Assignment 2: Network sniffing

Alessandro Serpi - 1647244

8 March 2019

Exercise 1: Intercept DHCP messages

1.1 Plan of the activities

Start *tcpdump*, then the DHCP server. Lastly, close *tcpdump*.

1.2 Implementation

On *boundary*, the packet capture is started in background using `tcpdump -i eth1 -w dhcp.pcap -n port 67 and port 68 &`, then the DHCP server is launched with `udhcpd`.

When both *pc1* and *pc2* have a local IP address, *tcpdump* can be killed.

1.3 Testing procedure

There is no testing mechanism (the lab has not been altered).

1.4 Final remarks

The `dhcp.pcap` file is transferred to the host using the command `docker cp`.

Exercise 2: Mistakes in the IP addresses

2.1 Identify the errors

2.1.1 Get the IP addresses

Get the local IP address of every interface with `ip a s` ⟨`interface`⟩, where `interface` is `eth0` for *pcX* and `eth1` and `eth2` for *boundary*.

2.1.2 Check connectivity

From every host, ping every other host. The type of failure should provide information about the error in the configuration.

2.2 Correct the errors

2.2.1 pc2

pc2 did not ping *pc6* directly, but it routed the ping through *border*. The error was in the netmask in `eth0` address declaration (`/29` instead of `/28`).

2.2.2 pc3

pc3 could not ping hosts outside of its LAN. The default gateway address was incorrect (`192.168.10.18` instead of `192.168.10.18`).

2.2.3 pc6

pc6 could reach neither the default gateway nor some hosts in the LAN. The netmask in `eth0` address was incorrect (`/29` instead of /28).

2.2.4 pc10

When trying to ping *pc10* from any other host, there was a warning. The address assigned to `eth0` was reserved for broadcast messages (`192.168.20.71`, changed to `192.168.20.70`).

2.3 Testing procedure

In every host the local IP address is retrieved using `ip addr show`, then every host pings every other hosts and 1.1.1.1. The test is successful if and only if all pings are successful.

2.4 Final remarks

Despite what the plan of activity states, it is not useful to ping all hosts from every host. Normally, at most three pings are necessary.

Exercise 3: Intercept RIP messages

3.1 Plan of the activities

Start *tcpdump* on an interface in the `internal` network, then launch *quagga*, the RIP provider, in every router. Lastly, close *tcpdump*.

3.2 Implementation

On *border*, the packet capture is launched in background using `tcpdump -i eth1 -w rip.pcap -n port 520 &`, then *quagga* is restarted with `/etc/init.d/quagga restart`.

After 20 seconds, *tcpdump* can be killed.

3.3 Testing procedure

There is no testing mechanism (the lab has not been altered).

3.4 Final remarks

The `rip.pcap` file is transferred to the host using the command `docker cp`.