Projet de Conception sur la communication réseau

Objectifs – Effectuer une implémentation « *fonctionnelle* » de type cheval de Troie
Mise en œuvre d'une communication réseau entre 2 applications
Conception et utilisation d'un protocole de communication réseau

Date de remise

Remise ***AVANT*** le ***23 Avril à 23h59***

Remise *(1 seule remise par équipe)*

Sur ***1 seul fichier archive*** (ZIP, 7z, rar, …) sur ***Moodle*** contenant la totalité des documents requis

- Une vidéo explicative de votre projet (***OBLIGATOIRE***)
  - Vous devez obligatoirement démontrer le bon fonctionnement, les différentes fonctionnalités et particularités de votre implémentation dans votre vidéo
  - Vous être limité à une vidéo de 8 minutes par équipe
  - Chacun des coéquipiers doit parler/expliquer équitablement dans la vidéo
- Un bref ***rapport en format éditable*** : MsWord, LateX, Open Office, Libre Office, …
- Une version du rapport en ***format PDF***
- Votre rapport doit contenir les instructions de compilation et d'utilisation de votre projet
- Un package, un installer, ou toute autre mécanique permettant l'exécution de votre « programme/implémentation » sur le poste de l'enseignant
- Tous le code de votre/vos implémentation(s) (*solution, projet, librairies, etc*..)

Directives générales

- Ce travail doit être réalisé ***seul*** ou ***en équipe de 2 personnes***
- Assurez-vous de rendre ***un projet complet***, ***fonctionnel*** avec le nécessaire pour que le correcteur soit ***apte à effectuer l'exécution*** afin de tester le fonctionnement de l'implémentation
- Votre implémentation devra être « ***exécutable*** » sur un ordinateur autre que le vôtre (*sous Windows 11*) (*Vous pouvez fournir une VM (Machine Virtuelle) de votre environnement de développement si vous utilisez des configurations spécifiques et/ou non standards → sur Clef USB ou hébergé avec un lien pour le télécharger*)
- Votre ***implémentation doit être effectuée en C++,*** mais ne doit pas obligatoirement être développée avec Visual Studio*

Sous validation et approbation de l'enseignant

Critères d'évaluation :

- Format et présentation de l'implémentation et des résultats
- Clarté, concision et exhaustivité de la vidéo et de son contenu
- Tout en restant concis : Complétude et pertinence des différentes sections du rapport (***captures d'écran***)
- Recommandations et améliorations potentielles pour une implémentation améliorée future
- Qualité de l'arborescence/structure de l'archive rendue (*livrable final contenant tous les éléments*)

Étapes du mini-projet :

Recherche, développement et implémentation(s) :

- Effectuer quelques recherches afin d'identifier un ou plusieurs projets « open source » (*sur GitHub, SourceForge ou autre*) afin de vous permettre de voir le fonctionnement d'une communication client-serveur, et potentiellement vous donner une structure de base au niveau de la connexion et de la communication
  - Vous devrez fournir le/les lien(s) et les sources du/des projet(s) utilisés afin d'effectuer ces expérimentations
  - Les exigences du projet devront être respectées dans votre implémentation finale
    - Le nécessaire devra être intégré, commenté et détaillé dans le code ainsi que dans le rapport afin de permettre d'identifier ce qui vient de votre cru et ce qui vient du « projet source »

- Détailler (*brièvement*) dans votre rapport votre démarche, vos expérimentations et votre raisonnement
  - Appuyer vos choix avec des justifications détaillées;
  - N'hésitez pas à identifier les obstacles que vous avez dû surmonter ainsi que les méthodes utilisées afin de les résoudre/contourner.

- Votre projet devra contenir les fonctionnalités suivantes :
  - S'exécuter lors que votre cible double clique sur un fichier exécutable (*.exe*) « *reçu par courriel par exemple…* »
    - Le « serveur » restera alors en attente d'une connexion du client afin qu'il puisse « prendre le contrôle » de l'ordinateur

  - Le client devra permettre de saisir l'adresse IP du serveur afin de s'y connecter (*soit via une saisie utilisateur, soit dans un fichier .ini*)
    - Porter une attention aux validations afin d'assurer la stabilité de vos programmes

  - Le serveur ne devra pas se fermer/arrêter lorsque le client se déconnecte mais rester actif et se remettre en mode « attente de connexion »

- Une fois la connexion établie, le client (*pirate*) devrait pouvoir envoyer n'importe laquelle des commandes listées ici-bas au serveur (*cible*) et que le serveur retourne le résultat au client
  - DIR, CD, MD, RD, DEL, COPY, SHUTDOWN (*commandes de l'invite de commande Windows*)
  - Note importante: *assurez-vous de transmettre également les paramètres des fonctions lors de l'envoie desdites commandes*

- En plus des commandes mentionnées ici-haut, les commandes afin de transmettre un fichier vers le serveur ainsi que les commandes afin de récupérer un fichier du serveur devront être implémentées

Défis :

- Si le programme serveur (*du côté de la personne infectée*) est exécuté sans afficher de console ou
- Si le programme serveur s'exécute en tant que service Windows

- Si la communication entre le client et le serveur est « chiffrée, obfusquée, … »

- Si le serveur permet l'exécution de plusieurs commandes natives de l'invite de commande en plus de celles de base spécifiées

Astuces:

- Tester les commandes dans un invite de commande directement sur le serveur (*localement*) afin de s'assurer que votre envoi et exécution de la commande produit et transfert le résultat adéquatement au client
- Vous pouvez exécuter des commandes de l'invite de commande en C++ à l'aide de :
  - ShellExecute(), ShellExecuteEx(), system() ou autres mécanismes / API

Liens:

https://docs.microsoft.com/fr-fr/windows/win32/shell/launch?redirectedfrom=MSDN

https://docs.microsoft.com/fr-fr/windows/win32/api/shellapi/nf-shellapi-shellexecutea?redirectedfrom=MSDN

https://docs.microsoft.com/fr-fr/windows/win32/api/shellapi/nf-shellapi-shellexecuteexa?redirectedfrom=MSDN

https://docs.microsoft.com/fr-fr/windows/win32/api/processthreadsapi/nf-processthreadsapi-createprocessa?redirectedfrom=MSDN

Clarifications :

Serveur : Cible
Client : Pirate