con una función función *_handle_ConnectionUp*. También contiene una variable global, *policyFile*, la cual almacena el path al archivo firewall-policies.csv. Cada vez que se establece una conexión entre el controlador POX y el switch OpenFlow, se ejecuta la función *_handle_ConnectionUp*.

La tarea a llevar a cabo consiste en leer el archivo de política y actualizar la función *_handle_ConnectionUp*. La función debería instalar reglas en el switch OpenFlow que eliminan paquetes cada vez que se encuentra direcciones MAC origen/destino que se corresponden con los pares especificados en la lista de MACs.

4 Comprobación de funcionamiento

Ejecuta el controlador POX:

```
$ cd ~/pox
$ ./pox.py log.level --DEBUG forwarding.l2_learning
taller_sdn_ee24.reto3firewall.firewall
```

De esta forma se ejecuta el controlador con las dos aplicaciones: learning switch y firewall.

Ahora ejecuta mininet:

```
$ sudo mn --topo single,3 --mac --switch ovsk --controller remote
```

En mininet trata de enviar un ping al host h2 desde el host h1:

```
mininet> h1 ping -c1 h2
```

Si has configurado todo correctamente, el host h1 no debería ser capaz de enviar un ping al host h2.

Ahora trata de ejecutar un ping del host h1 al host h3:

```
mininet> h1 ping -c1 h3
```

El host h1 ha de ser capaz de enviar un ping al host h3, ya que no hay ninguna regla instalada en la red que deshabilite la comunicación entre ambos.

Por último, trata de ejecutar un ping del host h3 al host h2:

```
mininet> h3 ping -c1 h2
```

Para debugear en cualquier momento las reglas instaladas en el switch puedes ejecutar el comando:

```
sudo ovs-ofctl dump-flows s1
```