ESIEA MS-SIS

Rapport de Projet
Lecteur de fichier Pcap

Auteur :
Vincent CLÉMENT

6 novembre 2016

Table des matières

Chapitre 1

Présentation du projet

1.1 Introduction

.Dans le cadre de la formation mastère spécialisé de l'esiea, il nous était demandé d'écrire un programme capable de lire des fichiers pcap. Le programme doit respecter un cahier des charges qui implique d'utiliser le langage de programmation Java qui est orienté objet. Il ne doit pas contenir d'interface graphique, simplement une interface en console. L'utilisation d'une API, autre que l'API standard Java est interdite. Les différents protocoles devront être detecté par le programme, une option donné au programme permet de choisir le protocole que l'on souhaite afficher.

— Ethernet ;
— ARP ;
— IP ;
— TCP ;
— UDP ;
— ICMP ;
— DHCP ;
— HTTP ;
— DNS.

Dans ce projet, je n'ai pas implémenter tout les protocoles. Il me manque les protocoles de couche 5 à 7 (DHCP, HTTP, DNS)
Pour utiliser le programme, il faut donner en argument le fichier pcap a analyser (pas encore possible de choisir le protocole).

1.2 Présentation du fichier Pcap

La première chose a effectué lors de l'ouverture du fichier pcap, est l'extraction du "Global Header" du fichier. Il contient plusieurs informations essentielles à la compréhension du fichier, notamment l'encodage des informations binaire. Si Les informations sont codés selon la norme "Big-Endian" dans ce cas, on accepte d'ouvrir le fichier, sinon on le rejette. Pour la norme Little-Endian il faut inverser chaque octet.

La figure ci-dessous montre comment est composé le fichier pcap.

Dans ce fichier, chaque paquet est composé de deux choses :

— Un Header Pcap ;
— Un Packet Data.

Le header donne les informations de base du paquet. Les données correspondent au paquet réseau. Pour commencer, j'ai créé un objet pour rassembler les informations contenu dans le pcap header, et un second objet qui correspond au donnée. Ensuite dans chaque objet, il existe un attribut qui a pour type un tableau d'octets et qui contient les informations du fichier pcap. Ensuite une methode permet d'extraire les octets qui nous intéresse pour les analyser. Par exemple, on peut utliser cette méthode pour extraire les 4 premiers octets et vérifier que ces octets sont égaux à "d4c3b2a1".

1.3 Les différents objets du projet

Les réseaux sont construit autour du modèle OSI, lui-même divisé en 7 couches. Pour ce projet, j'ai créé un objet par couche. Les informations contenu dans l'objet "Header Pcap" contiennent les données sur la première couche de ce modèle. Sur chaque couche, j'ai créé plusieurs méthodes permettant de récuperer les informations intéressante. Puis ensuite, j'ai créé plusieurs objets par protocoles. Certains protocoles comme TCP et UDP héritent de la classe "Couche4".

Le premier objet est le pcapheader. Il contient :
— Le temps en sec ;
— Le temps original du paquet ;
— La taille du paquet ;
— La teille effective du paquet ;
— Les donnée du paquet.

Ensuite on envoie les donnée du paquet a la couche 2 du modèle OSI. Une méthode de l'objet determine les informations du paquet, si un protocole est detecté, on crée un objet pour ce protocole et on envoie les inforamtions du paquet à l'objet. Sur chaque couche du modèle OSI, on a une encapsulation des paquets. Par exemple les informations de la couche 3 sont contenus dans la charge utile de la couche2. Sur chaque analyse de paquet, lorsque l'on détecte une couche en plus, on crée un objet correspondant à cette couche.

On descend au fur et à mesure dans les couche. Si à un moment, on trouve un protocole, dans ce cas on arrête de monter dans les couches et on affcihe les dernières informations du paquet.

Pour chaque paquet, on affiche les informations couche par couche, en terminant par le protocole. Chaque objet a sa méhode "Informations" qui permet d'afficher les informations pertinante de la couche en cours d'analyse.

1.3.1 couche 2

La couche 2 est constituée :
— Une adresse MAC Source ;
— Une adresse MAC Destination ;
— Un éventuel protocole ;
— Une charge utile.

1.3.1.1 ARP

L'objet ARP étend la couche2 est constitué :
— du hardware type ;
— du prototype type ;
— du type de requête (Request ou Reply) ;
— de l'IP Source ;
— de l'IP Destination.

Si les informations du header de la couche 2 n'indique pas que le protocole suivant est ARP ou IP. On arrête l'analyse.

1.3.2 couche 3

La couche 3 est constituée de toute les informations concernant IP :
— Version ;
— taille ;
— ttl ;
— protocole suivant ;
— IP Source ;
— IP Destination ;
— une charge utile.

Si le protocole de couche 3 est différents d'IPv4, on abandonne l'analyse. Si on détecte ICMP, TCP, UDP, on continue l'analyse et on créé l'objet correspondant.

1.3.3 couche 4

La couche 4 est constituée :
— Port Source ;
— Port Destination ;
— protocole ;

— charge utile.

En fonction du protocole, on créé les objets suivants :

1.3.3.1 ICMP

L'objet ICMP est constitué :

— type ;
— code ;
— checksum ;

En fonction du code, on determine la requete ICMP qui est effectué (Request ou reply ou ttl exceed, ou destination unreacheable).

1.3.3.2 TCP

L'objet TCP étends l'objet Couche4, il est constitué :

— port Source ;
— port Destination ;
— taille ;
— flags ;
— charge utile ;

En fonction du flag on peut determiner à quel niveau du "handshake TCP" on est : SYN, SYN-ACK, ACK, Fin, Fin-ACK.

1.3.3.3 UDP

L'objet UDP étends l'objet Couche 4, il est constitué :

— Port Source ;
— Port Destination ;
— taille