NET1 : TP2 – Scapy

Time : 2h

Authorized documents : All

Submission : file tp4_surname_name.txt

Prerequisites :

- Python 3+
- Linux distro (Installed by default on Kali)
- VirtualBox 7.x
- We recommand to use a virtual machine with a bridged network connection

Execution method : The workshop must be done in solo

Deadline : 1 week from the workshop date

NB :

- Correction is automatic, strictly follow the instructions !
- The answers of the underlined questions must not to be informed in the submitted file
- Use the school network, it doesn't work with mobile connection sharing

Install :

From source

https://github.com/secdev/scapy/releases

by *Pypi*

pip install scapy

- scapy – h -> show options
- exit() -> Exit from scapy

Partie 1 : Getting started

Check the headers of protocols with the "ls" command. « ls »

- Layer 2 : ls(Ether, verbose=True)
- Layer 3 : ls(IP, verbose=True), ls (ARP, verbose=True)
- Layer 4 : ls(TCP, verbose=True), ls (UDP, verbose=True)

The slash operator « / » is used to concatenate the layer protocols

1. PING

Command : *Ether()/IP(dest="127.0.0.1")/ICMP()*
Allows you to forge a ping request to the localhost ! The protocols suite of is so, Ethernet at layer 2, IP layer 3 and ICMP() at layer 4,5,6,7.

2. Packets sending

The commands *sr, srp, sr1, srp1* allow tu send forged packets.
sr : send at layer 2
srp : send at layer 3

So now, we try to ping the localhost

res=srp1(Ether()/IP(dst="127.0.0.1")/ICMP())

We can inspect the response as follow :

res[0].summary()
res=sr1(IP(dst="8.8.8.8")/)/ICMP())
res[0].summary()

Replace *srp1* by *srp* and *sr1* by *sr and* ping the google DNS server 1.1.1.1

3. Network packet capture

Scapy provides also a tool to capture and analyze sent and received packets, like ***sniff()***.
Here are some options :

a. ***count*** : defines the number of packets to capture
b. ***iface*** : Network interface
c. ***filter*** : for packet filtering
d. prn : executed function on each arrivig packet

Try the ping and use the following commands :

a. p= sniff(count=2, filter= 'icmp', iface='eth0') (remove "iface="eth0" for Windows OS)
b. p.show()
c. p[0].show()
d. p[1].show()
e. p= sniff(count=2, filter= 'icmp', iface='eth0', prn=lambda x:x.summary())

NB: The sniff must be running while you are doing the ping

Part 2 : Applications

AS TRACEROUTE

We will now try to discover gateway (router) address with the technique used by the traceroute application. A packet's TTL field is decremented by 1 every time it passes through a router. When the "ttl" expires, the last router crossed, sends an "icmp" packet to the source. You can therefore know the number of routers to pass through to reach the destination, if you send packets with a progressive "ttl", in order to force each crossed router to respond. Send multiple pings with a progressive "ttl" (1, 2, 3…).

a. Make a ping to « 8.8.8.8 » with a ttl=1
   Q1 : What type (protocol) of response do you get?
   why ?
   Q2 : What is the routeur address ?
   Q3 : What is your PC address ?
b. We will go further to find out how many routers will be necessary to cross to reach the Google DNS server (8.8.8.8). You will have to use the same technique used in the previous exercise. Make the ttl evolve each time.
   Q4 : How many routers is needed to cross to reach the destination
   Q5 : What are their IP addresses ?

4. ARP

You will forge an ARP request to obtain the mac address of your gateway.

ARP requests are broadcasted the applicant to the whole network with the goal to get a MAC address from an IP address. Only the machine with the IP address will respond. You will forge an ARP request in order to get back your router MAC address. The stack is ETHER()/ARP(). Use sniff to see and confirm the result.

Q6 : What command did you use?
Q7 : What is the MAC / IP association received?

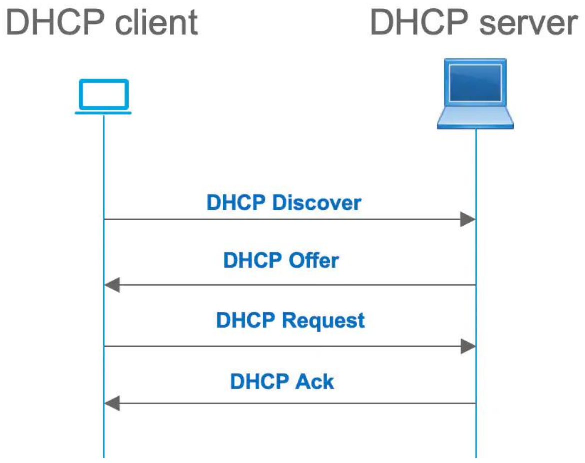
Tips: Use ls(ARP) or ARP().show() to see where and how to place the ttl parameter.

5. DHCP

DHCP is a client/server protocol that automatically distributes IP addresses and other associated configuration information to new connecting hosts. The DHCP exchange between a host and a DHCP server is carried out in 4 messages. The host's DHCP client broadcasts the **Discover message** to everyone in order to locate DHCP servers on the network. This message is special because the source address is: 0.0.0.0 and the IP address destination is 255.255.255.255. The DHCP server responds, if there is one, with an **Offer message** to propose an address to the client with some network parameters. The client uses the received parameters to set up its network card. Then, it responds to the server with a **request message**, with the aim to validate the receipt of its IP address. Once received the request, the server acknowledges with an **Ack message**. The ports used by DHCP are: client 67, server 68.

Fo further informations

- https://www.rfc-editor.org/rfc/rfc2131#section-1 (en particulier pages 10 "champs", 11 "flags")
- https://www.rfc-editor.org/rfc/rfc2132 (en particulier pages 5 "end" et 27 "m https://scapy.readthedocs.io/en/latest/api/scapy.layers.dhcp.html#scapy.layers.dhcp.DHCPOptionsFieldessages-types")
- https://github.com/secdev/scapy/blob/363d3766f53c3d55e92b0d51c5cdde7185733e3b/scapy/layers/dhcp.py#L208C14-L208C1



a. You have to forge and send a DHCP **discover** message to the DHCP server of your network. The stack is as follows*: ETHER / IP / UDP/BOOTP/DHCP* . You should remove the scapy pre-filter : conf.checkIPaddr=False

Q8 : What informations are contained in the BOOTP() function of the query? (field: value, field : value)
Q9 : What informations are contained in the DHCP() function of the query? (field: value, field : value)
Q10 : What informations are contained in the BOOTP() function of the response? (field: value, field : value)
Q11 : What informations are contained in the DHCP() function of the response? (field: value, field : value)

b. Then forward the DHCP request. Use sniff to confirm receipt of the server response. Can you explain the response options field?
Q12 : What informations are contained in the DHCP() function of the query? (field: value, field : value)
Q13 : What informations are contained in the DHCP() function of the response? (field: value, field : value)