Lab3Analysis

Wenjie Bai

April 10, 2020

1 Task1

1.1 Setup

Server on machine 192.168.15.4; tunnel Point A 10.0.1.1.

Client on machine 192.168.15.5; tunnel Point B 10.0.2.1.

on machine 192.168.15.4: run sudo ip addr add 10.0.1.1/24 dev tun0 sudo ifconfig tun0 up sudo route add -net 10.0.2.0 netmask 255.255.255.0 dev tun0

on machine 192.168.15.5 run sudo ip addr add 10.0.2.1/24 dev tun0 sudo ifconfig tun0 up sudo route add -net 10.0.1.0 netmask 255.255.255.0 dev tun0

When ping each other, results showed they are connected through tunnel points.

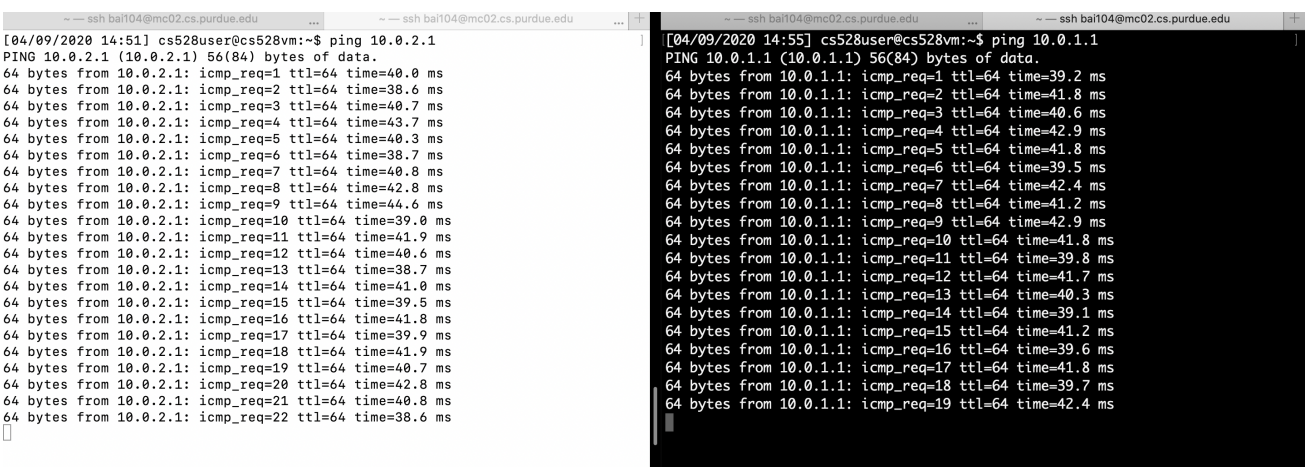

Figure 1: Task1

Q: W hy not use TCP in the tunnel?

A: Because the communicating peer needs to perform encryption and decryption in the VPN tunnel ends, which requires all data at hand. Since TCP might introduce data segmentation, encryption/decryption of partial data is not preferable.

2 Task2

The message in UDP channel is encrypted with AES and integrity is guaranteed by HMAC-SHA256. (details are in server.c and client.c)

Q: Why not use self-implemented encryption/MAC algorithm?

A: Because it is hard to devise an encryption/MAC algorithm. Self-implemented algorithm are prone to errors and lack of mathematical security proofs.

3 Task3

3.1 step 1 authenticating VPN Server

I generate server.key, server.crt and ca.crt using openssl commands. On the server side, in TSL setup, server.c loads the server's private key and the certificate file. Those files are sent to the client. On the client side, client.c load the ca.crt file and verity the authenticity of the server.

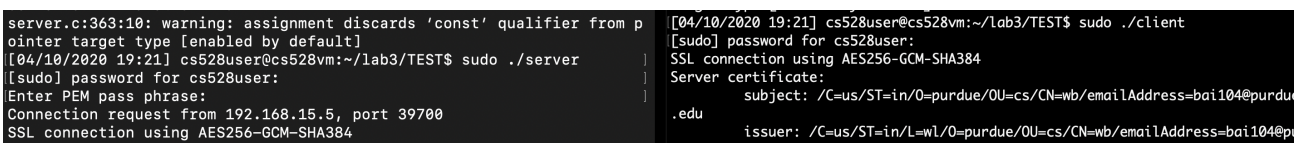

Figure 2: authenticating server

3.2 step 2 Authenticating VPN Client

First, I create an account on the server.

```
\adduser bai104
```

with password being 123456.

The records are stored in `/etc/shadow` file, server.c retrieves the record using function `getspnam()` which is defined in `shadow.h` and compares record with the hashed user password, if there is a match, then the user is legitimate.

```
server.c:363:10: warning: assignment discards 'const' qualifier from p
ointer target type [enabled by default]
server.c:454:5: warning: format '%u' expects argument of type 'unsigne
d int', but argument 2 has type 'unsigned char *' [-Wformat]
[04/10/2020 20:37] cs528user@cs528vm:~/lab3/othertasks$ sudo ./server
[sudo] password for cs528user:
Enter PEM pass phrase:
Connection request from 192.168.15.5, port 39704
SSL connection using AES256-GCM-SHA384
name: bai104
pwd: $6$p3t56b/.$HE5S.zOgAtPG4psCMKBbY/KWyZ9cl3rUPJLifBEbuO3IVE4/GzReE
zvGKZBFEkNGup/xFrPQWOM0Qq7mJXFOv0
authentication results: 1
```

```
[04/10/2020 20:37] cs528user@cs528vm:~/lab3/othertasks$ sudo ./client
[sudo] password for cs528user:
SSL connection using AES256-GCM-SHA384
Server certificate:
        subject: /C=us/ST=in/O=purdue/OU=cs/CN=wb/emailAddress=bai104@purdue
.edu
        issuer: /C=us/ST=in/L=wl/O=purdue/OU=cs/CN=wb/emailAddress=bai104@pu
rdue.edu
Enter login username:
bai104
Enter password:
123456
user authentication success
```

Figure 3: key exchange

3.3 step 3 key exchange

A session key is shared through TCP channel which is run in parent process. Then I use Inter-process communication function `pipe` to send the shared key to UDP tunnel, which is run in child process.

```
server.c: In function 'main':
server.c:363:10: warning: assignment discards 'const' qualifier from p
ointer target type [enabled by default]
server.c:454:5: warning: format '%u' expects argument of type 'unsigne
d int', but argument 2 has type 'unsigned char *' [-Wformat]
[04/10/2020 20:37] cs528user@cs528vm:~/lab3/othertasks$ sudo ./server
[sudo] password for cs528user:
Enter PEM pass phrase:
Connection request from 192.168.15.5, port 39704
SSL connection using AES256-GCM-SHA384
name: bai104
pwd: $6$p3t56b/.$HE5S.zOgAtPG4psCMKBbY/KWyZ9cl3rUPJLifBEbuO3IVE4/GzReE
zvGKZBFEkNGup/xFrPQWOM0Qq7mJXFOv0
authentication results: 1
shared key: 3220711678
parent process, TCP tunnel
shared key: 3220711678
```

```
[04/10/2020 20:37] cs528user@cs528vm:~/lab3/othertasks$ sudo ./client
[sudo] password for cs528user:
SSL connection using AES256-GCM-SHA384
Server certificate:
        subject: /C=us/ST=in/O=purdue/OU=cs/CN=wb/emailAddress=bai104@purdue
.edu
        issuer: /C=us/ST=in/L=wl/O=purdue/OU=cs/CN=wb/emailAddress=bai104@pu
rdue.edu
Enter login username:
bai104
Enter password:
123456
user authentication success
child process, TCP tunnel
child process, UDP tunnel
shared key: 3219518398
Allocated interface tun0.
```

Figure 4: key exchange

3.4 step 4 securing the tunnel

As with task2, confidentiality is achieved by AES, and integrity is achieved by HMAC.

The VPN can defend against man-in-the-middle attack. We use CA to verify server and use password to verify user.

3.5 step 5 break the tunnel

`SSL_shutdown(ssl);`

Q: Why is it important for the server to release resources? A: Because typically there are a lot of client requests, if the server does not close a connection, then the resources will

soon be depleted.