Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Réseaux Avancés II

Examen — mai 2021

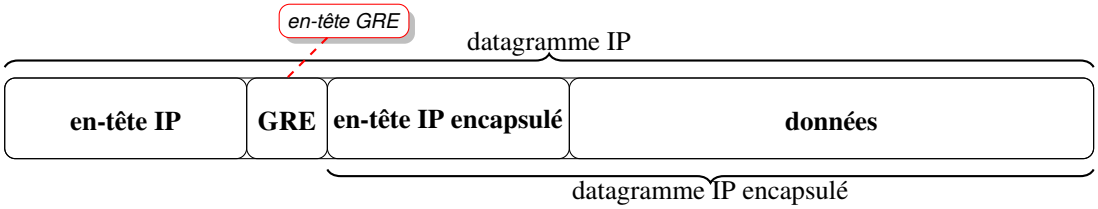
Durée : 1h30 — Tous documents autorisés

Programmation Python avec Scapy — 9 points

1 – Analyse automatique de contenu de paquet GRE, « *Generic Encapsulation* »

9pts Le protocole GRE est un protocole de « *tunneling* » de niveau 3 :

- ▷ son numéro de protocole est 47 ;
- ▷ il encapsule un datagramme IP dans un datagramme IP :



- ▷ le format de l'en-tête GRE est le suivant :

```
|0 1 2 3 4 5 6 7 8 9 a b c d e f|0 1 2 3 4 5 6 7 8 9 a b c d e f|
+-+-+-+-+-+-+-----+---------+-----+-------------------------------+
|C|R|K|S|s|Recur|  Flags  | Ver |          Type de protocole      |
+-+-+-+-+-+-+-----+---------+-----+-------------------------------+
```

- ▷ il est capable de transporter du trafic « *multicast* ».

a. Quelle est la taille d'une en-tête IP ? *(1pt)*
Quelle est la taille de l'en-tête GRE ?

b. Comment peut-on récupérer dans Scapy le contenu du datagramme IP encapsulé pour pouvoir l'analyser dans un objet `IP()` dans Scapy ? *(1pt)*

c. Si une adresse multicast est utilisée dans le datagramme IP encapsulé : *(1pt)*
Comment la reconnait-on ?
Où faut-il la chercher dans les champs du datagramme IP ?
Quel protocole est concerné ?
Est-ce qu'il y a des champs du protocole de transport qui ont des valeurs spécifiques par rapport à une adresse multicast donnée ?

d. Écrire un **programme Python** utilisant la bibliothèque Scapy pour : *(5pts)*
- ◇ intercepter le **trafic GRE** ;
- ◇ faire la **liste des adresses IP source et destination** des paquets IPs encapsulés ;
  Chaque adresse différente ne devra être affichée qu'une seule fois
- ◇ pour les paquets envoyés en **multicast** : indiquer le port destination et l'adresse multicast utilisée.

Pour l'analyse, vous afficherez le contenu du paquet à l'utilisateur de manière compréhensible.

e. Si le programme est installé sur un routeur, comment peut-il **recevoir** les paquets **directement depuis le firewall** ? *(1pt)*

xterm

```
>>> ls(IP)
version    : BitField (4 bits)                = (4)
ihl        : BitField (4 bits)                = (None)
tos        : XByteField                       = (0)
len        : ShortField                       = (None)
id         : ShortField                       = (1)
flags      : FlagsField (3 bits)              = (<Flag 0 ()>)
frag       : BitField (13 bits)               = (0)
ttl        : ByteField                        = (64)
proto      : ByteEnumField                    = (0)
chksum     : XShortField                      = (None)
src        : SourceIPField                    = (None)
dst        : DestIPField                      = (None)
options    : PacketListField                  = ([])
```

IPv6 — 11 points

2 – 5pts Analysez la trame suivante :

```
0000  96 95 68 6A B3 2E 9E A1 DD 88 13 B5 86 DD 60 00  ..hj..........`.
0010  00 00 00 17 11 01 FE 80 00 00 00 00 00 00 9C A1  ................
0020  DD FF FE 88 13 B5 20 01 08 D0 CD 0B 85 00 1E 83  ...... .........
0030  41 FF FE 28 95 0D D6 CA 00 16 00 17 58 4A 31 32  A..(........XJ12
0040  33 34 35 36 37 38 39 41 42 43 44 45 46           3456789ABCDEF
```

a. Que contient la trame ? *(1pt)*
Vous donnerez une description pertinente.

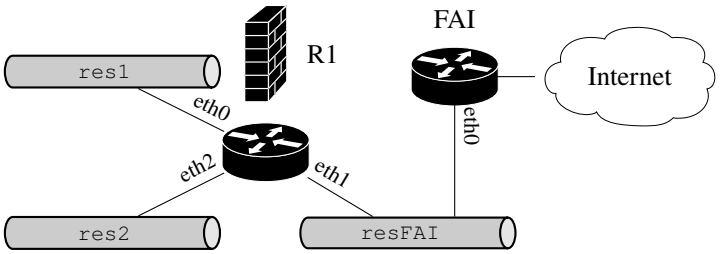
b. Est-elle passée par un routeur ? *(1pt)*

c. Est-ce que les adresses IPv6 ont été obtenues par « **auto-configuration** » ? *(1pt)*
Analysez ces adresses et justifiez votre réponse.

d. Peut-on apprendre des informations sur la nature du matériel en réception et/ou en envoi ? *(1pt)*

e. Est-ce un paquet « *forgé* » ? Pourquoi ? *(1pt)*

3 – 6pts Soit le réseau d'entreprise suivant :



a. Sur le routeur R1, on consulte la configuration du firewall : *(1pt)*

xterm

```
> sudo ip6tables -t nat -nvL
Chain PREROUTING (policy ACCEPT 2 packets, 267 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain INPUT (policy ACCEPT 2 packets, 267 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain OUTPUT (policy ACCEPT 2 packets, 267 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain POSTROUTING (policy ACCEPT 4 packets, 534 bytes)
 pkts bytes target     prot opt in     out     source               destination
 1526 95268 MASQUERADE  all      *     eth1    fde4:8dba:82e1::/64  ::/0
```

À qui correspond cette règle de firewall ?
Quel type de réseau est concerné ?

b. De quelle **type** doit être l'adresse `eth1` de R1 pour permettre l'**accès à Internet** ? *(1pt)*

c. Donnez : *(2pts)*
- ◇ un plan d'adressage pour les réseaux « *res1* » et « *res2* » utilisant la règle de firewall précédente.
- ◇ une adresse pour chacune des interfaces `eth0` et `eth2` du routeur R1.

d. Donnez la table de routage du routeur R1 permettant l'accès à Internet. *(2pts)*
Vous donnerez des adresses pour `eth1` de R1 et `eth0` du routeur FAI comme indiqué dans la question b).