实验 1 TLS 配置与流量分析

1.1 【实验目的】

1） 理解 TLS 协议原理；

2）掌握 apache 服务器的 HTTPS 部署方法；

3）掌握 TLS 流量分析方法。

1.2 【实验内容】

1） 配置 TLS 协议分析环境；

2） 配置 apache 服务器的 https 协议；

3） 对指定域名发起 HTTPS 请求，抓包分析 TLS 协议流程、提取其中的关键信息。

1.3 【实验原理】

本实验主要用到 TLS 协议的原理。TLS 的密码学安全目标包括：保密性、完整性、身份认证

对于保密性来说，通常是通过对称加密组件实现。对称加密的前提是通信双方需要有共享密钥，因此需要一个密钥协商组件。TLS 的设计中，将上述功能分为：

（1）对称加密传输的记录协议，即：Record Protocol

（2）认证密钥协商的握手协议，即：Handshake Protocol

另外，还有三个辅助协议：

（1）Change Cipher Spec 协议

（2）Alert 协议

（3）Application Data 协议

因此，在设计上，TLS 协议是由 TLS 记录协议（Record Protocol）和 TLS 握手协议（Handshake Protocol）两层协议构成。记录协议位于下层，握手协议位于上层，记录协议对上层数据包进行封装，然后利用 TCP 协议进行传输。握手协议又进一步划分为 4 个子协议。如图 1 所示：

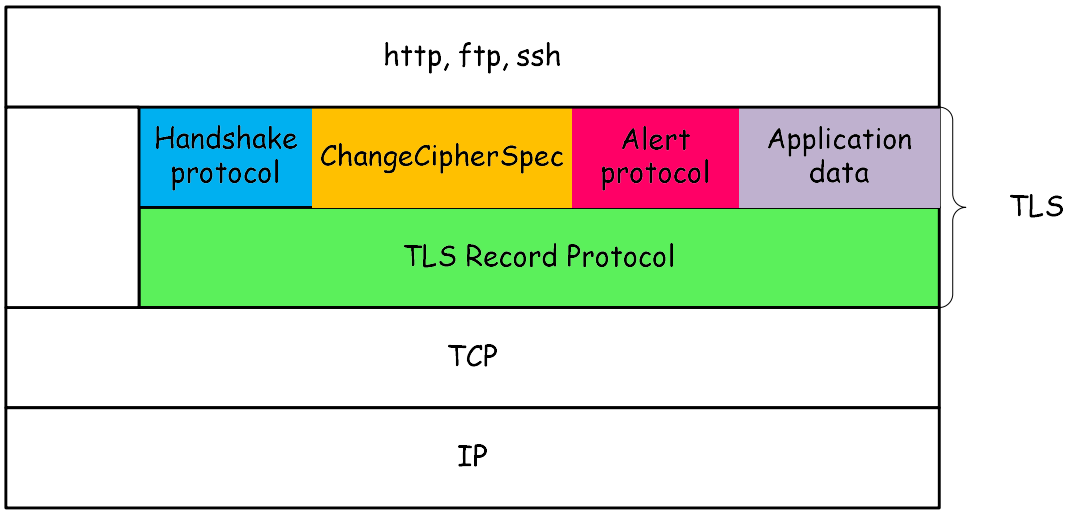


图 1 TLS 协议分层结构

记录协议负责实际的数据传输，需要确保传输数据的保密性和完整性，因此需要使用对称密码和消息认证码，而具体使用的算法及相关参数则是通过上层的握手协议来协商的。握手协议又分为 4 个子协议：handshake protocol, change cipher specific, alert protocol, 和 Application data 协议。这些协议又是如何完整协商的呢？下面我们来稍微具体地说明各协议的功能。

（1）TLS 记录协议

位于 TLS 协议的下层，负责安全传输数据，也就是确保数据传输的保密性和完整性。保密性和完整性通过对称密码和消息验证码来完成，因为候选的加密算法和消息验证码算法很多，因此需要通信双方协商来确定一致性，而这个协商过程就是由握手协议完成。

（2）TLS 握手协议

TLS 握手协议又细分为：握手协议（Handshake Protocol）、变更密码规格协议（ChangeCipherSpec Protocol）、警告协议（Alter Protocol）和应用数据协议（Aplication data Protocol）。

1) 握手协议

负责在客户端和服务器之间协商密码算法和共享密钥，同时完成基于证书的认证操作，为后面的应用数据传输做准备。握手协议的目的是使得通信两端就加密算法和相应的安全参数（比如共享密钥等）达成一致，在这种条件下，通信双方才能够正确理解（正确解密）

握手协议相当于下述对话：

客户端:“你好。我支持的协议版本号是 1.2，我能够支持的密码套件有 RSA/3DES、DSS/AES，请问我们使用哪个密码套件来通信？”

服务器:“你好哦。我们就用 RSA/3DES 密码套件；我的证书也给你看。”

客户端和服务器通过握手协议协商一致后，就会相互发出信号来切换密码，负责发出信号的就是下面的变更密码规格协议。

2) ChangeCipherSpec 协议

负责向通信对端传达变更密码规格的信号。

这个协议发送的消息相当于下面的对话。

客户端:“好，我们按照刚才的约定切换密码吧。”

如果在协议执行的中途发送错误，则会通过下面的警告协议传送相关信息给对端。

3) Alter 协议

负责在发生错误时将错误信息传给对端。

相当于如下对话:

服务器:“刚才的消息无法正确解密!”

变更密码规格后，如果没有出现错误，则会使用应用数据协议进行应用数据传输。

4) Application data 协议

负责将 TLS 承载的应用数据传递给通信对象。具体来说，就是把 http、ftp、smtp 的数据流传人 record 层做处理并传输。

1.4 【实验步骤】

步骤一、环境搭建

配置两台网络联通的主机，可以分别为 ubuntu linux 操作系统和 windows 操作系统。可以采用如下任何一种方式，其中 linux 主机作为服务器，windows 主机作为客户机。

1) Windows 宿主机安装 wireshark 并配置一台 ubuntu linux 虚拟机，确保宿主机和虚拟机能够网络联通，并测试是否能够抓包。

2) 在虚拟化平台配置两台虚拟机，其中之一为 windows/linux 操作系统（其上配置 wireshark 抓包软件），另一台为 ubuntu linux 操作系统，确保这两台虚拟机网络联通。

步骤二、Linux 服务器上配置 apache 服务器的 HTTPS

1、 创建证书、密钥等文件

命令：

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key                    -out
/etc/ssl/certs/apache-selfsigned.crt
```

openssl: 创建和管理 OpenSSL 证书、密钥等的基本工具。

req：子命令，用于说明要使用 X.509 证书签名请求（CSR）管理。

-x509：进一步说明要创建一个自签证书。

-nodes：用于说明跳过对证书的口令保护，这样就避免每次读取证书文件时需要输入口令。

-days 365：设置证书有效期为 365 天。

-newkey rsa:2048：用于说明要同时生成一个新的证书和新的密钥。rsa:2048 用于说明生成的 RSA key 是 2048 比特。

-keyout：说明私钥文件的存放位置

-out：说明证书文件的存放位置

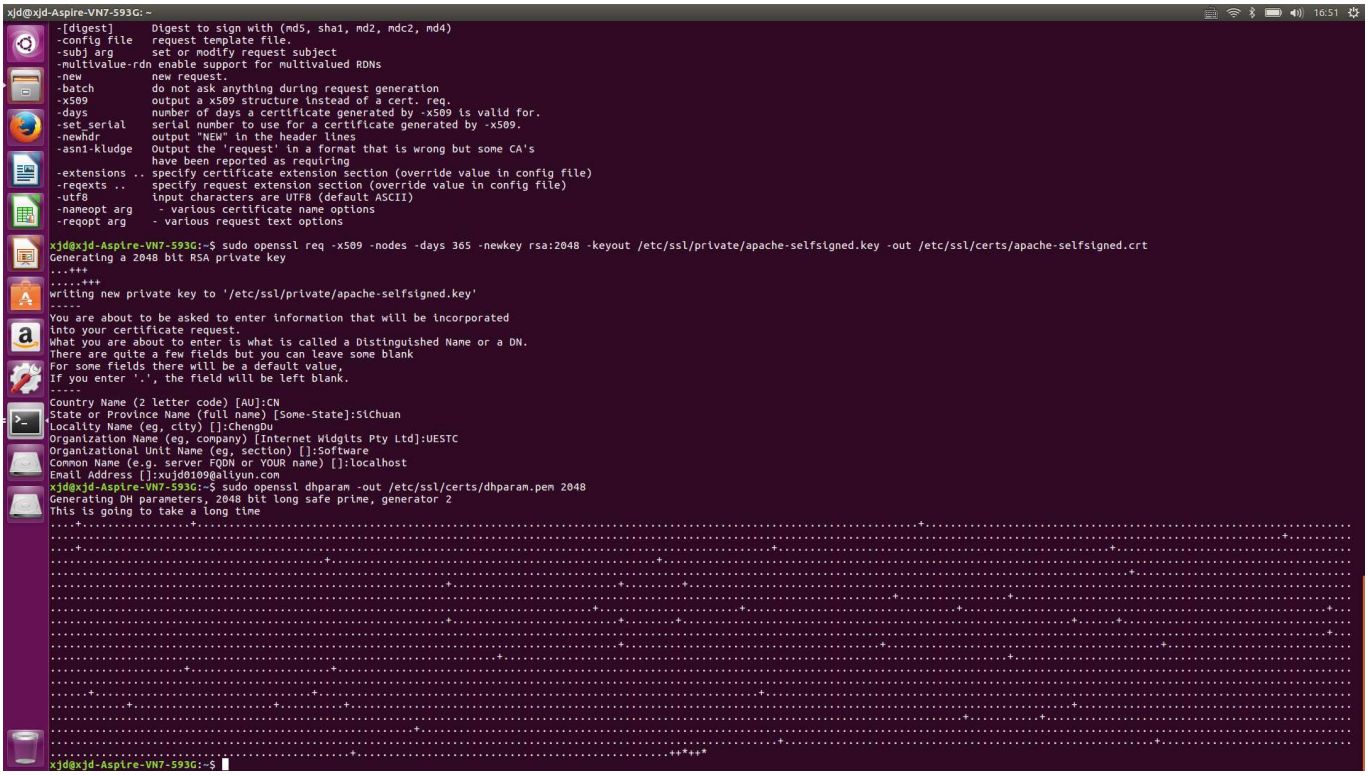
在创建证书的过程中，程序会提示用户输入一些信息，这些信息会嵌入到证书中。最重要的信息是 Common Name，你需要以服务器的域名或者 IP 地址来回答。下面是截图示例：

```
xjd@xjd-Aspire-VN7-593G: ~
 -nameopt arg     - various certificate name options
 -reqopt arg      - various request text options

xjd@xjd-Aspire-VN7-593G:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2
048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-se
lfsigned.crt
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to '/etc/ssl/private/apache-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:SiChuan
Locality Name (eg, city) []:ChengDu
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UESTC
Organizational Unit Name (eg, section) []:Software
Common Name (e.g. server FQDN or YOUR name) []:localhost
```

为了实现前向安全保密（Perfect Forward Secrecy），还需要创建一个强的 DH 群。命令为：

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```



2、 配置 Apache 服务器

前面已经准备好密钥和证书文件，下面我们将利用这些文件来配置 apache 服务器。包括：

（1）安装 apache2

命令：sudo apt-get install apache2

启动服务：

sudo service apache2 start

或者：

sudo /etc/init.d/apache2 start

或者

sudo systemctl start apache2.service

上述命令有对应的 stop 和 restart

查看状态：

sudo systemctl status apache2.service

启动之后，可以通过浏览器访问对应的 URL 来验证是否成功。

（2）创建 Apache 配置文件,并进行强加密设置

在 /etc/apache2/conf-available 目录下创建一个新文件，命名为：ssl-params.conf。文件内容为：

```
# from https://cipherli.st/

# and https://raymii.org/s/tutorials/Strong_SSL_Security_On_Apache2.html



SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH

SSLProtocol All -SSLv2 -SSLv3

SSLHonorCipherOrder On

# Disable preloading HSTS for now.  You can use the commented out header line that includes

# the "preload" directive if you understand the implications.

#Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains; preload"
```

```
Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains"

Header always set X-Frame-Options DENY

Header always set X-Content-Type-Options nosniff

# Requires Apache >= 2.4

SSLCompression off

SSLSessionTickets Off

SSLUseStapling on

SSLStaplingCache "shmcb:logs/stapling-cache(150000)"



SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/dhparam.pem"
```

需要注意的地方是，把上述文件的最后一行的参数设置为前面生成的 DH 文件。

（详细的信息可参看：https://cipherli.st/）

（3）修改默认的 Apache SSL 虚拟主机文件

默认的 Apache SSL 虚拟主机文件为：

/etc/apache2/sites-available/default-ssl.conf

修改之前，先做一下备份。命令为：

sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.bak

然后，打开/etc/apache2/sites-available/default-ssl.conf 文件进行编辑：

原文件的内容大致如下：

```
<IfModule mod_ssl.c>

      <VirtualHost _default_:443>

            ServerAdmin webmaster@localhost



            DocumentRoot /var/www/html
```

```
            ErrorLog ${APACHE_LOG_DIR}/error.log

            CustomLog ${APACHE_LOG_DIR}/access.log combined


            SSLEngine on


            SSLCertificateFile     /etc/ssl/certs/ssl-cert-snakeoil.pem

            SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key


            <FilesMatch "\.(cgi|shtml|phtml|php)$">

                        SSLOptions +StdEnvVars

            </FilesMatch>

            <Directory /usr/lib/cgi-bin>

                        SSLOptions +StdEnvVars

            </Directory>


            # BrowserMatch "MSIE [2-6]" \

            #             nokeepalive ssl-unclean-shutdown \

            #             downgrade-1.0 force-response-1.0


      </VirtualHost>

</IfModule>
```

修改后的版本如下：

```
<IfModule mod_ssl.c>

      <VirtualHost _default_:443>

            ServerAdmin your_email@example.com

            ServerName server_domain_or_IP


            DocumentRoot /var/www/html
```

```
            ErrorLog ${APACHE_LOG_DIR}/error.log
            CustomLog ${APACHE_LOG_DIR}/access.log combined

            SSLEngine on

            SSLCertificateFile     /etc/ssl/certs/apache-selfsigned.crt
            SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

            <FilesMatch "\.(cgi|shtml|phtml|php)$">
                        SSLOptions +StdEnvVars
            </FilesMatch>
            <Directory /usr/lib/cgi-bin>
                        SSLOptions +StdEnvVars
            </Directory>

            BrowserMatch "MSIE [2-6]" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0

      </VirtualHost>
</IfModule>
```

注意对上面红色字体部分是修改后的内容。

（4）修改未加密的 Virtual Host file 来自动重定向请求到加密的 Virtual host

为了更好的安全性，通常也推荐设置为自动重定向 http 访问到 https 访问，通过修改配置文件/etc/apache2/sites-available/000-default.conf 来完成。

打开文件进行编辑，仅需要加入一个 Redirect 指令来指向 SSL 版本的站点，示例如下：

```
<VirtualHost *:80>
        . . .

        Redirect "/" "https://your_domain_or_IP/"

        . . .
</VirtualHost>
```

注意上面的红色字体。

3、 设置防火墙（无防火墙设置的忽略此步）

如果系统设置 ufw 防火墙，则需要进行配置，以使得 SSL 流量能够进入。在安装的时候，apache 会在 ufw 注册一些 profile，以方便设置。命令查看：

```
sudo ufw app list
```

示例结果：

```
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
```

查看当前的设置。命令：

```
sudo ufw status
```

输出示例：

```
Status: active

To                         Action      From
--                         ------      ----
```

```
OpenSSH                    ALLOW       Anywhere
Apache                     ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Apache (v6)                ALLOW       Anywhere (v6)
```

为了放行 SSL 流量，执行如下命令：

```
sudo ufw allow 'apache full'
```

然后，在删除一条冗余的 profile：

```
sudo ufw delete allow 'apache'
```

4、使设置生效：

需要使模块 mod_ssl、mod_headers（配置文件中某些设置需要）生效。命令为：

```
sudo a2enmod ssl
sudo a2enmod headers
```

接下来使得 SSL 虚拟主机生效，使用命令：

```
sudo a2ensite default-ssl
```

ssl-params.conf 生效，命令为：

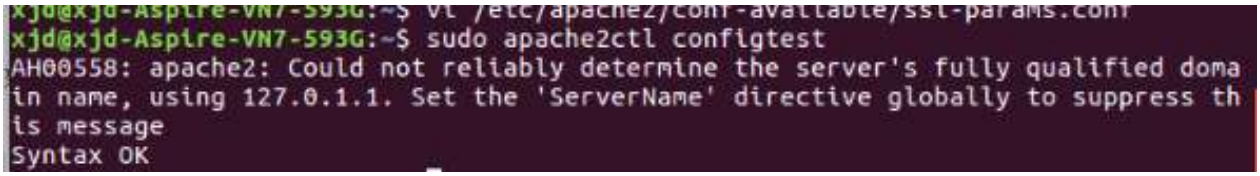
```
sudo a2enconf ssl-params
```

到这一步，服务器已经加载了必要的模块。为了进一步检查配置文件里面是否有语法错误，输入命令：

```
sudo apache2ctl configtest
```

如果都没有问题，示例输出为：

```
AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1.1. Set the
'ServerName' directive globally to suppress this message
Syntax OK
```



下一步，重启服务器：

```
sudo systemctl restart apache2
```

测试配置结果，从浏览器访问服务器：

```
https://ip
```

因为是自签证书，因此浏览器会弹出警告。

进一步测试 http 重定向 https 设置。

```
http://ip
```

如果测试通过，你确定只允许加密流量，则可以设置永久重定向。编辑文件：

```
/etc/apache2/sites-available/000-default.conf
```

结果如下：

```
<VirtualHost *:80>
        . . .

        Redirect permanent "/" "https://your_domain_or_IP/"

        . . .
</VirtualHost>
```

进一步需要测试配置语法是否正确：

```
sudo apache2ctl configtest
```

重启服务器：

```
sudo systemctl restart apache2
```

步骤三、TLS 流量分析

启动 wireshark，设置显示过滤器。因为只对 TLS 协议数据进行分析，所以这里仅显示 TLS 协议数据即可。设置显示过滤器为：ssl。开启抓包模式。如果浏览器启动后就会访问一些未指明的域名，可以等这一阶段过后，重新开启抓包模式。

启动 firefox 浏览器（或者 IE 浏览器、Chrome 浏览器），访问前面配置好的 apache 服务器或者域名 www.baidu.com（缺省就为 HTTPS）或者 www.amazon.com 等。为了防止 wireshark 占用太多资源，这个时候可以停止抓包功能。

为了仅分析 TLS1.2 的流量，需要对 Firefox 进行设置，以限制其仅支持 TLS1.2 协议。在 Firefox 浏览器的地址栏中输入 about:config 并按下回车键，打开 Firefox 的高级配置页面。在弹出的警告中，点击 "我了解风险" 按钮，以继续。在高级配置页面中，的搜索框中输入 security.tls.version.max 并按下回车，在搜索结果中可以看到 security.tls.version.max 配置项，把该配置项的值设置为 3（表示支持的最高 TLS 版本是 TLS1.2），单击 "确定" 以保存更改。

根据抓取的流量完成如下分析：

（1)根据 TLS 协议的原理，按照消息交换的顺序对截图说明每条消息的含义。

（2)根据 ClientHello 信息，计算 JA3 指纹。

从 ClientHello 提取 Version, Accepted ciphers, List of extensions, Elliptic Curves,Elliptic Curve Formats 字段的值。所有数值用十进制表示，用,分隔字段，用-分隔同一字段内的值。如果缺少某个字段，则留空，但是分隔符不能少。按如下方式计算：

JA3 = MD5(连接好的字符串)

（3)根据 ServerHello 信息，计算 JA3S 指纹。

从 ServerHello 提取 Version, Accepted cipher, List of extensions(type)字段的值。所有数值用十进制表示，用,分隔字段，用-分隔同一字段内的值。按如下方式计算：

JA3S = MD5(连接好的字符串)

1.5 【实验报告】

1） 说明实验过程。

2） 进行结果分析。

【注：实验中同时注意记录出错的情况，并说明是如何解决问题的】