Programming Assignment

Course: Computer Networks
Professor Yeali S. Sun

November 28, 2019

1. The Problem of "A Secure Person2Person (P2P) Micropayment System" （安全的第三方支付使用者對使用者小額付款系統）

一套具安全傳輸的簡單網際網路第三方支付使用者對使用者小額付款（Micropayment）系統。此系統包含三大功能。一、第三方支付Server端對Client端（使用者）的統一管理，包含帳號管理、好友名單管理、認證以及Client帳戶管理等。二、 Client間即時通訊。 三、 Client與Server以及Client間的通訊，都可以各自加密，加密的鑰匙（encryption key，又稱secret key）由當下通訊的雙方議定。

本作業的目標是設計與實作一套簡單的好友間轉帳功能。同學將設計、實作一套安全傳輸的簡單「安全的第三方支付使用者對使用者小額付款系統」包含Client 與 multithreaded Server端的軟體，以及安全傳輸的軟體撰寫。

Client端的兩個主要功能：

- 安全的與第三方支付 Server的通訊
- 一對一安全的Client間對談

Multi-threaded server 端的主要功能：

- 接受 client 的安全連結，並根據要求(request)回覆訊息（reply）

安全通訊的主要功能：

- 每一個Client 與 Server間，以及Client間的通訊，都必須加密，加密的鑰匙（encryption key,又稱secret key）由當下通訊的雙方議定。

2. Requirements

本次作業分三階段繳交。第一階段，同學要先完成Client端的程式；第二階段完成Multi-threaded server端程式；第三階段完成安全通訊的功能。

A. Client

第一階段client端程式的開發，同學必須要能 a) 登入助教所提供server端程式（填入使用者名稱、Port Number），及b) 接收Server端回覆的client目前的帳戶餘額與上線清單訊息。Client端進行離線動作前需主動告知Server端程式。

上線清單包含的訊息為上線的總人數、線上的使用者名稱、其IP address以及可用來通訊的port number。本作業所有的通訊皆須採用TCP protocol 以達到可靠傳輸。

B. Multi-Threaded Server

A multithreaded server is capable of serving multiple requests in parallel. The server will create a separate thread to handle each of the connections for accepted requests. There will also be a main thread, in which the server listens for clients that want to establish connections.

第二階段server端程式的開發要能提供client 端的註冊與登入，發送client目前的帳戶餘額與上線清單的回覆訊息給client，以及接收處理Client端離線前的通知，Server提供的功能**請使用thread及worker pool 的方式進行程式的開發，不要使用fork**。

C. Client 與 Server 溝通

Client 與 Server 間的溝通訊息主要有四種：

(1) Client 端對 Server 註冊：

Client 端傳給 Server 端的訊息為：

```
REGISTER#<UserAccountName><CRLF>
```

Server 端會回給 Client 端註冊成功與不成功的訊息各為：

```
100<space>OK<CRLF>
210<space>FAIL<CRLF>
```

(2) Client 端登入 Server：

Client 端傳給 Server 端的訊息為：

```
<UserAccountName>#<portNum><CRLF>
```

若使用者有註冊過，Server 端會回給 Client 端上線清單，清單格式為：

```
<accountBalance><CRLF>
<number of accounts online><CRLF>
<userAccount1>#<userAccount1_IPaddress>#<userAccount
1_portNum><CRLF>
<userAccount2>#<userAccount2_IPaddress>#<userAccount
2_portNum><CRLF>….
```

若使用者尚未註冊過，Server 會回傳給 Client 端驗證失敗的訊息：

```
220<space>AUTH_FAIL<CRLF>
```

(3) Client 端向 Server 要最新的帳戶餘額與上線清單：

Client 端傳給 Server 端的訊息為：

```
List<CRLF>
```

Server 端會回給 Client 端上線清單，清單格式為：

```
<accountBalance><CRLF>
<number of accounts online><CRLF>
<userAccount1>#<userAccount1_IPaddress>#<userAccount
1_portNum><CRLF>
<userAccount2>#<userAccount2_IPaddress>#<userAccount
2_portNum><CRLF>….
```

(4) Client 端結束程式：

Client 端傳給 Server 端的訊息為：

```
Exit<CRLF>
```

Server 端會回給 Client 端上線清單，清單格式為：

```
Bye<CRLF>
```

(5) Client 端送 micropayment transaction 訊息給 Server：

Client 端之間的訊息傳送格式為：

```
<MyUserAccountName>#<payAmount>#<PayeeUserAccountName><CRLF>
```

本訊息要用 Client 的 PKI private key 加密，再用好友的 public key 加密。

好友收到後用自己的 private key 解密。再用自己的 private key 將訊息加密，送給 server.

訊息內容假設都是 ASCII 7-bit 字元文字(text)內容。

注意事項：

(1). Server端不替Client端做任何訊息的relay。

(2). 可使用的語言及Library：Unix/Linux Socket Programming(in C/C++)、Win Socket；不可使用Java/C#。

D. 安全傳輸

socket 安全傳輸部分，請使用 openssl 這個 open source toolkits https://www.openssl.org/，並且你的 source code 中使用 openssl toolkits 進行加密的安全傳輸。

【套件安裝】

在 Unix/Linux 上(ex.Ubuntu)可直接用下列指令進行安裝

Apt-get install openssl

在 Windows 上請自行上網下載 Openssl for winsocket 版本

3. 作業繳交

本次作業分三階段繳交：

A. 第一階段：Client 端程式

助教提供的 Server 端程式需在 Linux kernel 2.6.x 環境上執行。同學可以裡用這個程式來測試自己的 Client 端程式功能是否正常。執行 server 端程式 command 的格式為：

```
$ ./<server_name><space><portNum><space><Option>
```

Option的選項有-d,-s,-a，說明如下：
(每次執行只能輸入一個Option參數)
-d：Server只會顯示簡單的訊息表示Client註冊、登入或離開。
-s：除了以上，Server在每次有Client登入或離開時都會顯示現在上線的清單。
-a：除了以上，Server還會顯示每一次Client與Server之間的訊息傳送。
此Option是為了方便同學除錯，顯示的訊息可以當作參考。
Server name為程式執行檔名稱，port必須在1024到65535之間。

B. 第二階段：Server端程式
完成 Server 端的程式。可以用你的Client端的程式與Server端的程式一起執行。

C. 第三階段：Client端以及Server端安全通訊程式
完成Client端以及Server端的安全通訊程式。

4. Demo

A. 第一階段：助教會在系上工作站上執行Server端程式(Server IP之後助教由信件公布)。測試Client端程式是否能登入Server以及和其它Client傳輸訊息。使用Unix/Linux Socket撰寫程式的同學，用工作站執行Client端程式進行demo；使用Win Socket撰寫程式的同學，請自行準備Client端執行環境，或事先確認管五電腦實驗室的電腦是否可以run你的程式。

Demo時間：December 10 - 13, 2019
將提供時間表供同學填選

Demo地點：管五電腦實驗室

B. 第二階段：

助教會測試你的Client and Server 程式是否可以正常執行。

Demo時間：December 24 - 27, 2019
將提供時間表供同學填選

Demo地點：管五電腦實驗室

C. 第三階段：

助教會測試你的Client and Server 程式是否可以正常執行。

Demo時間：January 14 - 17, 2019
將提供時間表供同學填選

Demo地點：管五電腦實驗室

5. Summission

A. 第一階段：

(i). 需繳交Source code以及說明文件

- 上傳繳交的部份包含以下四項；
  1. Source Code (Client端程式的原始碼)
  2. 操作說明文件電子檔
     （包含如何編譯、執行Client端程式，程式需求執行環境等）
  3. Binary執行檔 (已Compile及Linking完成並可執行的Client端程式。)
  4. Makefile 編譯程式
- 請將上述四項檔案壓縮成：學號_part1.tar.gz (e.g. b027050xx_part1.tar.gz)，
  Email至 r06725035@ntu.edu.tw 或 r06725041@ntu.edu.tw。
  主旨:[network part1] 學號 name (e.g., b027050xx XXX)

(ii). Deadline：5 pm, December 13, 2019

B. 第二階段：

(i). 需繳交Source code以及說明文件

- 上傳繳交的部份包含以下四項；
  1. Source Code (Server端程式的原始碼)
  2. 操作說明文件電子檔
     （包含如何編譯、執行multi-thread server端程式，程式需求執行環境等）
  3. Binary執行檔 (已Compile及Linking完成並可執行的Server端程式。)
  4. Makefile 編譯程式
- 請將上述四項檔案壓縮成：學號_part2.tar.gz (e.g. b027050xx_part2.tar.gz)，
  Email至 r06725035@ntu.edu.tw 或 r06725041@ntu.edu.tw。。

  主旨: [network part2] 學號 name (e.g. b027050xx XXX)

(ii). Deadline：5 pm, December 27, 2018

C. 第三階段：

(iii). 需繳交Source code以及說明文件

- 上傳繳交的部份包含以下四項；
  5. Source Code (Server端程式的原始碼)
  6. 操作說明文件電子檔
     （包含如何編譯、執行multi-thread server端程式，程式需求執行環境等）
  7. Binary執行檔 (已Compile及Linking完成並可執行的Server端程式。)
  8. Makefile 編譯程式
- 請將上述四項檔案壓縮成：學號_part3.tar.gz (e.g. b027050xx_part3.tar.gz)，Email至 r06725035@ntu.edu.tw 或 r06725041@ntu.edu.tw。。
- 主旨: [network part3] 學號 name (e.g. b027050xx XXX)

(iv). Deadline：5 pm, January 17, 2019

6. Grading

(a) (20%) Client 端程式的評分方式如下：

- 說明文件：20%
- 基本要求（Client端可以註冊（填入使用者名稱與初始儲值額）、登入Server、Client間可以彼此傳送訊息）：80%
- Bonus（介面、GUI、Exception handling）：15 %

(b) (35%) Server 端程式的評分方式如下：

- 說明文件：20%
- 基本要求 （Server端可以接收多個Client的註冊及登入要求並各用一個thread 處理client 端的連線、提供client 端的註冊（填入使用者名稱與初始儲值額）、登入（填入使用者名稱、port number）、發送上線清單給client，以及接收處理Client端離線前的通知）：80%
- Bonus（介面、GUI、Exception handling）：15 %

(c) (45%) 具安全通訊的Client與Server程式，評分方式如下：

- 說明文件：20%
- 具安全通訊的Client與Server程式：80%
- Bonus（介面、GUI、Exception handling）：15 %