TP 3 Adressage (Couche Internet et Couche Liaison)

Table des matières

Adressage (Couche Internet et Couche Liaison)

1. TP 4 Réseaux : ARP, ICMP et Fragmentation IP

Pour les étudiants souhaitant de tester ce TP à domicile, il suffit de connecter au réseau domestique et puis d'utiliser des adresses IP du réseaux locale (192.168.1.1--192.168.1.254).

Partie 1 (ARP) :

Ouvrez votre CMD et Wireshark. Ensuite, ouvrez l'interface réseau que vous utilisez (NpcapLoopback si vous utilisez localhost, sinon l'interface Wifi si vous êtes connecté au réseau local ou réseau de la faculté) sur Wireshark.

1. Connectez au réseau local et repérez les messages ARP diffusés dans le réseau (ajoutez un filtre *arp.opcode==2 || arp.opcode==1*). Pour cette étape vous devez capturer l'interface Wifi ou Ethernet (les message arp ne seront pas reçus par l'interface loopback);
2. Quels sont les types des messages ARP affichés après l'application du filtre ? Repérez votre adresse MAC dans les des message ARP?
3. Exécutez la commande *ping* sur votre CMD vers une machine distante (*ou vers le loopback 127.0.0.1*) et repérez les paquets ICMP Echo request / reply dans Wireshark. Pour repérer les messages ICMP echo request/reply, ajoutez le filtre *icmp.type == 0 || icmp.type==8* dans la capture de paquets Wireshark.
4. *Sélectionnez un paquet ICMP echo request/reply et repérez le payload (champ Data du paquet ICMP) générés par le ping dans le pacquet afficher par Wireshark.*
5. Exécutez la commande *tracert* sur votre CMD vers une machine distante (ou site web par exemple *google.dz*)(ajoutez le filtre *icmp.type == 0 || icmp.type==8 || icmp.type==11).* Quels sont les types de messages ICMP générés par la commende *tracert* ?
6. Repérez la copie du message expiré dans le message ICMP type 11 (*Time Exeeded*) .

Remarque : La commande traceroute de linux n'utilise pas ICMP Echo Request et ICMP Echo/Reply. Celle-ci génère des paquets vers la destination avec le même principe de TTL incrémenté que tracert. Ainsi, les sauts intermédiaires réponds avec icmp.type==11 et icmp.type==3 (ajoutez le filtre *icmp. type == 3 || icmp.type==11)*.

Partie 2 (Fragmentation IP et flags):

1. Sélectionnez l'interface Wifi et exécutez le programme python suivant. Celui-ci transmet un message UDP vers le port 50000 d'une machine locale (et cette dernière ne possède pas de programme écoutant ce port). Ainsi, le contenu ne peut pas être livré (*Destination Unreachable*).
2. Quel est le type du message ICMP et le code reçu à cause de cette transmission (capturez la transmission avec Wireshark) ?
3. Repérez la copie du message qui n'a pas pu être livrer au port 50000 dans le message ICMP.
4. Changez l'adresse IP destination dans le programme suivant en mettant *IP=" 172.16.160.2"* (ou une adresse de votre choix dans le réseau local par exemple 192.168.1.1 si vous êtes connectés au réseau local chez vous) et *MESSAGE ="ABC"*1500.*(*MESSAGE ="ABC"*1500* == une chaîne de caractères de 4500 (3 x 1500) où "ABC" se répète pour 1500 fois)
5. Utilisez le filtre *(ip.frag_offset!=0 || ip.flags.mf==1)* dans Wireshark pour repérer la transmission des fragments ;
6. *Exécutez le code python et repéré les flags allumés dans le header IP pour indiquer la fragmentation.*

```python
import socket
IP = "192.168.1.1"
PORT = 50000
MESSAGE = "Ey 192.168.1.1! tu ecoute le port 50000?"
print "UDP target IP:", IP
print "UDP target port:", PORT
print "message:", MESSAGE
#creation d'un socket UDP
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
sock.sendto(MESSAGE, (IP, PORT))

```