TD/TP de Modèle OSI et TCP/IP

Créer un fichier texte pour y vos résultats

TP1 : Introduction à Wireshark pour l'Analyse de Réseau

Objectif :

Permettre aux étudiants de se familiariser avec Wireshark, un outil puissant pour l'analyse et le diagnostic des réseaux. À la fin de ce TP, les étudiants seront capables de capturer et d'analyser des paquets réseau, d'identifier différents protocoles et de comprendre les échanges de données sur un réseau.

Matériel requis :

- Un ordinateur avec Wireshark installé (disponible sur [le site officiel de Wireshark](https://www.wireshark.org/))

- Accès à un réseau local (LAN) ou à une connexion internet)

(chacun doit procéder à l'installation de Wireshark sur son SE : Windows/Linux)

Durée :

2 heures

1. Introduction à Wireshark (15 minutes) et principe de fonctionnement

Wireshark capture des trames de la couche liaison (Ethernet) d'un ordinateur, ce qui permettra de capturer tous les messages envoyés (ou reçus) par tous les protocoles exécutés sur cet ordinateur. Ceci car la communication se fait selon le principe d'encapsulation, et donc tous les protocoles des couches supérieures sont finalement encapsulés dans une trame Ethernet. Grâce à l'analyseur de paquets, qui comprend la structure des messages échangés par tous les protocoles, Wireshark peut afficher le contenu de chaque champ d'un message dépendant du protocole qu'il l'a échangé.

Exemple. Supposons qu'on a utilisé le navigateur pour consulter le site web du centre universitaire de de Nazi boni .

Donc, au niveau « Application » on a utilisé le protocole HTTP. Le message HTTP est encapsulé dans des messages TCP ou UDP, qui sont à leurs tour encapsulés dans des paquets IP, encapsulés par la suite dans des trames Ethernet, et enfin transmis sur le support physique.

Wireshark capture une trame Ethernet. Comme il sait bien le format de clette trame, il peut identifier le datagramme IP encapsulé dedans. Aussi, il connait également le format du datagramme IP, et ainsi il peut extraire le segment TCP. Enfin, il connait la structure du segment TCP, et donc il peut extraire le message HTTP qu'il contient. Enfin, il analyse le message HTTP et l'affiche selon la structure de la donnée http

Comprendre l'interface et les fonctionnalités de base de Wireshark.

Étapes :

1. Ouvrir Wireshark.

2. Présentation de l'interface :

https://www.it-connect.fr/decouverte-de-linterface-de-wireshark/

- Menu principal
- Barre d'outils
- Fenêtre de liste de paquets
- Fenêtre de détails des paquets
- Fenêtre des bytes des paquets

Activité :

- Familiarisez-vous avec l'interface en explorant les menus et les différentes sections.

2. Capture de paquets réseau (20 minutes)

Objectif :

Apprendre à capturer des paquets réseau sur une interface spécifique.

Étapes :

1. Sélectionner une interface réseau active (Wi-Fi ou Ethernet).

2. Démarrer la capture de paquets :

- Cliquez sur l'icône "Start Capturing Packets" (représentée par un requin bleu).

3. Laissez la capture tourner quelques minutes.

4. Arrêter la capture en cliquant sur l'icône "Stop Capturing Packets" (représentée par un carré rouge).

5 Capture et analyse de trafic généré par la commande ping

Pour cet exercice, il est demandé de noter son adresse IPv4 ainsi que celle de sa passerelle.

On va pour cela utiliser la commande ipconfig dans la console de commande de Windows (voir

https://fr.wikipedia.org/wiki/Ipconfig)

faites **ipconfig** pour noter

- Adresse IP de votre ordinateur : _____
- Adresse IP de votre passerelle : _____

La commande ping utilise le protocole ICMP

(https://fr.wikipedia.org/wiki/Internet_Control_Message_Protocol)

On va maintenant observer le trafic réseau généré par la commande ping :

- Lancer une nouvelle capture avec Wireshark (on acceptera de ne pas sauvegarder la précédente)
- Dans la console de commande de Windows, exécuter la commande ping x.x.x.x

(remplacer x.x.x.x par l'@IPv4 de la passerelle)

- Une fois les 4 réponses obtenues, stopper la capture.
- Appliquer le filtre icmp pour n'afficher que les trames générées par la commande ping.

On peut maintenant analyser le trafic capturé :

- Nombre de trames générées : _____
- @IP source d'une trame request : _____
- @IP destination d'une trame request : _____
- @IP source d'une trame reply : _____
- @IP destination d'une trame reply : _____
- Durée écoulée entre la trame request et la trame reply du premier échange : _____
- Nombre d'octets de données de la trame request : _____
- Nombre d'octets de données de la trame reply : _____

Activité :

- Observez les paquets capturés dans la fenêtre de liste de paquets.

Filtres de capture :

- Utilisez les filtres pour capturer des types de trafic spécifiques :
  - `tcp` : pour capturer uniquement le trafic TCP.
  - `udp` : pour capturer uniquement le trafic UDP.

- `http` : pour capturer uniquement le trafic HTTP.

3. Analyse de paquets (30 minutes)

Objectif :

Apprendre à utiliser les filtres d'affichage et à analyser des paquets spécifiques.

On peut remarquer qu'il y a beaucoup de types de paquets capturés et affichés par Wireshark, à cause du grand nombre de protocoles supportés par ce logiciel. Pour pouvoir chercher à l'intérieur des données capturées et n'afficher que le trafic qui nous intéresse, on utilise des filtres.

La syntaxe générale d'une expression simple d'un filtre est :

Protocole	•	Paramètre	Opérateur	Valeur

Exemple : ip . addr == 192.168.1.1

Six opérateurs de comparaison sont disponibles :

Format anglais:	Format de type C:	Signification:
eq	==	Equal
ne	!=	Non équal (Not Equal)
gt	>	Plus grand que (Greater than)
lt	<	Plus petit que (Less than)
ge	>=	Plus grand ou égale à (Greater or equal)
le	<=	Plus petit ou égal à (Less or equal)

Quatre opérateurs logiques sont disponibles :

Format anglais:	Format de type C:	Signification:
and	&&	Logical AND (et)
or	||	Logical OR (ou)
xor	^^	Logical XOR (ou)
not	!	Logical NOT (non)

ip.addr == 192.168.1.1 → Affiche les paquets avec une adresse source ou destination de 192.168.1.1

tcp || ip || dns → Affiche les trafics TCP ou IP ou DNS.

ip.src == 192.168.1.1 && ip.dst ! = 172.16.10.2 → Affiche les paquets avec une adresse IP source égale à

192.168.1.1 et de destination différente de 172.16.10.2

eth.addr == 00:2F:4C:01:23:6C → affiche le trafic de la machine dont l'@ MAC est 00:2F:4C:01:23:6C

Remarque. Ces filtres sont des filtres d'affichage, et il y a aussi les filtres de capture qui déterminent quels types de paquets à capturer.

Étapes :

1. Utiliser les filtres d'affichage :

- Entrez des filtres dans la barre de filtre (ex. : `http`, `dns`) pour isoler les paquets d'intérêt.

2. Sélectionner un paquet pour voir ses détails dans la fenêtre du milieu.

3. Explorer les en-têtes de paquets pour comprendre les échanges de données.

Activité :

- Essayez différents filtres d'affichage et examinez les paquets correspondants.

4. Études de cas (40 minutes)

Étude de cas 1 : Analyse d'une requête HTTP

Objectif :

Comprendre le fonctionnement des requêtes et réponses HTTP.

Étapes :

1. Ouvrir un navigateur web et aller sur un site web (ex. : `http://w3.uqo.ca/iglewski/ens/inf1493/src/html2/html2_http.php`).

2. Capturer le trafic réseau pendant cette action.

3. Utiliser le filtre `http` pour isoler les paquets HTTP.

4. Sélectionner une requête HTTP et analyser les détails (ex. : méthode, URL).

5. Sélectionner une réponse HTTP et analyser les détails (ex. : code de statut, en-têtes).

Activité :

- Identifiez et analysez une requête et une réponse HTTP dans le trafic capturé.

Étude de cas 2 : Résolution de noms de domaine avec DNS

Objectif :

Comprendre le fonctionnement des requêtes et réponses DNS.

Étapes :

1. Ouvrir un navigateur web et aller sur un site web (ex. : `http://example.com`**un autre site ici** avec une connexion).

2. Capturer le trafic réseau pendant cette action.

3. Utiliser le filtre `dns` pour isoler les paquets DNS.

4. Sélectionner une requête DNS et analyser les détails (ex. : nom de domaine demandé).

5. Sélectionner une réponse DNS et analyser les détails (ex. : adresse IP retournée).

Activité :

- Identifiez et analysez une requête et une réponse DNS dans le trafic capturé.

Étude de cas 3 : Analyse d'une connexion TCP

Objectif :

Comprendre le processus d'établissement et de terminaison des connexions TCP.

Étapes :

1. Démarrer une capture de paquets.

2. Ouvrir un navigateur web et aller sur un site web (ex. : `http://example.com`).

3. Utiliser le filtre `tcp` pour isoler les paquets TCP.

4. Identifier les paquets SYN, SYN-ACK et ACK pour le handshake TCP.

5. Identifier les paquets FIN, FIN-ACK et ACK pour la terminaison de la connexion TCP.

Activité :

- Identifiez les étapes du handshake et de la terminaison TCP dans le trafic capturé.

-iedentifier le mot de passe qui est clair.

5. Détection d'anomalies (15 minutes)

Objectif :

Apprendre à identifier des paquets suspects ou des comportements anormaux sur le réseau.

Étapes :

1. Discuter des signes courants de problèmes réseau (ex. : délais, pertes de paquets).

2. Utiliser les filtres pour identifier des paquets anormaux ou suspects (ex. : paquets malformés, tentatives de scan de ports).

3. Analyser un exemple de trafic anormal (ex. : paquets de scan de ports, tentatives de connexion répétées).

Activité :

- Utilisez des filtres pour détecter des anomalies dans une capture de trafic.

6. Conclusion et questions (15 minutes)

Activité :

1. Résumer les objectifs du TP et les compétences acquises.

2. Répondre aux questions des étudiants et clarifier tout point restant.

Exercices supplémentaires

1. **Analyse du protocole ARP :**
   - Capturer le trafic ARP et analyser les requêtes et réponses ARP.
   - Utiliser le filtre `arp`.

2. **Analyse du protocole ICMP :**
   - Utiliser la commande `ping` pour générer du trafic ICMP.
   - Capturer et analyser les paquets ICMP.
   - Utiliser le filtre `icmp`.

3. **Exercice de filtrage avancé :**
   - Utiliser des filtres combinés pour capturer des scénarios spécifiques (ex. : `tcp && ip.addr == 192.168.1.1`).

Conseils pratiques

- Encouragez les étudiants à explorer par eux-mêmes et à poser des questions.

- Fournissez des exemples de filtres courants et expliquez leur utilisation.

- Proposez des exercices supplémentaires pour les étudiants qui souhaitent approfondir leurs connaissances.

Ce TP permettra aux étudiants de se familiariser avec les fonctionnalités de base de Wireshark et de comprendre l'importance de l'analyse réseau .

Pour aller loin

Ressources :

https://www.youtube.com/watch?v=LkTuYaWZsrs

https://www.youtube.com/watch?v=XEAzadqtVoE

La documentation est disponible sur : https://sip.goffinet.org/wireshark/introduction-wireshark/