

Week-end Challenge

03

Sujet :

Pour ce challenge, vous apprendrez les bases de la programmation réseau en développant un simple client TCP en C.

Règles :

- Un seul rendu par étudiant, mais vous pouvez vous entre-aider a condition de ne pas avoir le même rendu
- Votre code doit être écrit en C et doit compiler avec les flags suivants: `-Wall -Wextra -Werror`
- Essayez de respecter la norme le plus possible pour que votre code reste lisible, mais vous pouvez prendre quelques libertés: fonctions plus longues (plus que 25 lignes), utilisation du `switch` ou du `for`, etc

Partie obligatoire :

nom du programme	<code>mini-tcp</code>
fichiers de rendu	<code>client.c</code>
arguments	le domaine puis le port sur lequel envoyer la requête
Librairies autorisées	toutes les fonctions de la libc
Description	Votre programme se connectera à un serveur distant en TCP.

Description generale

Vous allez développer un programme en utilisant les fichiers `Makefile`, `wec03.h` et `hash_file.c` fournis dans la section ressources. le seul fichier que vous allez écrire s'appellera `client.c`.

Votre projet devra réaliser ces simples tâches:

- Créer une connection TCP au serveur distant dont le domaine est `challenge.cln.ac` avec le port `1942`
- Vous envoyez la string suivante au serveur: `? [login] [hash]` avec votre login d'intra a la place de `[login]` et le hash de votre fichier `client.c` a la place de `[hash]` (cf. sections suivante)

- Le serveur répond avec la string suivante: `= [code]` ou `[code]` est un entier aléatoire entre 0 et 46340
- Vous retournez ensuite la string suivante: `! [code2]` ou `[code2]` est le code donné par le serveur mis au carré
- Si toutes les données sont correctes, le serveur renvoie `success` et termine la connexion.

Le but de ce projet est de réaliser cette tâche le plus rapidement possible, le serveur distant enregistre les requêtes valides et les stocke sur une base de données.

Note: même si vous avez déjà envoyé une requête valide, vous devrez obligatoirement en envoyer une nouvelle si vous changez le code de votre client.c. Cela mettra à jour votre hash et date de soumission. Le fichier client.c utilisé pour faire la requête doit être exactement le même que le fichier envoyé par mail.

Hashing

Dans ce projet, nous vous demanderons d'envoyer au serveur le contenu de votre fichier client.c passé dans une fonction de hashing. Ne vous inquiétez pas, nous avons déjà écrit cette fonction et vous pouvez l'utiliser comme suit:

```
unsigned long hash = hash_file("./client.c");
```

Ce nombre 'hash' devra être envoyé au serveur en même temps que votre login. Ceci est une protection pour vous empêcher de tricher, car nous vérifierons si le hash du fichier rendu correspond au hash que vous avez envoyé.

Note: une fonction de hashing retournera un nombre différent même si un seul caractère est différent. Nous vous conseillons donc de ne pas inclure de 42header dans votre fichier client.c pour éviter que la ligne 'Updated:' ne change si vous re-sauvegardez votre fichier.

Simulation

Le serveur principal(1942) termine la connexion automatiquement après 3 secondes pour éviter que vous ne puissiez entrer les valeurs à la main. Pour vous permettre de comprendre plus facilement le comportement du serveur, vous pouvez interagir et faire des tests avec le serveur de simulation dont le domaine est aussi

`challenge.cln.ac` mais sur le port 1943. Ce que vous envoyez sur ce serveur ne sera pas sauvegardé sur la base de données. Voici un exemple d'utilisation valide en ligne de commande (les lignes jaunes sont entrées par l'utilisateur):

```
> telnet challenge.cln.ac 1943
Trying 51.178.40.137...
Connected to 137.ip-51-178-40.eu.
Escape character is '^]'.
? ancoulon 7894307453
= 3758
! 14122564
success
Connection closed by foreign host.
>
```

Vous devrez évidemment repliquer ce comportement en C, sans utiliser un exécutable tel que telenet.

Ressources :

- [Fichiers C fournis](#)
- [Github du WEC](#)
- [Guide de programmation réseau en C \(en\)](#)
- [Implementation C d'un serveur-client TCP \(en\)](#)

Rendu :

La première étape du rendu est d'envoyer une requête valide au serveur de rendu (port 1942). lorsque cela est fait, envoyez le fichier client.c utilisé par email a tutors@s19.be avec le sujet suivant: `wec03 - [login]`.

Rendu au plus tard dimanche 10:19pm.

Résultats :

Les résultats seront donnés le lundi suivant. Les points seront distribués en fonction de l'ordre dans lequel les requêtes valides sont reçues. Ce challenge est donc une course!