IFT3820/IFT6833 – TRAVAIL PRATIQUE #2 – 17 juin 2002

« Sniffer » où l'art et la manière d'écouter aux portes !

Mohamed Lokbani

Date de remise: Lundi 8 juillet, 22h30 au plus tard, sans possibilité de retard.
Équipes: Le travail en équipe de deux est permis (même encouragé). Vous ne remettez alors qu'un travail par équipe.
Remise : Deux remises à effectuer : électronique et papier (le cours ou la démo du 08 juillet).
Conseil: N'attendez pas la semaine avant la remise pour commencer, vous n'aurez pas le temps.

Barème : Il est important de noter que vous devez utiliser les machines du laboratoire A.A. 1340 pour répondre aux exercices suivants. Chacune de vos réponses, doit être claire et accompagnée des explications nécessaires. Ce TP est noté sur **14 points**.

Remise : Vous devez faire **deux types** de remise :

- Électronique : **8 juillet 2002**, avant 22h30.
  Dans le répertoire où se trouve votre fichier à remettre, tapez la commande:

```
remise dift3820 tp2 vos_documents
```

- Remise Papier : le cours ou la démo du **08 juillet**.

Exercice -1- (un peu de Web ...) **3 pts**
En quelques lignes, expliquer comment la fragmentation fonctionne en IPv6, ses avantages et ses inconvénients, et en quoi diffère-t-elle de celle utilisée dans IPv4.

Exercice -2- (des agents un peu spéciaux ...) **5 pts**
Après avoir réussi à mettre la main sur un enregistrement contenant une écoute d'un échange de données entre deux machines, client et serveur, l'étape suivante consiste à rendre intelligible le contenu de cet échange. Comment faire cela?

Soit le fichier *ecoute.txt* contenant cet échange sous la forme de 104 trames Ethernet, dont nous avons retiré le préambule, le début et le FSC, où chaque trame est représentée sur une ligne. La première valeur à coté de chaque trame représente le numéro de la trame dans le fichier. Cette écoute a été réalisée avec la commande tcpdump, un "logiciel" "espion" (appelé dans le jargon technique « sniffer », traduction de « renifleur ») permettant d'écouter et d'analyser les trames circulant sur le réseau. Cette commande n'est accessible qu'à l'usager root, pour des raisons dont vous allez vous en rendre compte par vous même dans cet exercice.

Question -1- En utilisant uniquement la première trame, et tout en justifiant votre réponse, indiquer pour cette connexion TCP ce qui suit: Quelle est l'adresse physique, l'adresse IP et le port du client? Quelle est l'adresse physique, l'adresse IP et le port du serveur?

Question -2- Quelle est l'application (i.e. HTTP, SSH, FTP, TELNET etc.) qui tourne par dessus la connexion TCP et comment êtes-vous arrivés à cette conclusion?

Question -3- En vous basant sur le diagramme des échanges de segments TCP entre un client et un serveur, identifier (par le numéro associé à la trame) les paquets responsables du changement d'état de la transmission, coté client. Vous devez formuler votre réponse comme suit: (Exemple) Quand la machine

source reçoit la requête OPEN de l'utilisateur, l'état TCP change de CLOSED à SYN-SENT et le paquet p est envoyé.

Question -4- Refaire la même question, cette fois coté serveur.

Question -5- À partir des données fournies dans le fichier *ecoute.txt*, recréer le scénario complet (ce qui se produit dans la réalité i.e. loin des bits) des échanges entre le client et le serveur. On ne vous demande pas de donner une réponse exacte bit à bit de tout l'échange. Nous dire «en français» tout ce que le client a vu sur son terminal, et tout ce qu'il a tapé comme caractères sur son clavier.

Exercice -3- (faire, défaire et refaire ...) **6 pts**

Question -1- Écrire le programme *assemble.c* qui à partir des trames lues du fichier d'entrée, *fragments.txt*, regroupe les fragments associés au même paquet et les affiche en sortie dans le fichier *data.txt*, sur une seule ligne et cela pour chaque paquet (ici données fictives):

`00d0 b7b2 7517 00a0 c9fc 1821 0800 4510 0036` (etc. pour les fragments du paquet 1)
`4000 4006 5e15 84cc 1828 84cc 1aa2 85b3 0017` (etc. pour les fragments du paquet 2)
(Etc. pour les fragments du ième paquet)

Nous avons retiré le préambule, le début et le FSC des trames fournies dans le fichier *fragments.txt*

Question -2- Écrire le programme *fragmente.c* qui à partir des paquets lus du fichier, en entrée, *data.txt*, construit l'ensemble des trames à envoyer à travers le réseau.

Pareillement que dans la question -1-, ne pas inclure dans la trame le préambule, le début et le FSC. Supposer aussi que la taille des données dans une trame est de 1500.

Aide : Nous avons utilisé la commande ping (le protocole associé est ICMP) pour envoyer deux fois (3008+20) octets de données. Vu que le MTU est de 1500 (1480+20), la fragmentation devient donc obligatoire. Grâce à une écoute placée entre les deux machines, nous avons enregistré le trafic ICMP échangé entre ces deux machines.

Tous les échanges sont regroupés sous les deux formats suivants:
- *ethsortie.txt*: fichier contenant le résultat détaillé d'une écoute des échanges entre deux machines.
- *fragments.txt*: fichier contenant les trames obtenues suite à cette écoute.

Quelques exigences :
Vous devez :

- Écrire vos programmes en utilisant le langage C ou le langage C++.
- Respecter les noms attribués aux programmes (*assemble.c* et *fragmente.c*)
- Compiler sur une des machines du laboratoire (1340) et assurer vous que vous utilisez la bonne version du compilateur gcc :
  La commande: « gcc –v » doit produire le résultat suivant :
  Reading specs from /usr/lib/gcc-lib/i386-redhat-linux/egcs-2.91.66/specs
  gcc version egcs-2.91.66 19990314/Linux (egcs-1.1.2 release).
- Prendre en compte que si vos programmes ne compilent pas ou ne font pas toutes les choses demandées dans la spécification, la note sera de 0/6.
- Inclure des commentaires dans vos programmes et bien les (programmes) indenter.
- Remettre un document (3 pages maximum) expliquant la conception de vos programmes (ne faites pas un copié collé de vos commentaires).