網路技術與應用

Socket Programming Part 3

B08705012 資管三 胡家愷

一、操作說明文件

1. 執行 make 以編譯程式碼
2. 執行 make client 或是 ./client<space><server ip><space><server port> 執行 client 端程式
3. 執行 make server 或是 ./server <space><server port> 執行 server 端程式

二、執行環境：Ubuntu 20.04

三、安全傳輸實作的方法及流程說明

1. Client 端
   a. 生出自己的 private key 跟 public key (RSA2048)
      openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout a.key -out a.crt
   b. 建立 SSL 通道前要先初始化

```
SSL_CTX* InitClientCTX()
{
    SSL_CTX *ctx;
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();
    ctx = SSL_CTX_new(SSLv23_client_method());
    if (ctx == NULL) {
        ERR_print_errors_fp(stdout);
    }
    return ctx;
}
```

   c. 建立 SSL 通道 SSL* ssl = SSL_new(ctx);
   d. 將 SSL 與 socket 做連結 SSL_set_fd(ssl, sockfd);
   e. 連線至 Server SSL_connect(ssl);
   f. 連線之後 server 會馬上傳自己的 public key 過來，把他存下來
   g. 現在訊息傳輸前會先拿 server 的 public key 加密再做傳送
      透過以下函式

```
void encryptAndSend(SSL* ssl, char* msg){
```

```
    BIO* bio = BIO_new(BIO_s_mem());
    int len = BIO_write(bio, serverPubKey,
strlen(serverPubKey));
    EVP_PKEY* evp_key = PEM_read_bio_PUBKEY(bio, NULL,
NULL, NULL);
    RSA *peer_rsa_pubkey = EVP_PKEY_get1_RSA(evp_key);
    char *cipher_text = (char
*)malloc(RSA_size(peer_rsa_pubkey));
    RSA_public_encrypt((strlen(msg) + 1)*sizeof(char),
(const unsigned char*) msg, (unsigned char*)
cipher_text, peer_rsa_pubkey, RSA_PKCS1_PADDING);
    SSL_write(ssl, cipher_text,
RSA_size(peer_rsa_pubkey));
 }
```

h. Server 收到之後會再去做解密

2. Server 端

a. 生出自己的 private key 跟 public key (RSA2048)

openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 -keyout a.key -out a.crt

b. 建立 SSL 通道前要先初始化

```
SSL_CTX* InitServerCTX()
{
    SSL_CTX *ctx;
    SSL_library_init();
    OpenSSL_add_all_algorithms();
    SSL_load_error_strings();
    ctx = SSL_CTX_new(SSLv23_server _method());
    if (ctx == NULL) {
        ERR_print_errors_fp(stdout);
    return ctx;
}
```

c. 在這邊我們要將 server 的 public key 及 private key 載入到 ctx 裡面，讓 SSL 通道有加密功能

```
void LoadCertificates(SSL_CTX* ctx, char* CertFile, char*
KeyFile) {
   // 載入使用者的 certificate。 Certificate 裡含有公鑰
   SSL_CTX_use_certificate_file(ctx, CertFile,
SSL_FILETYPE_PEM)
```

```
    // 載入私鑰
    SSL_CTX_use_PrivateKey_file(ctx, KeyFile,
SSL_FILETYPE_PEM)
    // 檢查使用者憑證與私鑰是否吻合
    if ( !SSL_CTX_check_private_key(ctx) ) {
        fprintf(stderr, "Private key does not match the
public certificate\n");
        abort();
    }
}
```

d. 建立 SSL 通道 SSL* ssl = SSL_new(ctx);

e. 將 SSL 與 socket 做連結 SSL_set_fd(ssl, sockfd);

f. 接收 client 的連線 SSL_accept(ssl);

g. 接收連線之後立刻把自己的 public key 送給 client

h. 用自己的 private key 解密 client 傳來的訊息

```
        FILE *key_file = fopen(pri, "r");
       RSA *privateKey = PEM_read_RSAPrivateKey(key_file,
NULL, NULL, NULL);
        RSA_private_decrypt(RSA_size(privateKey), (const
        unsigned char*)peer_msg_encrypted, (unsigned char
        *)peer_msg, privateKey, RSA_PKCS1_PADDING);
```

3. 轉帳的過程也是類似上面的過程，client a 傳給 client b 時以 client b 作為 server，client a 作為 client 進行上述的傳輸，client b 解密後再使用 server 的 public key 加密依正常流程傳給 server。

四、參考資料：

1. How to write a multithreaded server in C (threads, sockets) - YouTube
2. Multithreaded Server Part 2: Thread Pools - YouTube
3. /docs/man3.0/man3/index.html (openssl.org)
4. socket 编程之 openssl 入门_我们的征途是星辰大海-CSDN 博客_openssl socket
5. ssl server client programming using openssl in c - Aticleworld
6. SSL 握手通訊詳解及 linux 下 c/c++ SSL Socket(另附 SSL 雙向認證客戶端程式碼) - IT 閱讀 (itread01.com)
7. OpenSSL& public key and private key & Certificate | by 莊子弘 | Medium
8. C++ (Cpp) RSA public encrypt Examples - HotExamples
9. C++ X509 get pubkey 函數代碼示例 - 純淨天空 (vimsky.com)
10. C++ EVP PKEY get1 RSA 函數代碼示例 - 純淨天空 (vimsky.com)