Laboratoire 2

1 Durée prévue : 2h00

2 Objectifs

Au terme de ce laboratoire, l'étudiant sera capable de :

- → Comprendre l'usage de base de l'outils Wireshark
- → Dessiner un diagramme en flèches pour illustrer le fonctionnement d'un protocole
- → Utiliser des règles de filtrage simples dans Wireshark
- → Constater les risques de transmission d'informations sensibles sur un protocole non-crypté

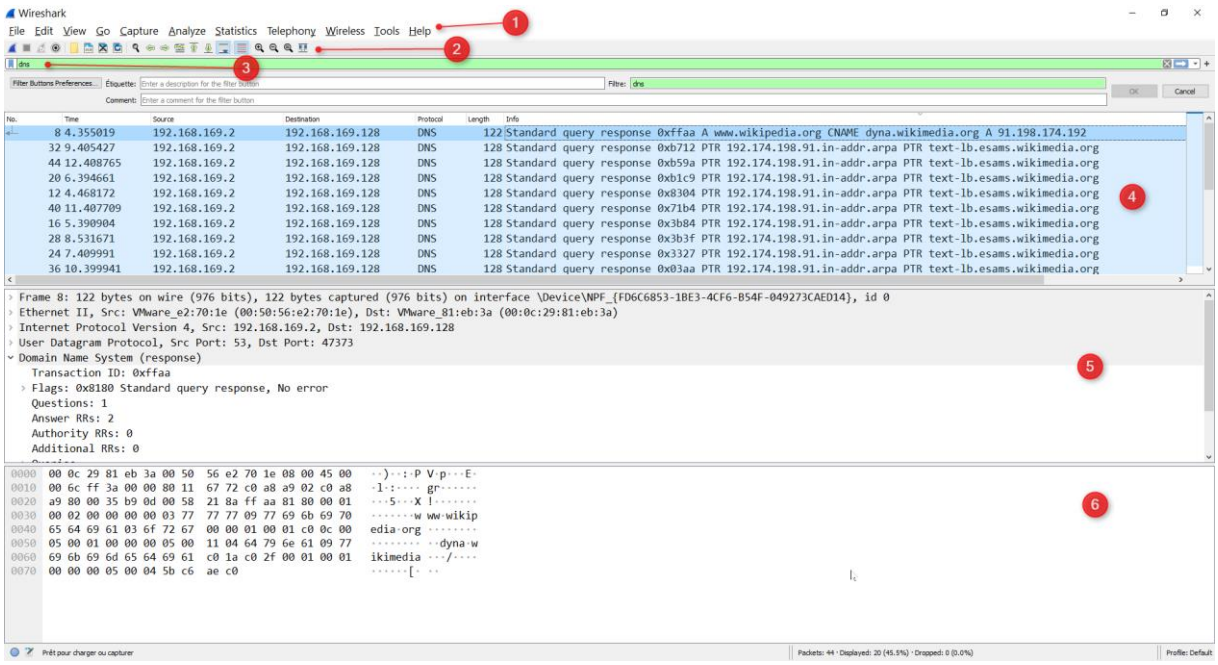
3 Introduction

L'analyseur de protocoles (ou sniffer) est un outil essentiel pour comprendre le fonctionnement des protocoles réseaux. L'analyseur Wireshark est un logiciel libre incontournable en la matière. Mais, il est nécessaire de prendre le temps de l'apprivoiser. Dans ce laboratoire, vous allez découvrir un usage simple de l'outil. Wireshark permet la capture de données envoyées sur un réseau, mais surtout d'afficher ces données de façon structurée, permettant une compréhension du fonctionnement des protocoles réseaux.

Pour ce laboratoire, vous devrez installer la dernière version de Wireshark disponible en téléchargement sur https://www.wireshark.org/download.html.
De nombreuses distributions linux incluent Wireshark dans leur gestionnaire de paquets. Ainsi sous CentOS 7 on tapera simplement `sudo yum install wireshark`. Au besoin, de nombreux tutoriels sont disponibles sur internet pour chaque distribution Linux.

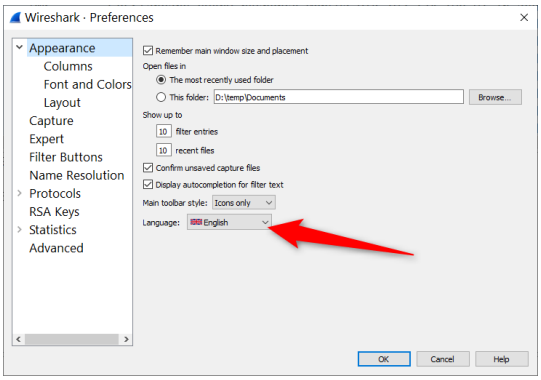
4 Présentation de l'interface

L'interface de l'analyseur se décompose en plusieurs barres, zones ou fenêtres.



1. La **barre de menus** regroupe la liste d'items classiques d'une application.
   Remarquez dans le menu File vous pouvez sauvegarder ou charger un fichier de capture réseau. Une capture peut très bien avoir été réalisée sur une sonde distante ou avec un autre outil et être analysée avec Wireshark à postériori. Les captures de trames que vous devrez d'ailleurs analyser seront fournies lors des premiers TP.

   Changez la langue de l'interface de l'outil via le menu Edit > Preferences….



2. La **barre d'accès rapide** regroupe tous les raccourcis sur les manipulations d'une capture les plus courantes.

3. La **barre de filtrage** sert à saisir l'expression de filtrage à postériori d'une capture pour isoler tout ou partie d'un échange réseaux. Ceci sera très utilisé pour permettre une analyse fine.

4. La **zone contenant la liste des trames capturées**
   Sur chaque ligne, on retrouve :
   a. le numéro de packet ;
   b. le temps de capture ;
   c. sa source ;
   d. sa destination ;
   e. le protocole de plus haut niveau décodé ;
   f. le résumé des champs caractéristiques du protocole.

5. La **zone d'affichage de la pile des protocoles décodés** pour la trame sélectionnée.
   Cette zone donne la liste de la pile de protocoles décodés allant du niveau physique (en haut) jusqu'au niveau le plus haut reconnu (en bas) du modèle en couche. Le protocole de niveau le plus haut reconnu est celui qui apparaît dans la colonne protocole de la zone contenant la liste des trames capturées (voir point 4.e ci-dessus), dans notre exemple, il s'agit du protocole DNS.
   a. La première ligne ou niveau Frame correspond à une pseudo couche physique. Comme il n'est pas possible de réaliser la capture directement à partir des composants électroniques qui pilotent l'interface réseau sans perturber le fonctionnement du système, l'opération a lieu au niveau liaison à l'aide de la bibliothèque libpcap.
   b. La deuxième ligne correspond au niveau liaison. On y détaille le type, les champs de la trame et les adresses physiques.
   c. La troisième ligne correspond au niveau réseau. On y détaille les champs du protocole réseau reconnu : adresses logiques et indicateurs d'état.
   d. La quatrième ligne correspond au niveau transport. On y détaille les champs du protocole de transport reconnu : état de la connexion, numéros de ports utilisés et diverses options.
   e. La cinquième ligne correspond au niveau application. On y trouve les données utilisateur.

   Pour détailler les données d'un niveau, il suffit de cliquer sur le triangle situé à gauche au niveau de chaque couche. Dans notre exemple, le niveau application, DNS, est développé.

6. La **zone d'affichage brut de la trame sélectionnée**, affiche tous les octets de la trame en hexadécimal.

5 Exercice 1 – Diagramme de flux de données

AVERTISSEMENT

L'**espace de travail** choisi pour les exemples est le répertoire `C:\Network`.

Si vous travaillez sur un ordinateur du Campus, vous n'avez pas les droits permettant de créer ce répertoire. À la place, créez un répertoire `Network` sur le disque réseau `H:` qui apparaît dans l'*Explorateur Windows*. Cet emplacement mémoire de 1 Go représente votre espace personnel sur le serveur. Il est accessible de n'importe quel ordinateur du campus.

Pour vous rendre dans cet emplacement via la console/le terminal, vous devez simplement saisir la commande `H:`.

5.1 L'objectif est atteint si vous savez :

- Utiliser un filtre d'affichage
- Interpréter les données du protocole
- Réaliser un diagramme en flèche
- Rédiger des conclusions d'activités constatées

5.2 Exercice 2 :

1. Créez dans votre espace de travail un répertoire nommé `labo2` (en minuscules et sans espace), ce qui donne `C:\Network\labo2` (ou `H:\Network\labo2`). C'est dans ce répertoire que vous enregistrerez tout le travail de ce 2e labo. Téléchargez le fichier **`T2-telnet-ssh.pcap`** disponible sur Learn

   https://learn-technique.helmo.be/mod/resource/view.php?id=231246 dans votre dossier.

2. Exécutez le programme Wireshark.

3. Dans ce laboratoire, nous ne réaliserons pas de capture à proprement parlé. Nous allons nous baser sur une capture enregistrée et vous allez devoir rédiger un rapport d'analyse de ces flux réseaux.
   Le fichier de captures réseaux à charger dans wireshark est **`T2-telnet-ssh.pcap`** (via le menu File > Open…)

4. D'après les données relevées dans wireshark, rédigez un rapport précisant les informations suivantes :

Rapport rédigé par : …….
Date de rédaction : ……/……/……
Date et heure de début de la capture : ……/……/…… à …h….m….s,……..
Date et heure de fin de la capture : ……/……/…… à …h….m….s,……..
Nombre de trames capturées : ……

5. Dans wireshark, dans la **zone contenant la liste des trames capturées**, vous constatez que les lignes sont de différentes couleurs. Chaque couleur désigne un protocole. Listez les différents protocoles des trames capturées et reportez l'information dans votre rapport (indice, il y en a 9).

Protocoles capturés : …….

6. Dans wireshark, appliquez un filtre d'affichage sur le protocole « **icmp** », analysez les paquets capturés.

7. Dessinez un diagramme en flèches comme ci-dessous et complétez avec les adresses ip relevées dans wireshark.
   Exportez votre diagramme en format de fichier image **jpg** ou **png** (**`diagramme-icmp.jpg`** ou **`diagramme-icmp.png`**) dans votre espace de travail **`\Network\labo2`**.

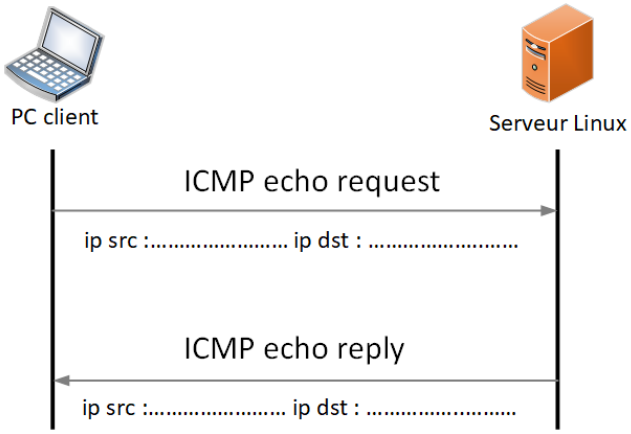


Diagramme en flèches à reproduire et compléter

Les diagrammes en flèches montrent de manière claire les paquets échangés entre différents nœuds. Ils permettent ainsi d'illustrer le fonctionnement d'un protocole.
Dans cet exercice, vous devez dessiner un diagramme en flèches selon une capture Wireshark.

Vous pouvez utiliser les logiciels suivants pour dessiner votre diagramme :

- Microsoft Visio (Windows)
- Omnigraffle (Mac)
- Microsoft Powerpoint
- OpenOffice/LibreOffice Drawing
- Dia (open source, multi-plateforme, http://projects.gnome.org/dia/)

8. D'après les données relevées dans wireshark, dans votre rapport précisez les activités réalisées sur le PC client, en précisant la date et l'heure de cette activité. Si nécessaire, utilisez une recherche sur internet sur le protocole ICMP pour cibler votre réponse.

Exemple : « En date du ……, à …h….m….s, nous pouvons déduire que l'utilisateur a exécuté la commande …… à partir de la machine à l'adresse IP …….. ».

9. Dans wireshark, appliquez un filtre d'affichage sur le protocole « **telnet** » et analysez les paquets capturés.

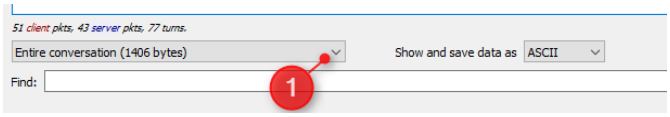
Le protocole Telnet permet l'interfaçage d'un terminal virtuel avec une application à travers Internet. Ce protocole fournit donc les bases permettant de relier un client telnet (système composé d'un affichage et d'un clavier) à un interpréteur de commande ou une application (côté serveur).

10. Recherchez le nombre de trames correspondant aux protocole telnet. Et renseignez dans votre rapport, l'heure de début de la session telnet et l'heure de la dernière trame. Bonus, pouvez-vous identifier le port TCP (couche session) du protocole telnet du serveur ?

11. Vous allez maintenant découvrir toute la puissance de l'analyseur de protocoles réseaux. Allez dans le menu Analyze > Follow > Stream TCP

12. A partir de ces éléments, complétez votre rapport en précisant les données critiques que vous avez pu relever.

13. Attention, l'affichage par défaut est bidirectionnel. Si l'application côté serveur fait un écho de ce qu'elle reçoi en entrée, certains caractères seront affichés en double (une fois en entré et une fois en sortie). Pour éviter cela, sélectionnez le flux souhaité :



14. Dans wireshark, appliquez un filtre d'affichage sur le protocole « **ssh** » et analysez les paquets capturés.

Le protocole SSH permet à un client (un utilisateur ou bien même une machine) d'ouvrir une session interactive sur une machine distante (serveur) afin d'envoyer des commandes ou des fichiers de manière sécurisée.

15. Recherchez le nombre de trames correspondant aux protocole ssh. Et renseignez dans votre rapport, l'heure de début de la session et l'heure de la dernière trame capturée. Bonus, pouvez-vous identifier le port TCP (couche session) du protocole ssh sur le serveur ?

16. Faite de même que pour les données du telnet, allez dans le menu Analyze > Follow > Stream TCP

17. Quel constat pouvez-vous faire ? Complétez votre rapport avec vos découvertes, en précisant vos recommandations quant à l'usage du telnet et du ssh pour l'administration distante du serveur linux.

18. Envoyez votre rapport sur Learn
https://learn-technique.helmo.be/mod/assign/view.php?id=231245