README

Bonjour, notre projet s'intitule "Décodeur et Visualiseur de flux" et comme son nom l'indique, il permet au choix de décoder une trame ou de visualiser le flux de trafic circulant dans le réseau.

Il contient 9 classes au total dans le package Protocole : Octet, DemiOctet, EthernetEntete, IPV4, TCP, HTTP, Trame, ConteneurTrame, MainClasse.

Octet permet de stocker les valeurs de deux chiffres hexadécimaux nécessaires pour convertir les String en Octet, elle nous sera utile dans toutes les classes

Demi Octet qui permet de stocker la valeur d'un chiffre hexadécimal représentant 4 bits, elle nous sera utile pour les champs comme IHL, les flags et autres.

EthernetEntete permet de stocker tous les champs Ethernet dont elle contient l'information et permet même de traiter le cas d'une trame avec préambule.

IPV4 permet de stocker les champs IP dont elle contient les informations et offre la possibilité de décoder l'option **RecordedRoute**.

TCP à l'instar des autres protocoles décrits ci-dessous contient tous les champs TCP et permet de décoder les options suivantes : End of Option List-No-Operation-Maximum Segment Size-WSOPT - Window Scale-SACK Permitted-TSOPT - Time Stamp Option

HTTP contient deux méthodes toString(): la première permet de convertir tous les chiffres hexadécimaux en ascii, la deuxième ne fait cette conversion que pour la première ligne (ligne de requête GET pour une requête et la ligne d'envoi du code statut pour la réponse).

Trame comme son nom l'indique, permet de représenter une trame grâce à une liste de String. Elle contient les champs pertinents pour la partie Visualisation de Traffic (@IP et numéros de Port...) et contient des getters indispensables pour l'interface graphique.

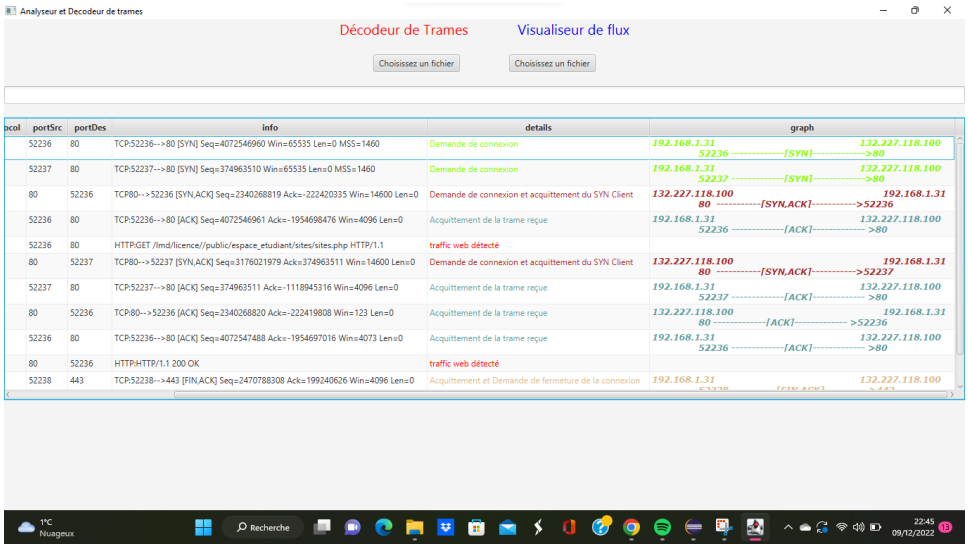
Conteneur Trame fait la lecture du fichier d'entrée et stocke les trames qu'il contient dans une liste de Trames.

Et Enfin la ***MainClasse*** permet de lancer le programme par appel à "launch" qui permet d'ouvrir la fenêtre graphique. Cette classe offre deux fonctionnalités : la première se déclenche quand l'utilisateur appuie sur le bouton B1 du décodeur de Trame et utilise un "Label" pour afficher la trame décodée sur l'interface graphique et permet aussi d'enregistrer automatiquement le fichier issu du décodage et qui va porter le nom du fichier d'entrée+"decoded"; la deuxième se déclenche à l'appui de B2 du visualiseur de flux, utilise une "tableView" pour reproduire les tableaux et le FLOWGAPH de wireshark et permet aussi de fitlrer les trames selon le protocol, les @IP et les n° de port et ceci grâce au "textField" en haut de la fenêtre.

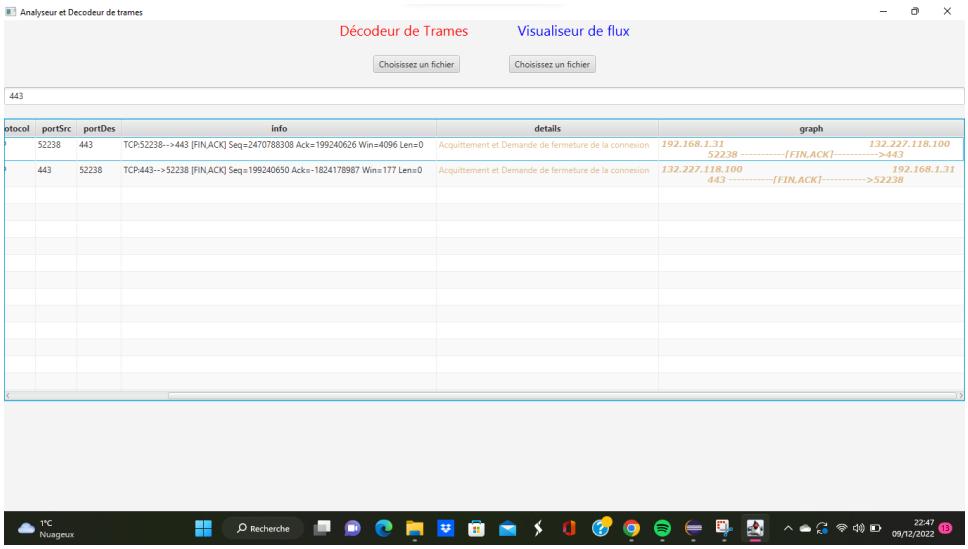
Remarque : Si l'utilisateur décide de faire du décodage et donne à l'entrée un fichier contenant plusieurs trames, on a fait le choix de décoder uniquement la première trame pour ne pas encombrer la fenêtre graphique.

Exemple Décodeur Trame

Exemple Analyseur flux :



Après Filtrage sur le port 433 :



Exemple avec un protocole que nous n’avons pas décodé :