哈尔滨工业大学(深圳)

《网络与系统安全》 实验报告

---

# 实验五

TLS 实验


学 院: 计算机科学与技术学院

姓 名: 梁鑫嵘

学 号: 200110619

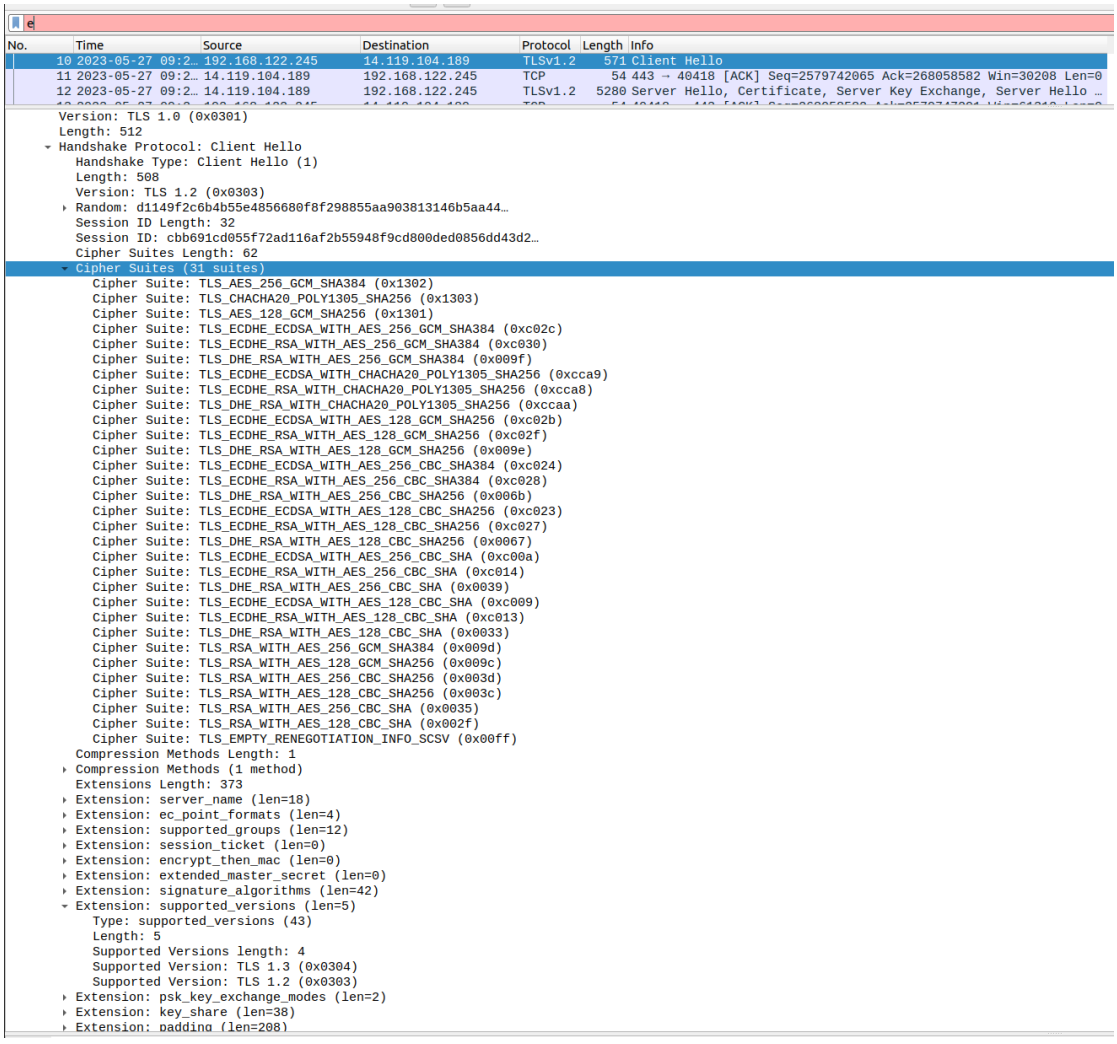
专 业: 计算机科学与技术学院


日 期: 2023 年 4 月

1.在客户端容器中执行如下命令 ./handshake.py www.baidu.com 根据执行结果回答下面三个问题。

（1）客户端和服务器端使用的加密算法有哪些，分别起什么作用?

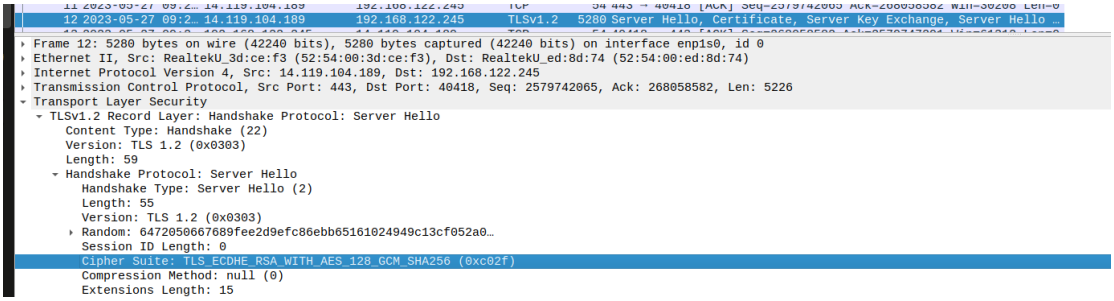
从 Client Hello 包得知，客户端支持的加密算法有：

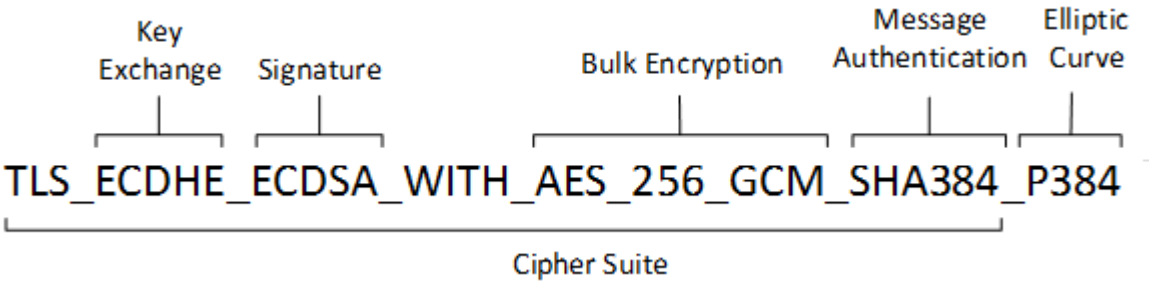



主要为 AES、ECDSA、CHACHA20、RSA 等。

从 Server Hello 包中得知，服务端使用的是

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256




（2）分析打印出来的服务器端证书

```
After making TCP connection. Press any key to continue ...
=== Cipher used: ('ECDHE-RSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
=== Server hostname: www.baidu.com
=== Server certificate:
{'OCSP': ('http://ocsp.globalsign.com/gsrsaovsslca2018',),
 'caIssuers': ('http://secure.globalsign.com/cacert/gsrsaovsslca2018.crt',),
 'crlDistributionPoints': ('http://crl.globalsign.com/gsrsaovsslca2018.crl',),
 'issuer': ((('countryName', 'BE'),),
            (('organizationName', 'GlobalSign nv-sa'),),
            (('commonName', 'GlobalSign RSA OV SSL CA 2018'),)),
 'notAfter': 'Aug  6 05:16:01 2023 GMT',
 'notBefore': 'Jul  5 05:16:02 2022 GMT',
 'serialNumber': '4417CE86EF82EC6921CC6F68',
 'subject': ((('countryName', 'CN'),),
             (('stateOrProvinceName', 'beijing'),),
             (('localityName', 'beijing'),),
             (('organizationalUnitName', 'service operation department'),),
             (('organizationName',
               'Beijing Baidu Netcom Science Technology Co., Ltd'),),
             (('commonName', 'baidu.com'),)),
 'subjectAltName': (('DNS', 'baidu.com'),
                    ('DNS', 'baifubao.com'),
                    ('DNS', 'www.baidu.cn'),
                    ('DNS', 'www.baidu.com.cn'),
                    ('DNS', 'mct.y.nuomi.com'),
                    ('DNS', 'apollo.auto'),
                    ('DNS', 'dwz.cn'),
                    ('DNS', '*.baidu.com'),
                    ('DNS', '*.baifubao.com'),
                    ('DNS', '*.baidustatic.com'),
                    ('DNS', '*.bdstatic.com'),
                    ('DNS', '*.bdimg.com'),
                    ('DNS', '*.hao123.com'),
                    ('DNS', '*.nuomi.com'),
                    ('DNS', '*.chuanke.com'),
                    ('DNS', '*.trustgo.com'),
                    ('DNS', '*.bce.baidu.com'),
                    ('DNS', '*.eyun.baidu.com'),
                    ('DNS', '*.map.baidu.com'),
                    ('DNS', '*.mbd.baidu.com'),
                    ('DNS', '*.fanyi.baidu.com'),
                    ('DNS', '*.baidubce.com'),
                    ('DNS', '*.mipcdn.com'),
                    ('DNS', '*.news.baidu.com'),
                    ('DNS', '*.baidupcs.com'),
                    ('DNS', '*.aipage.com'),
                    ('DNS', '*.aipage.cn'),
                    ('DNS', '*.bcehost.com'),
                    ('DNS', '*.safe.baidu.com'),
                    ('DNS', '*.im.baidu.com'),
                    ('DNS', '*.baiducontent.com'),
                    ('DNS', '*.dlnel.com'),
                    ('DNS', '*.dlnel.org'),
                    ('DNS', '*.dueros.baidu.com'),
```

证书分析：

1. 颁发者是 BE（比利时）的 GlobalSign

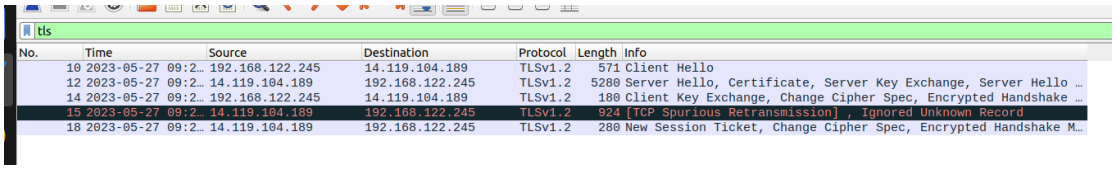
2. 持有者是 CN（中国）的 Beijing Baidu Netcom Science

Technology

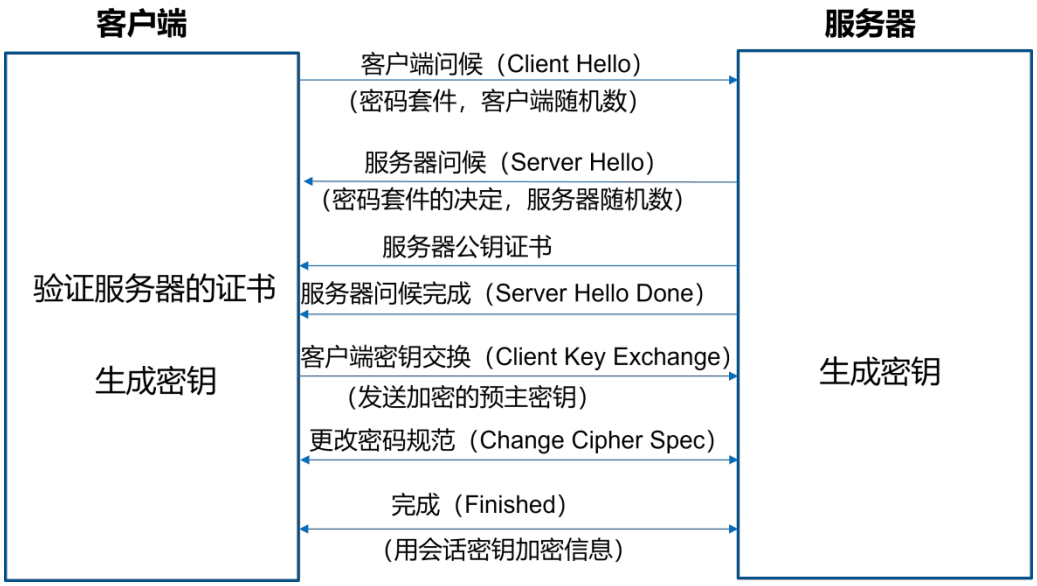
3. 此证书对下列域名都有效

（3）抓包分析 TLS 握手协议

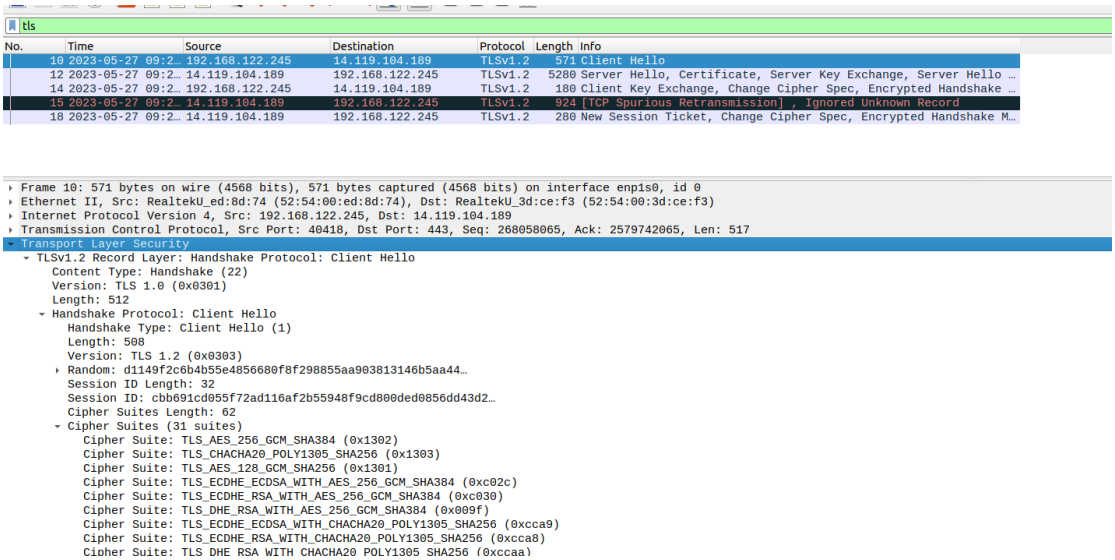
对脚本运行过程抓包：



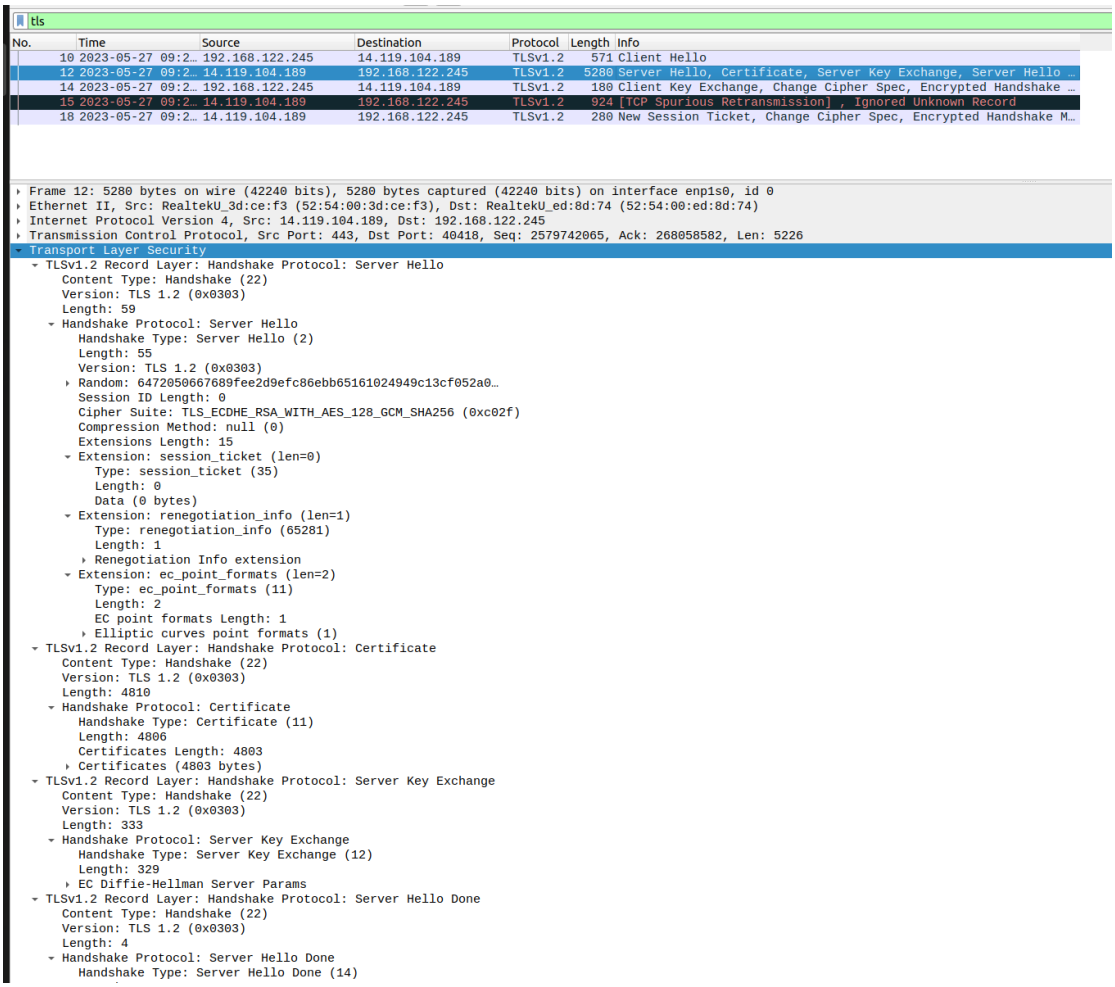

运行结束后有 4 个 TLSv1.2 导书的 TLS 握手过程：

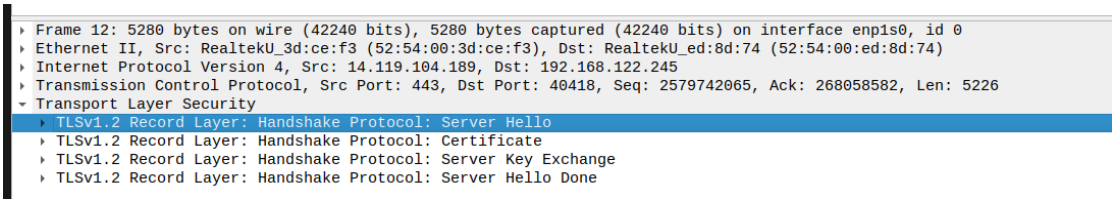


## Client Hello：



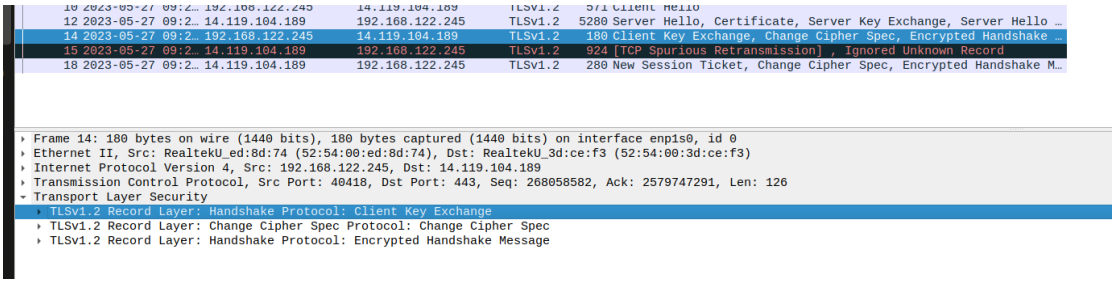

## Server Hello：

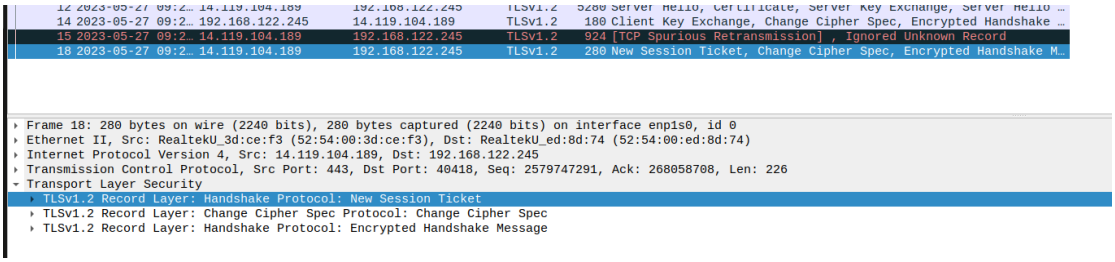



此时一次传输了 Server Hello、服务器证书、服务器密钥交换、服务器问候完成。

客户端密钥交换、更改密码规范、传递加密信息：



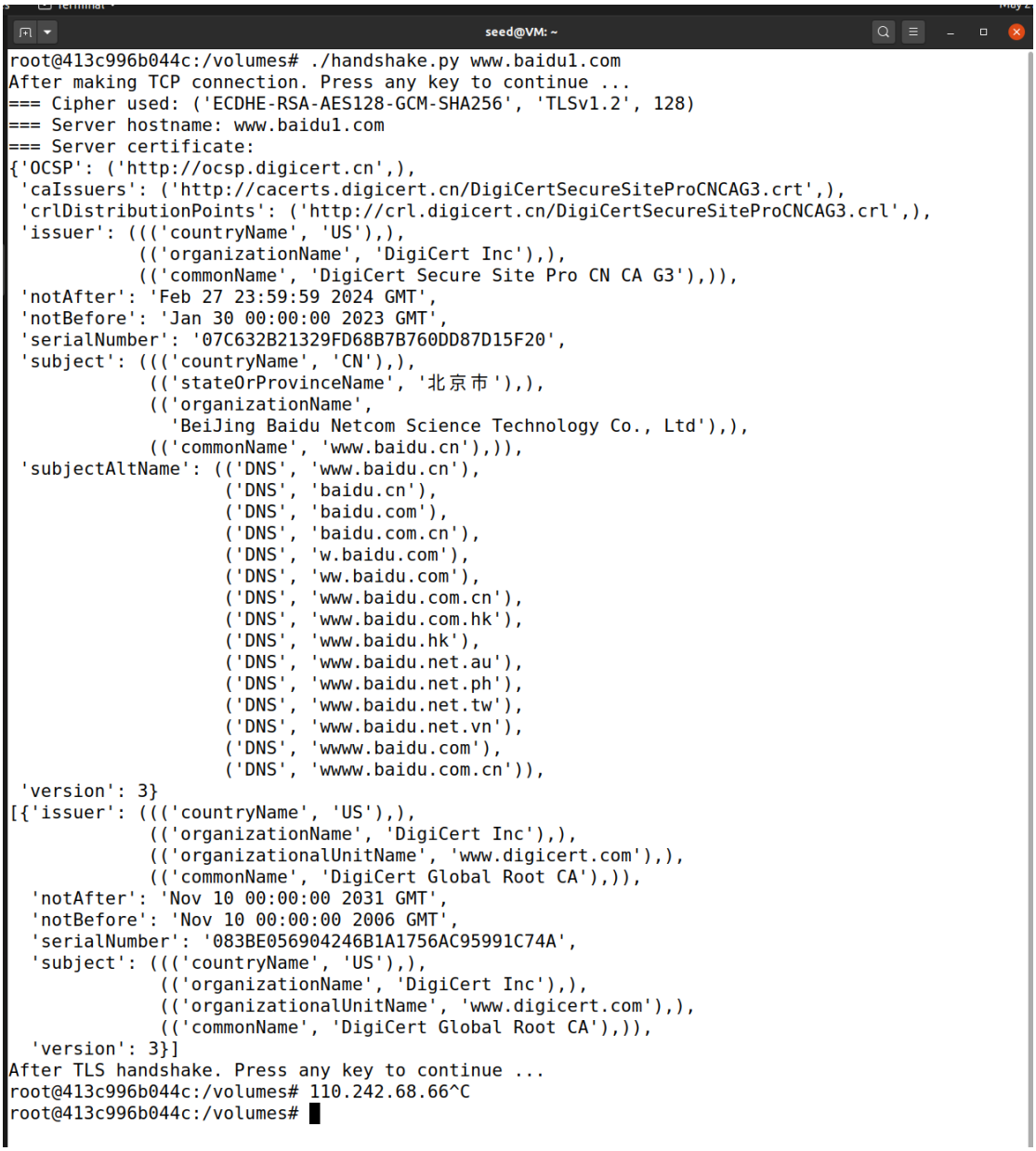

2. 更改证书文件路径，请同学们将 www.baidu.com 网站的测试过程截图保存（如果不将证书拷贝过来应该有报错信息，拷贝过来之后应该正常），也可选用其他网站做测试。

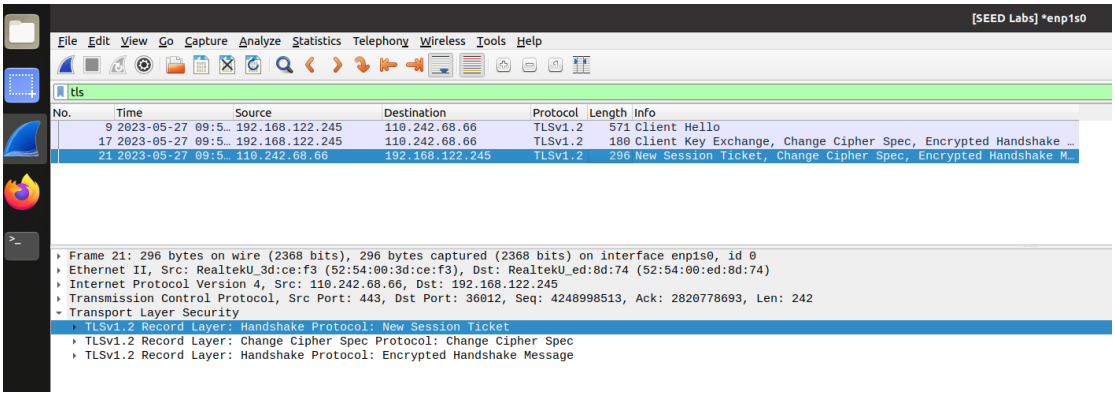
3.

```
root@413c996b044c:/volumes# ./handshake.py www.baidu.com                    [36/36]
After making TCP connection. Press any key to continue ...
=== Cipher used: ('ECDHE-RSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
=== Server hostname: www.baidu.com
=== Server certificate:
{'OCSP': ('http://ocsp.globalsign.com/gsrsaovsslca2018',),
 'caIssuers': ('http://secure.globalsign.com/cacert/gsrsaovsslca2018.crt',),
 'crlDistributionPoints': ('http://crl.globalsign.com/gsrsaovsslca2018.crl',),
 'issuer': ((('countryName', 'BE'),),
            (('organizationName', 'GlobalSign nv-sa'),),
            (('commonName', 'GlobalSign RSA OV SSL CA 2018'),)),
 'notAfter': 'Aug  6 05:16:01 2023 GMT',
 'notBefore': 'Jul  5 05:16:02 2022 GMT',
 'serialNumber': '4417CE86EF82EC6921CC6F68',
 'subject': ((('countryName', 'CN'),),
             (('stateOrProvinceName', 'beijing'),),
             (('localityName', 'beijing'),),
             (('organizationalUnitName', 'service operation department'),),
             (('organizationName',
               'Beijing Baidu Netcom Science Technology Co., Ltd'),),
             (('commonName', 'baidu.com'),)),
 'subjectAltName': (('DNS', 'baidu.com'),
                    ('DNS', 'baifubao.com'),
                    ('DNS', 'www.baidu.cn'),
                    ('DNS', 'www.baidu.com.cn'),
                    ('DNS', 'mct.y.nuomi.com'),
                    ('DNS', 'apollo.auto'),
                    ('DNS', 'dwz.cn'),
                    ('DNS', '*.baidu.com'),
                    ('DNS', '*.baifubao.com'),
                    ('DNS', '*.baidustatic.com'),
                    ('DNS', '*.bdstatic.com'),
                    ('DNS', '*.bdimg.com'),
                    ('DNS', '*.hao123.com'),
                    ('DNS', '*.nuomi.com'),
                    ('DNS', '*.chuanke.com'),
                    ('DNS', '*.trustgo.com'),
                    ('DNS', '*.bce.baidu.com'),
                    ('DNS', '*.eyun.baidu.com'),
                    ('DNS', '*.map.baidu.com'),
                    ('DNS', '*.mbd.baidu.com'),
                    ('DNS', '*.fanyi.baidu.com'),
                    ('DNS', '*.baidubce.com'),
                    ('DNS', '*.mipcdn.com'),
                    ('DNS', '*.news.baidu.com'),
                    ('DNS', '*.baidupcs.com'),
                    ('DNS', '*.aipage.com'),
                    ('DNS', '*.aipage.cn'),
                    ('DNS', '*.bcehost.com'),
                    ('DNS', '*.safe.baidu.com'),
                    ('DNS', '*.im.baidu.com'),
                    ('DNS', '*.baiducontent.com'),
                    ('DNS', '*.dlnel.com'),
                    ('DNS', '*.dlnel.org'),
[0] 0:docker-compose- 1:[tmux]*Z                                "VM" 12:18 27-May-22
```

3. 请同学们将修改 www.baidu.com 网站主机名的测试过程截图保存在报告里并分析执行的结果，也可选用其他网站做测试。

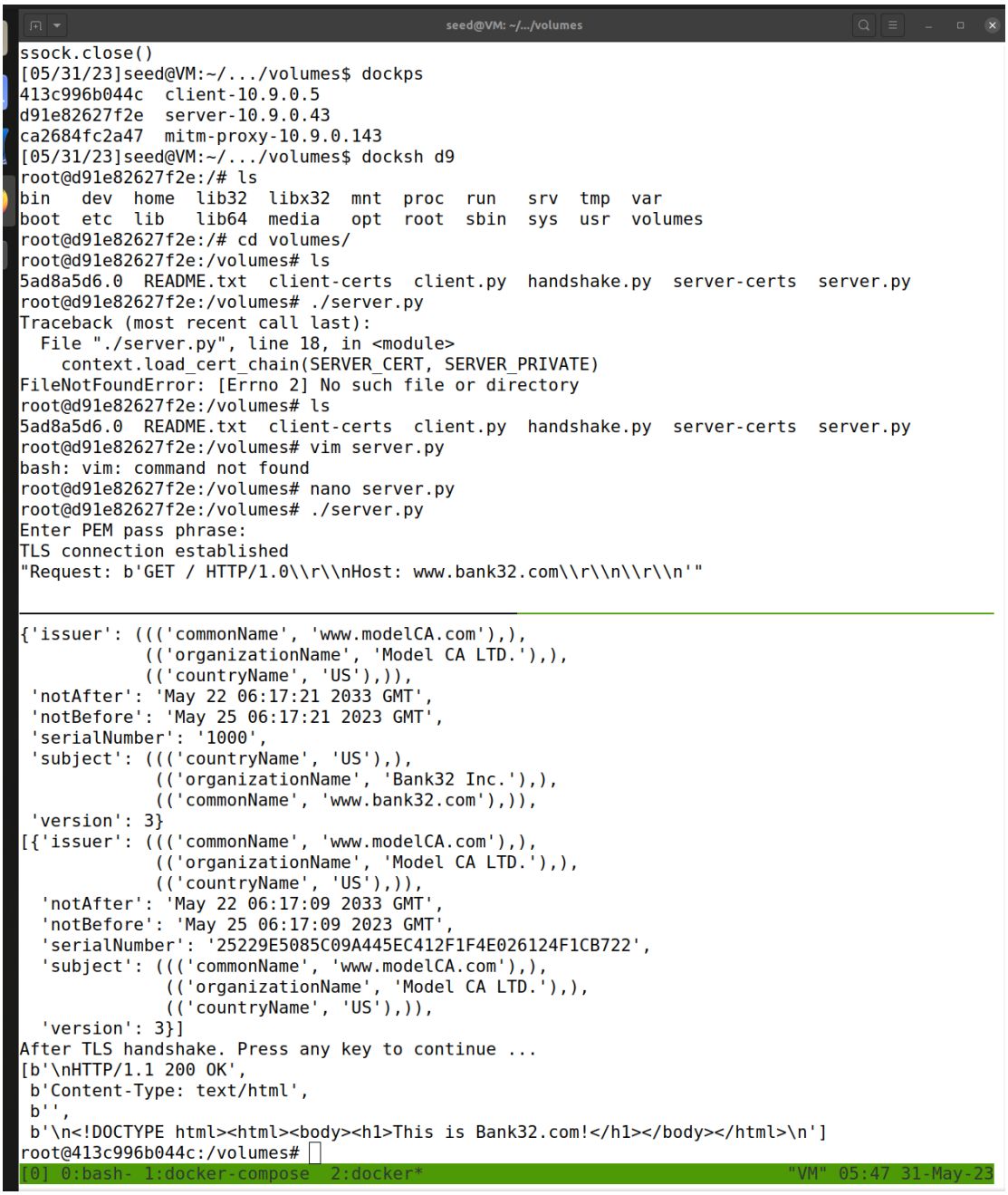
修改 hosts 后，由于没有校验域名正确性，所以整个流程可以顺利进行。


seed@VM: ~
root@413c996b044c:/volumes# ./handshake.py www.baidu1.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('ECDHE-RSA-AES128-GCM-SHA256', 'TLSv1.2', 128)
=== Server hostname: www.baidu1.com
=== Server certificate:
{'OCSP': ('http://ocsp.digicert.cn',),
'caIssuers': ('http://cacerts.digicert.cn/DigiCertSecureSiteProCNCAG3.crt',),
'crlDistributionPoints': ('http://crl.digicert.cn/DigiCertSecureSiteProCNCAG3.crl',),
'issuer': ((('countryName', 'US'),),
(('organizationName', 'DigiCert Inc'),),
(('commonName', 'DigiCert Secure Site Pro CN CA G3'),)),
'notAfter': 'Feb 27 23:59:59 2024 GMT',
'notBefore': 'Jan 30 00:00:00 2023 GMT',
'serialNumber': '07C632B21329FD68B7B760DD87D15F20',
'subject': ((('countryName', 'CN'),),
(('stateOrProvinceName', '北京市'),),
(('organizationName',
'BeiJing Baidu Netcom Science Technology Co., Ltd'),),
(('commonName', 'www.baidu.cn'),)),
'subjectAltName': (('DNS', 'www.baidu.cn'),
('DNS', 'baidu.cn'),
('DNS', 'baidu.com'),
('DNS', 'baidu.com.cn'),
('DNS', 'w.baidu.com'),
('DNS', 'ww.baidu.com'),
('DNS', 'www.baidu.com.cn'),
('DNS', 'www.baidu.com.hk'),
('DNS', 'www.baidu.hk'),
('DNS', 'www.baidu.net.au'),
('DNS', 'www.baidu.net.ph'),
('DNS', 'www.baidu.net.tw'),
('DNS', 'www.baidu.net.vn'),
('DNS', 'wwww.baidu.com'),
('DNS', 'wwww.baidu.com.cn')),
'version': 3}
[{'issuer': ((('countryName', 'US'),),
(('organizationName', 'DigiCert Inc'),),
(('organizationalUnitName', 'www.digicert.com'),),
(('commonName', 'DigiCert Global Root CA'),)),
'notAfter': 'Nov 10 00:00:00 2031 GMT',
'notBefore': 'Nov 10 00:00:00 2006 GMT',
'serialNumber': '083BE056904246B1A1756AC95991C74A',
'subject': ((('countryName', 'US'),),
(('organizationName', 'DigiCert Inc'),),
(('organizationalUnitName', 'www.digicert.com'),),
(('commonName', 'DigiCert Global Root CA'),)),
'version': 3}]
After TLS handshake. Press any key to continue ...
root@413c996b044c:/volumes# 110.242.68.66^C
root@413c996b044c:/volumes#


[SEED Labs] *enp1s0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls
No. Time Source Destination Protocol Length Info
9 2023-05-27 09:5… 192.168.122.245 110.242.68.66 TLSv1.2 571 Client Hello
17 2023-05-27 09:5… 192.168.122.245 110.242.68.66 TLSv1.2 180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake …
21 2023-05-27 09:5… 110.242.68.66 192.168.122.245 TLSv1.2 296 New Session Ticket, Change Cipher Spec, Encrypted Handshake M…
Frame 21: 296 bytes on wire (2368 bits), 296 bytes captured (2368 bits) on interface enp1s0, id 0
Ethernet II, Src: RealtekU_3d:ce:f3 (52:54:00:3d:ce:f3), Dst: RealtekU_ed:8d:74 (52:54:00:ed:8d:74)
Internet Protocol Version 4, Src: 110.242.68.66, Dst: 192.168.122.245
Transmission Control Protocol, Src Port: 443, Dst Port: 36012, Seq: 4248998513, Ack: 2820778693, Len: 242
Transport Layer Security
TLSv1.2 Record Layer: Handshake Protocol: New Session Ticket
TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message

4.请分析 TLS 客户端编程和 server.py 的代码，说明客户端和服务器程序的关键步骤。

从客户端访问服务端：




## 1. TCP 连接

```
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM, 0)
sock.bind(('10.9.0.43', 443))
sock.listen(5)

while True:
    newsock, fromaddr = sock.accept()
    try:
        ssock = context.wrap_socket(newsock, server_side=True)
```

```
# Create TCP connection
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((hostname, port))
input("After making TCP connection. Press any key to continue ...")
```

2. 数据传输

a) 服务端 TLS

```
try:
    ssock = context.wrap_socket(newsock, server_side=True)
    print("TLS connection established")
    data = ssock.recv(1024)                 # Read data over TLS
    pprint.pprint("Request: {}".format(data))
    ssock.sendall(html.encode('utf-8'))  # Send data over TLS

    ssock.shutdown(socket.SHUT_RDWR)     # Close the TLS connection
    ssock.close()
```

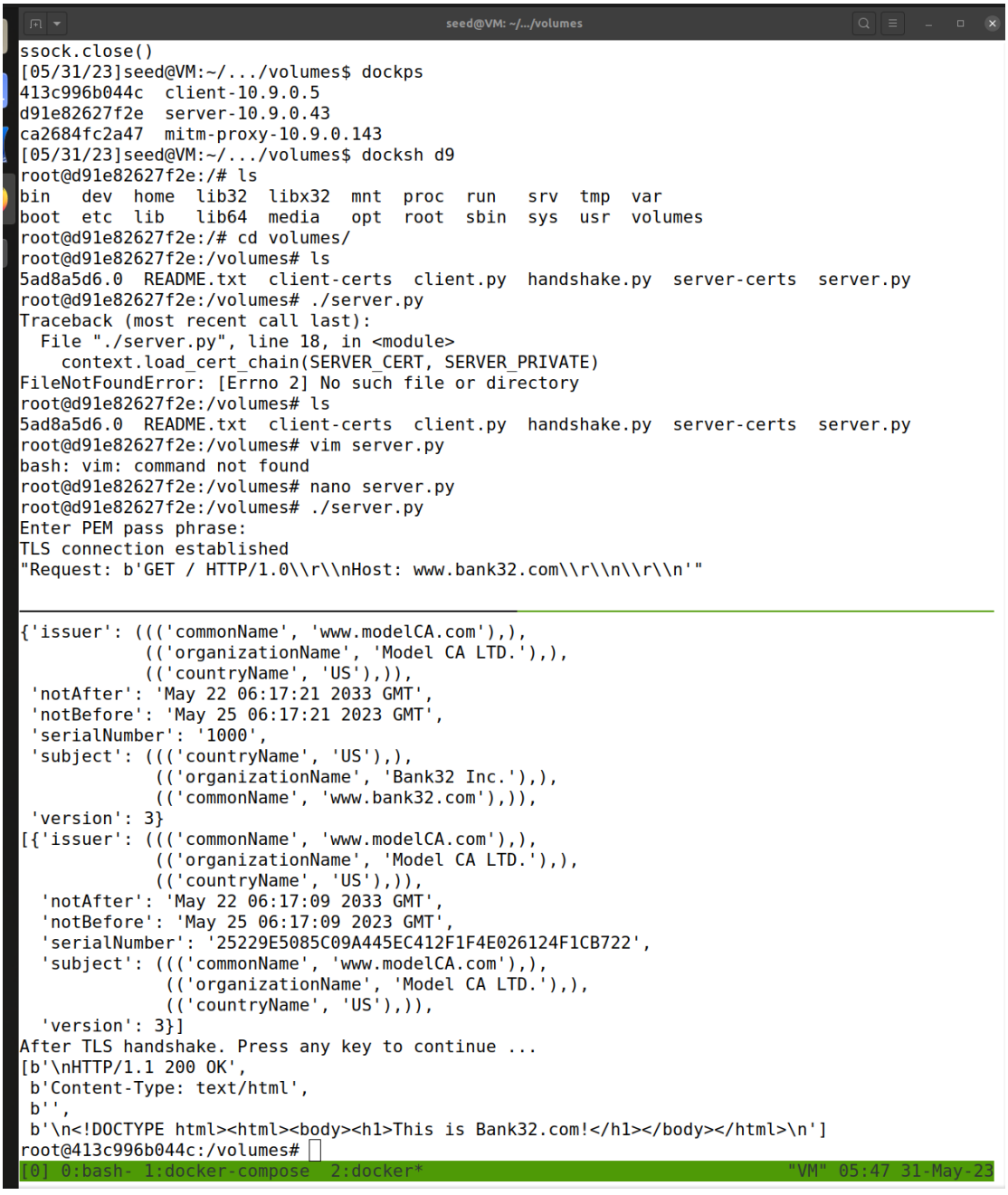
b) 客户端 TLS

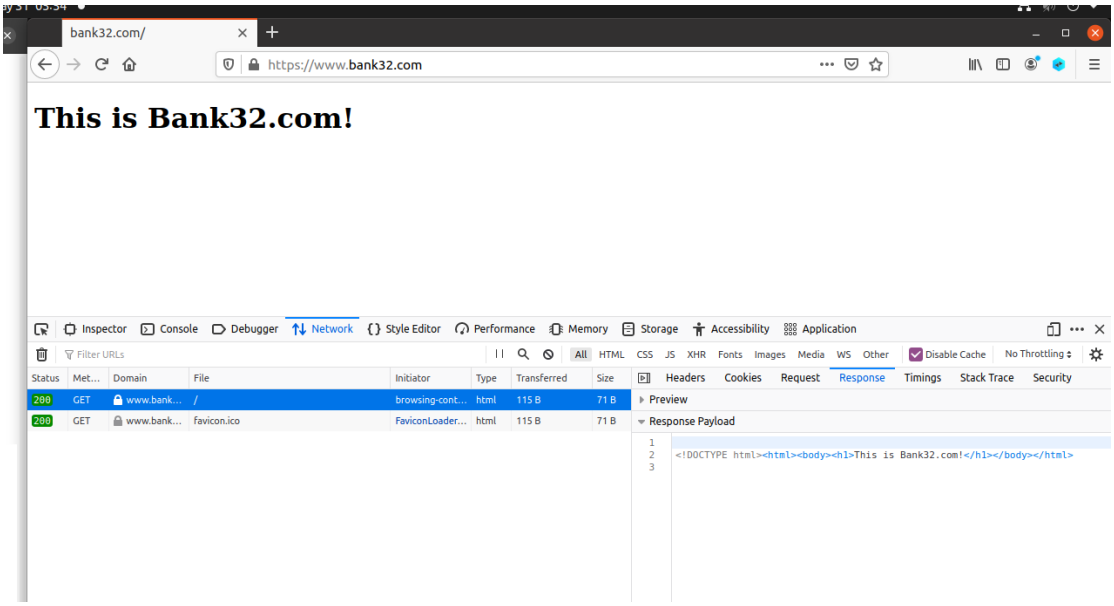
```
# Add the TLS
ssock = context.wrap_socket(sock, server_hostname=hostname,
                            do_handshake_on_connect=False)
ssock.do_handshake()   # Start the handshake
print("=== Cipher used: {}".format(ssock.cipher()))
print("=== Server hostname: {}".format(ssock.server_hostname))
print("=== Server certificate:")
pprint.pprint(ssock.getpeercert())
pprint.pprint(context.get_ca_certs())
print("After TLS handshake. Press any key to continue ...")
```

3. 客户端 HTTP 请求 Over TLS

```
# Send HTTP Request to Server
request = b"GET / HTTP/1.0\r\nHost: " + hostname.encode('utf-8') + b"\r\n\r\n"
ssock.sendall(request)
# Read HTTP Response from Server
response = ssock.recv(2048)
while response:
    pprint.pprint(response.split(b"\r\n"))
    response = ssock.recv(2048)
```

5.请分别用 client.py 和浏览器两种方式访问服务器，并记录你观察的结果（截图）






现象：

client.py 能够正确获取服务器的网页，

浏览器也能够正常请求服务器网页并显示。