Alain KARAM

KARA12019801

Devoir 1

**Partie I: Ouvrir *Wireshark.***

A screenshot of a computer

Description automatically generated

Après avoir ouvert *Wireshark,* sélectionner l’interface réseau qui est connectée a votre réseau.

Grace a *NpCap* nous allons pouvoir faire une capture de paquets.A screenshot of a computer

Description automatically generated

**Partie II : Analyser les paquets.**

1. Que signifient les différentes couleurs dans le journal Wireshark ?

Nous remarquons que chaque paquet est surligner d’une couleur spécifique.

* Vert clair :
  + HTTP et FTP Data : Représente les paquets HTTP (port 80) et les transferts de données FTP.
* Bleu clair :
  + DNS : Représente les paquets DNS (port 53).
* Noir :
  + TCP Errors : Paquets avec des erreurs TCP, comme les réinitialisations (RST) ou les tentatives infructueuses de connexion.
* Jaune :
  + UDP : Paquets utilisant le protocole UDP.
* Rose :
  + TCP ACK : Paquets TCP avec des confirmations (ACK).
* Magenta :
  + ICMP : Paquets ICMP, souvent utilisés pour le ping.
* Vert foncé :
  + TCP SYN/FIN : Paquets TCP contenant des segments SYN ou FIN, qui sont utilisés pour établir ou terminer des connexions TCP.
* Gris :
  + TCP (générique) : Autres paquets TCP qui ne correspondent pas à une catégorie spécifique.

A screenshot of a computer

Description automatically generated

1. Pourquoi votre ordinateur reçoit-il des paquets qui sont adressés à une autre machine?

Même si un paquet n'est pas destiné à votre ordinateur, il peut passer par votre réseau. Cela se produit parce que votre ordinateur capture tout le trafic réseau en mode promiscuité, y compris les paquets destinés à d'autres machines.

1. Combien de paquets votre ordinateur envoie / reçoit-il en un seul clic de souris lorsque vous visitez un site Web ?
   1. Pendant la capture de paquet, ouvrez dans une fureteur le lien (utiliser au labo 1) ci-dessous :  
      <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>
   2. Après voir ouvert le site, arrêtez la capture de paquet.
   3. Ouvrez un terminal et tapez la commande ci-dessous :

*$ nslookup gaia.cs.umass.edu*

* 1. La commande *nslookup* permet d’obtenir les adresses IP du site.

A screenshot of a computer

Description automatically generated

* 1. Maintenant que nous savons l’adresse IP, retournez dans *Wireshark* et dans le champs de filtre entrez :

tcp.port == 80

* 1. Cela nous permet de filtrer les paquets enregistrés pendant la capture et de voir seulement les paquets dont le Protocol est TCP et utilise le port 80

A screenshot of a computer

Description automatically generated

* 1. Nous remarquons qu’en un seul clic de souris lorsque nous visitons ce site web spécifique, que 3 paquets sont échangés pour établir une connexion TCP :
     1. SYN -> (envoyé par le client)
     2. SYN-ACK -> (réponse du serveur)
     3. ACK - (confirmation du client)

1. Pourriez-vous organiser ou filtrer le trafic pour le rendre plus facile à comprendre ?

L’utilisation des filtres permet la faciliter et l’organisation du trafic de paquet. Tel qu’utiliser dans la question 3, on peut filtrer par Protocol ou par adresse IP.

1. Pourquoi votre ordinateur envoie-t-il autant de paquets ?

L'envoi de nombreux paquets plutôt qu'un seul très gros paquet s'explique par le modèle en couches des réseaux et les spécificités des protocoles de communication. Le réseau est divisé en plusieurs couches : application, transport, réseau, liaison, et physique. À mesure que l'information descend dans ces couches, elle est segmentée en unités de plus en plus petites, jusqu'à devenir des bits (1 et 0) pour être transmises sous forme de signaux physiques.

**Le Modèle en Couches**

* **Couche Application** : C'est la couche où les applications et les services réseau, comme les navigateurs web, fonctionnent. L'information est générée sous forme de données applicatives (ex. : un fichier HTML).
* **Couche Transport** : Ici, les données de la couche application sont segmentées en petits paquets. Selon le protocole utilisé (TCP ou UDP), ces paquets peuvent être traités différemment.
* **Couche Réseau** : Les paquets de la couche transport sont encapsulés dans des datagrammes IP, ajoutant des informations de routage nécessaires pour naviguer sur le réseau.
* **Couche Liaison** : Les datagrammes IP sont encapsulés dans des trames Ethernet, prêtes à être envoyées sur le réseau local.
* **Couche Physique** : Les trames Ethernet sont converties en signaux électriques ou optiques (bits 1 et 0) pour être transmises physiquement sur le réseau.

**Différences entre TCP et UDP**

* **TCP (Transmission Control Protocol)** :
  + **Fiabilité** : TCP assure que tous les paquets arrivent à destination dans le bon ordre. Il utilise des accusés de réception (ACK), des retransmissions et des numéros de séquence pour garantir la livraison des données. (tel que vu dans la question 3)
  + **Contrôle de Flux** : TCP ajuste dynamiquement la vitesse de transmission pour éviter la congestion du réseau.
  + **Utilisation** : Idéal pour les applications où la fiabilité est cruciale, comme le chargement de pages web, le transfert de fichiers (FTP), et les emails (SMTP).
* **UDP (User Datagram Protocol)** :
  + **Non Fiabilité** : UDP envoie les paquets sans garantie de livraison, d'ordre ou d'intégrité. Il n'y a pas de mécanisme de retransmission.
  + **Vitesse** : UDP est plus rapide que TCP car il n'a pas le surcoût du contrôle de flux et des accusés de réception.
  + **Utilisation** : Idéal pour les applications où la vitesse est plus importante que la fiabilité, comme le streaming vidéo, les jeux en ligne, et les communications vocales (VoIP).

1. Que signifient SYN, ACK, FIN et GET ?

SYN, ACK, FIN et GET sont des indicateurs utilisés principalement dans les protocoles TCP et HTTP pour gérer la communication entre les ordinateurs.

Voici ce que chacun de ces termes signifie :

* **SYN (Synchronize)**
  + **Protocole**: TCP (Transmission Control Protocol)
  + **Description**: SYN est un drapeau utilisé pour initier une connexion TCP entre un client et un serveur. Lorsqu'un client veut établir une connexion avec un serveur, il envoie un paquet SYN pour signaler le début de la communication.
  + **Rôle**: Il fait partie du processus de handshake à trois voies de TCP.
    - **Étape 1**: Le client envoie un paquet SYN au serveur pour demander une connexion.
    - **Étape 2**: Le serveur répond avec un paquet SYN-ACK pour accuser réception de la demande de connexion et synchroniser la connexion.
    - **Étape 3**: Le client envoie un paquet ACK pour confirmer la réception du SYN-ACK et établir la connexion.
* **ACK (Acknowledgment)**
  + **Protocole**: TCP
  + **Description**: ACK est un drapeau utilisé pour accuser réception des paquets reçus. Il est utilisé pour confirmer que les données ont été reçues correctement.
  + **Rôle**: ACK fait également partie du handshake à trois voies et est utilisé tout au long de la connexion TCP pour confirmer la réception des paquets de données.
    - **Après SYN-ACK**: Le client envoie un paquet ACK pour finaliser l'établissement de la connexion.
    - **Pendant la transmission des données**: Chaque paquet reçu est confirmé avec un paquet ACK pour assurer la fiabilité de la communication.
* **FIN (FINISH)**
  + **Protocole**: TCP
  + **Description**: FIN est un drapeau utilisé pour terminer une connexion TCP. Lorsqu'une des parties veut fermer la connexion, elle envoie un paquet FIN.
  + **Rôle**: Il signale la fin de l'envoi de données par l'expéditeur.
    - **Étape 1**: L'expéditeur envoie un paquet FIN pour indiquer qu'il a terminé d'envoyer des données.
    - **Étape 2**: Le récepteur envoie un paquet ACK pour accuser réception du FIN.
    - **Étape 3**: Le récepteur envoie ensuite un paquet FIN pour terminer la connexion de son côté.
    - **Étape 4**: L'expéditeur répond avec un paquet ACK pour confirmer la fin de la connexion.
* **GET**
  + **Protocole**: HTTP (HyperText Transfer Protocol)
  + **Description**: GET est une méthode de requête HTTP utilisée pour demander des données à un serveur. Elle demande au serveur de renvoyer des ressources spécifiques, telles que des pages web, des images ou des fichiers.
  + **Rôle**: GET est l'une des méthodes les plus courantes pour récupérer des informations sur le web.