Protocoles réseaux
TP n° 9 : Capture sous Wireshark

Wireshark permet d'analyser des traces, mais également d'en capturer. Ce tp vous donne quelques clés pour effectuer des captures, mais vous trouverez plus d'informations sur le sujet sur le wiki de Wireshark : `https://wiki.wireshark.org`.
Sous Wireshark, pour capturer des paquets, il faut au préalable choisir la ou les interfaces sur lesquelles faire les capture. Dans le menu, choisir `Capture`, puis `Options` et sélectionner une interface active (en général ici `en0`).

Ce TP ne donnera pas du tout les mêmes résultats sur votre machine personelle par wifi, car beaucoup de cibles sont sur le réseau local filaire...

Exercice 1 :

Démarrer alors une capture, puis la stopper après quelques secondes et essayer de comprendre les informations reçues.

Comme vous avez pu le constater, il est difficile de se repérer et de comprendre l'activité réseau car il y a plusieurs protocoles qui s'exécutent en parallèle. Wireshark permet d'appliquer deux types de filtre pour obtenir un affichage mieux ciblé. Il y a les filtres d'affichage et ceux de capture.
Les filtres d'affichages permettent de filtrer les paquets affichés dans la fenêtre Wireshark. On écrit le filtre dans la barre de filtre située sous la barre du menu. Voici quelques filtres d'affichage : On peut mettre plusieurs filtres à la suite séparés par des `and` ou des `or`. La négation s'exprime par le `not`.

- <*nom d'un protocole*> comme par exemple `ssh`, `http`, pour n'afficher que les paquets concernant le protocole ;
- <`tcp.port ==` *numero de port*>, pour afficher uniquement les paquets qui ont le port donné dans la source ou la destination ;
- <`http.host == "`*adresse http*`"`> ou <`http.host contains "`*partie d'adresse http*`"`>, pour afficher uniquement les paquets qui concernent une adresse web donnée partiellement ou totalement.

Exercice 2 :

Pour les applications ci-dessous, on vous demande de faire une capture en appliquant un filtre d'affichage de façon à faire disparaitre de l'affichage le maximum de paquets non concernés par l'application lancée.

1. Faire un `ssh` sur une machine de l'ufr.
2. Faire un `telnet` sur `monjetas` pour le service `daytime`.
3. Faire une requête `http` sur le site `www.informatique.univ-paris-diderot.fr`. Puis sur le site `www.google.fr`. Pourquoi cette dernière est-elle décevante ?
4. Quelles sont les requêtes ARP et DNS que vous voyez passer ? Interprétez.

Les filtres de capture permettent d'appliquer les filtres au moment de la capture, donc de réduire le nombre de paquets capturés. Pour appliquer un filtre de capture à une interface, il faut le faire avant de démarrer la capture. Pour cela, aller dans le menu `Capture`, puis `Options`. Sélectionner l'interface sur laquelle on souhaite appliquer un filtre, puis dans la barre `Filtre de capture`, écrire le filtre. Voici quelques filtres de capture :

- <`host` *adresse IP*>, pour capturer uniquement les paquets provenant ou à destination de l'adresse donnée ;
- <`tcp port` *numero de port*>, pour capturer uniquement les paquets TCP qui ont le port donné dans la source ou la destination.

Exercice 3 :

Reprendre l'exercice précédent en appliquant cette fois-ci des filtres de capture.

Exercice 4 : un petit `ping`

Lancer une capture Wireshark en filtrant pour n'avoir que les paquets ICMP, puis exécuter la commande `ping -c 5 192.168.70.10`

1. Combien de paquets ICMP circulent ?
2. Quelle est la taille des entêtes IP ?
3. Quelle est la taille des données ?
4. Dans l'entête IP, quelles sont les valeurs de «*Protocol*», «*Identification*» et «*Flags*» ?
5. Quels sont les types de messages ICMP échangés ?
6. Pour chaque paquet analyser le champ ICMP «*Identifier*» et «*Sequence number*».
7. Comment se fait la correspondance entre une requête et une réponse ICMP ?
8. Quels sont les champs constants et les champs qui sont modifiés dans les différents entêtes ?
9. Quel lien avec les champs `icmp_seq`, `ttl` et `time` de la sortie de `ping` sur le terminal ? Interpréter leurs différentes valeurs.

Exercice 5 : un gros `ping`

Éxecuter la commande
`ping -s 8000 www`

1. À l'aide de `man ping` déterminer ce que vous faites
2. Le MTU est de 1500 octets. Pourtant, le ping réussit. À l'aide de Wireshark, déterminez pourquoi.
3. Quelles sont les données du paquet ICMP ?
4. La fragmentation de couche 3 (couche réseau) peut être évitée à l'aide du bit « don't fragment » (DF) de l'entête IPv4. À l'aide de Wireshark, déterminez si TCP sur votre machine utilise le bit DF.
5. Pourquoi les implémentations modernes de TCP évitent-elles la fragmentation de couche 3 ? (Indication : que se passe-t-il si un fragment est perdu ?) Pourquoi ce problème ne se manifeste-t-il pas avec la segmentation de couche 4 ?
6. Et pour UDP : est il souhaitable/possible d'éviter la fragmentation ?
7. Pourquoi IPv6 n'implémente-t-il pas la fragmentation par les routeurs ?

Exercice 6 : une adresse bavarde

Éxecuter la commande
`ping -b 192.168.70.255`

1. Pourquoi autant de réponses ? Qui répond ?
2. Quelle est la nature de l'adresse IP `192.168.70.255` ?
3. Interpréter les différentes valeurs des champs `icmp_seq`, `ttl` et `time`