

# MODULE 06

## SÉANCE SYSTÈME 02

### TP D'INFORMATIQUE

Durée 2h30

## CODAGE DE LA REQUÊTE HTTP EN C++ : EXTRACTION DE LA LISTE DES APPAREILS (EXTRACTION XML)

### BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

### COMPÉTENCE(S)

C08 - CODER

### OBJECTIFS PÉDAGOGIQUES

Codage d'un client TCP permettant d'envoyer les requêtes GET nécessaires à l'identification des appareils présents dans l'installation.

### CONNAISSANCES ISSUES DU RÉFÉRENTIEL

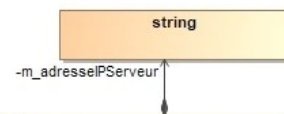
- Programmation orientée objet Niveau 3
- Programmation réseau Niveau 3
- Outils logiciels d'évaluation, de traçabilité de l'information Niveau 3

### CONNAISSANCES OPÉRATIONNALISÉES

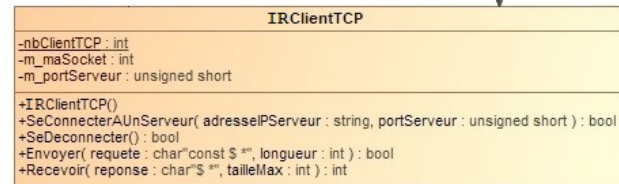
- Envoyer une trame TCP Niveau 3
- Analyser une communication réseau avec Wireshark Niveau 3
- Versionner un code Niveau 3

## TP

## Envoi de la requête GET et affichage de la réponse XML



Créer le programme C++ permettant de se connecter au serveur HTTP (classe IRClientTCP). Envoyer la requête GET identifiée dans la dernière séance (MS06SS01). Votre code :



```

client.SeConnecterAUnServeur("172.20.21.38",80);

client.Envoyer("GET services/user/values.xml?var=SP%200A83.STATUS HTTP/1.1 \r\n\r\n");
  
```

La réponse est contenue dans 2 trames TCP / IP : appeler deux fois la méthode Recevoir() de la classe IRClientTCP, en concaténant les réponses dans un string. Penser à se déconnecter du serveur. Afficher la réponse et le contenu XML. Votre code :

Bonus : placer dans un fichier de configuration l'adresse et le port, qu'un objet de la classe ifstream sera chargé de lire. Intégrer ces paramètres dans votre code ainsi qu'à l'intérieur de la requête HTTP à envoyer au serveur.

```

172.20.21.26
80
  
```

## TP Extraction de la liste des appareils présents dans la réponse XML

Rechercher la documentation de la méthode `find` de la classe `string` : donner un exemple d'utilisation :

```
str.find("Salut")
```

Quelle valeur est retournée lorsque la chaîne de caractères n'est pas trouvée ?

```
18446744073709551615
```

Rechercher la documentation de la méthode `substr()` de la classe `string` : donner un exemple d'utilisation :

Elle permet de récupérer un certain nombre de caractères dans une `string`

A l'aide de la méthode `find()` rechercher dans la réponse XML du serveur la chaîne "`<id>`" en plaçant l'indice retourné dans une variable entière `debut`. Votre code :

```
str.find("<id>");
```

Rechercher ensuite la chaîne "`</id>`" en plaçant l'indice retourné dans une variable entière `fin`. Votre code :

```
str.find("</id>");
```

Afficher le contenu des variables `debut` et `fin` et vérifier leur valeur en comptant les caractères dans la réponse XML (utiliser Notepad ++ pour vérifier les indices).

Utiliser la méthode `substr()` de la classe `string` afin d'afficher le nom du premier appareil : justifier les valeurs des arguments :

```
substr(debut+4, fin-debut-4)
```

Mettre en place une boucle `do...while` permettant d'extraire le nom de tous les appareils : `do...while(debut!=-1)`.

Lancer une capture Wireshark permettant de visualiser la connexion TCP, l'envoi de la requête, la réception des données XML ainsi que la déconnexion. Quel filtre utiliser ?

```
xml && ip.addr==172.20.21.38
```

Versionner le code complet.

## Bonus

Gérer les requêtes permettant d'avoir les informations d'un appareil (/services/user/deviceInfo.xml?id=SP...), %20 remplacera les espaces présents dans le nom de l'appareil.

Gérer les requêtes permettant d'obtenir l'état d'un capteur (/services/user/values.xml?var=SP... .STATE)

Versionner le code complet.