1)
What is the signature algorithm? (5pt)

The signature is sha256WithRSAEncryption.

2)
When does the certificate expire? (5pt)

The certificate expires on Apr 19 02:19:20 2026 GMT.

3)
What is the serial number? (5pt)

```
Serial Number:
    7f:6c:1c:88:21:79:6a:df:2a:a3:71:9f:28:a8:ae:f1:99:2a:14:c5
```

4)
What is the signature value? (5pt)

```
Signature Value:
    25:58:2f:4a:e8:9e:3f:44:d1:11:39:e2:a5:7e:56:54:8c:32:
    eb:14:58:fc:69:e9:83:c3:fc:17:05:b2:86:e0:56:90:2d:96:
    68:2e:35:10:77:45:15:9c:27:5c:3f:8a:41:de:23:9c:01:d3:
    00:68:bb:fd:04:9f:85:a3:ac:92:3f:cc:8c:c4:80:33:26:1d:
    da:f7:ce:ed:33:b3:a9:73:76:1b:c8:92:e3:03:03:dd:bc:5c:
    5e:51:60:4d:e1:45:33:51:84:b0:3b:a6:b8:16:e5:56:42:79:
    52:94:8a:d6:21:ab:ef:66:2e:7b:af:9e:e5:df:90:5a:68:5a:
    86:3a:b2:52:3c:f6:9b:25:13:d6:c2:dc:37:5a:de:8d:b4:0f:
    1d:af:73:6a:dd:5f:5c:9e:db:a0:db:a3:43:76:89:63:e0:92:
    c3:2c:c0:7a:e9:31:b0:fd:65:01:f1:83:44:e4:e9:84:ef:c0:
    24:79:e9:9d:16:35:3b:bb:da:2e:75:a4:15:48:f3:3d:3e:79:
    82:1d:26:56:8b:07:8f:5e:b5:a9:f1:07:9e:b2:ea:35:92:45:
    e4:0a:52:1c:43:fa:aa:73:6f:1e:74:b4:e4:c6:29:e8:f6:9a:
    2c:1f:e7:7b:fb:07:1d:da:82:fe:7b:07:02:4d:5a:e2:51:77:
    91:e0:da:96
```

5)
Compared to the previous certificate which parts of the server certificate are different

than the previous certificate created in Task 1.1? (5pt)

The parts that are different were subject, issuer, basicConstraints, and subjectAltName.

6)
What are the new values for the different parts? (10pt)

In task 1.1 the common name was RootCA and the organizational unit was security whereas in task 1.2 CN and OU were set to localhost and Web respectively. Task 1.1 Had a self signed certificate whereas task 1.2 has a different issuer(RootCA). SubjectAltName is present in task 1.2 and has DNS: localhost. Even the basic constraints were different. CA:FALSE in task 1.2.

7) Take a screenshot of the terminal while listening on port 4443 (10pt)

```
cpre3310@cpre3310:~/homework/Lab11$ openssl s_server -accept 4443 -cert server.crt -k
ey server.key
Using default temp DH parameters
ACCEPT
```

8) Take a screenshot of the TLS handshake and verification result (10pt)

```
cpre3310@cpre3310:~/homework/Lab11$ openssl s_client -connect localhost:4443 -CAfile
rootCA.pem
CONNECTED(00000003)
Can't use SSL_get_servername
depth=1 C = US, ST = State, L = City, O = MyOrg, OU = Security, CN = RootCA
verify return:1
depth=0 C = US, ST = State, L = City, O = MyOrg, OU = Web, CN = localhost
verify return:1
---
Certificate chain
 0 s:C = US, ST = State, L = City, O = MyOrg, OU = Web, CN = localhost
   i:C = US, ST = State, L = City, O = MyOrg, OU = Security, CN = RootCA
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Apr 19 03:13:21 2025 GMT; NotAfter: Apr 19 03:13:21 2026 GMT
---
```

```
Server certificate
-----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIUL/6Ci/0lUf0o37YrT623J3lYXXkwDQYJKoZIhvcNAQEL
BQAwYDELMAkGA1UEBhMCVVMxDjAMBgNVBAgMBVN0YXRlMQ0wCwYDVQQHDARDaXR5
MQ4wDAYDVQQKDAVNeU9yZzERMA8GA1UECwwIU2VjdXJpdHkxDzANBgNVBAMMBlJv
b3RDQTAeFw0yNTA0MTkwMzEzMjFaFw0yNjA0MTkwMzEzMjFaMF4xCzAJBgNVBAYT
AlVTMQ4wDAYDVQQIDAVTdGF0ZTENMAsGA1UEBwwEQ2l0eTEOMAwGA1UECgwFTXlP
cmcxDDAKBgNVBAsMA1dlYjESMBAGA1UEAwwJbG9jYWxob3N0MIIBIjANBgkqhkiG
9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2Imr+Z9tC3JDW9WUXP7LGhZDXV5OMmsCWGan
vDKLFgCPfH7kfrfjqoy/gDkjpRPKxDEX3l93JlIzMRgUvHWFqfWG1T8p2LO4YgT8
TKv0GGASbrAIZQFYj/WSCmKEOdr9qGfx2rutWpcX4XrnQfAoWIAYzX6EitQufdNr
HVxs5AmGTKF8fYP4eh0YqyTsYRcQowplA5UuQtHYTz2M71lHCyP2uj52Xw24UHDC
esWynf8kkjSvKGdo1a+Ue4DupZZt3H4OmU4jOmMNEitvYZdAUZzMKV2CNUTp5Yz1
rYNxekFOC6YxZhccF0/PxkoXIoTxQAarL7DNa5fm0JtEfSYywwIDAQABo2MwYTAJ
BgNVHRMEAjAAMBQGA1UdEQQNMAuCCWxvY2FsaG9zdDAdBgNVHQ4EFgQUNA2myRhC
nGxJY2SuoeaTjFgDNNQwHwYDVR0jBBgwFoAUjGpV+lcBYVldHO9pojiytW6naHkw
DQYJKoZIhvcNAQELBQADggEBALlb2Bxx1tLZ4x4dqUGHiJCP3jZmO5Cj7jwPhjAD
6uBzpj5MlabBdJjGxHWMKH8JwjKEv5H8Qn+PvVZZmhGJOnThnWk7VJUWTncqkH4H
+nv7s0z/BDcDbx+z9vIsSH5Di4u+H17DafcIsWzFodC1Putwe7fzRN7Or31Kpf/H
beKEVHH80CNReSoW8/OhS4PBPTRjk/IN/Q/v8vOmdpFnhxE+X6AKn52iktJuSUX5
y2mz5tjxyTqqDbhBrvrv2kdL89JX7EC+4You/vreAMBu6VijC0g2ldSV253AprPm
7PANVe8vndzYSZw6B/W/pJ1YrmKQ8B8NOB0d4hqG2GbNAIc=
-----END CERTIFICATE-----
subject=C = US, ST = State, L = City, O = MyOrg, OU = Web, CN = localhost
issuer=C = US, ST = State, L = City, O = MyOrg, OU = Security, CN = RootCA
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1503 bytes and written 373 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: FBD76724FB75CF6A6C2CB8BD17E102D8569537BF9B11D3341A1DA0190E2299D3
    Session-ID-ctx:
    Resumption PSK: 655A7F0DD02C06405762B291F2BE628F1594F4746271295132163A1CFE2C44251
43E543BEAA59BFB4FA81B8C28F47AC9
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
```

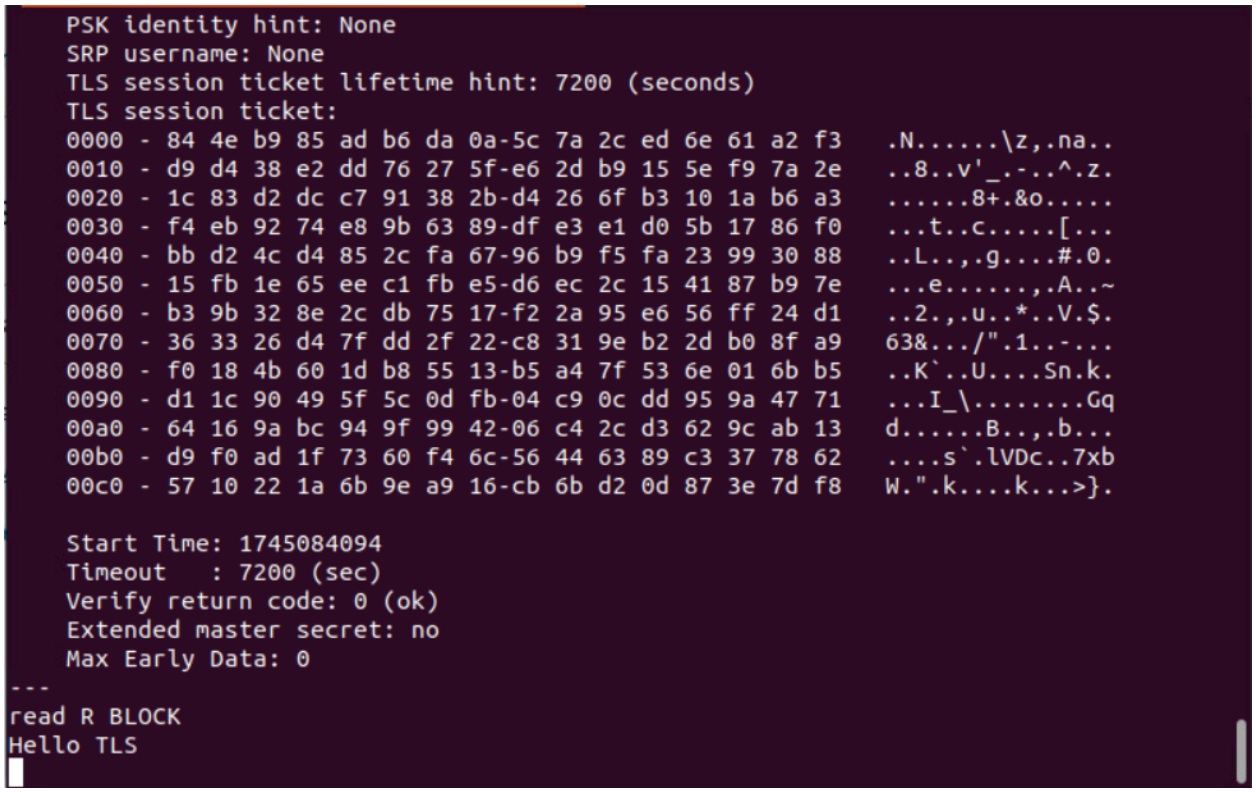
```
    0000 - 84 4e b9 85 ad b6 da 0a-5c 7a 2c ed 6e 61 a2 f3   .N......\z,.na..
    0010 - f8 b4 dc 05 58 cd 8e c7-0b d9 24 32 00 b7 fb db   ....X.....$2....
    0020 - 55 0c 6a 7f ce e8 8f b2-5f aa 38 86 7f df 82 dc   U.j....._.8.....
    0030 - f1 58 cf 73 6c bb f6 0d-1d df ab ae 7d a7 b4 3b   .X.sl.......}..;
    0040 - 84 7a 87 9c ca f8 96 09-3e 47 5c 61 01 99 80 17   .z......>G\a....
    0050 - c1 3d e9 71 e0 14 af 86-d9 9d 09 59 05 0b 9f b7   .=.q.......Y....
    0060 - 98 d3 66 78 fb 68 fc dd-1a 97 ec 9b 5d fa e0 98   ..fx.h......]...
    0070 - cf 0d 5f 78 cf 20 66 4e-75 89 57 b3 83 a7 84 0b   .._x. fNu.W.....
    0080 - 56 a7 21 14 2d a8 f6 af-a7 d3 83 91 51 8e a1 d1   V.!.-.......Q...
    0090 - ee 27 df 39 a8 fa b0 5d-5d 03 e7 8b 17 e7 30 57   .'.9...]].....0W
    00a0 - f9 ec 42 36 c1 10 01 e3-62 ff 13 8c 83 50 37 bc   ..B6....b....P7.
    00b0 - 30 d6 c5 09 2d a2 d3 96-d7 03 c0 fd 6f 06 56 72   0...-.......o.Vr
    00c0 - 03 5c c2 6a a5 b2 1f b9-23 89 31 50 a3 8a 6f bf   .\.j....#.1P..o.

    Start Time: 1745084094
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

```
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 6C7E22B25BBC5BE351324B77DF06E7C4DA0772F0D8233F5C2608892280A2D6A6
    Session-ID-ctx:
    Resumption PSK: B84C38CE4DAE388C611ED052FEE6315043DAEBD41C1A0EFBBF0190154DB91AA90
43953CC8CB9724ECE99918D8AF015BD
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 7200 (seconds)
    TLS session ticket:
    0000 - 84 4e b9 85 ad b6 da 0a-5c 7a 2c ed 6e 61 a2 f3   .N......\z,.na..
    0010 - d9 d4 38 e2 dd 76 27 5f-e6 2d b9 15 5e f9 7a 2e   ..8..v'_.-..^.z.
    0020 - 1c 83 d2 dc c7 91 38 2b-d4 26 6f b3 10 1a b6 a3   ......8+.&o.....
    0030 - f4 eb 92 74 e8 9b 63 89-df e3 e1 d0 5b 17 86 f0   ...t..c.....[...
    0040 - bb d2 4c d4 85 2c fa 67-96 b9 f5 fa 23 99 30 88   ..L..,.g....#.0.
    0050 - 15 fb 1e 65 ee c1 fb e5-d6 ec 2c 15 41 87 b9 7e   ...e......,.A..~
    0060 - b3 9b 32 8e 2c db 75 17-f2 2a 95 e6 56 ff 24 d1   ..2.,.u..*..V.$.
    0070 - 36 33 26 d4 7f dd 2f 22-c8 31 9e b2 2d b0 8f a9   63&.../".1..-...
    0080 - f0 18 4b 60 1d b8 55 13-b5 a4 7f 53 6e 01 6b b5   ..K`..U....Sn.k.
    0090 - d1 1c 90 49 5f 5c 0d fb-04 c9 0c dd 95 9a 47 71   ...I_\........Gq
    00a0 - 64 16 9a bc 94 9f 99 42-06 c4 2c d3 62 9c ab 13   d......B..,.b...
    00b0 - d9 f0 ad 1f 73 60 f4 6c-56 44 63 89 c3 37 78 62   ....s`.lVDc..7xb
    00c0 - 57 10 22 1a 6b 9e a9 16-cb 6b d2 0d 87 3e 7d f8   W.".k....k...>}.
```

```
    Start Time: 1745084094
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

9) Take a screenshot of the message from both client and server terminals (10pt)

Client:



Server:

```
BDC4TDjOTa44jGEe0FL+5jFQQ9rr1BwaDvu/AZAVTbkaqQQ5U8yMuXJOzpmRjYrw
Fb2hBgIEaAPevqIEAgIcIKQGBAQBAAAArgcCBQDWQnjWswMCAR0=
-----END SSL SESSION PARAMETERS-----
Shared ciphers:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SH
A256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA
384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-CHACHA20-POLY13
05:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA25
6:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:ECDHE-ECDSA
-AES128-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:E
CDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:DH
E-RSA-AES128-SHA:AES256-GCM-SHA384:AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:AES2
56-SHA:AES128-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SH
A256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+S
HA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224:DSA+SHA224:DSA+SHA256:DSA+SHA384:
DSA+SHA512
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA
-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA51
2:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224
Supported groups: x25519:secp256r1:x448:secp521r1:secp384r1:ffdhe2048:ffdhe3072:ffdhe
4096:ffdhe6144:ffdhe8192
Shared groups: x25519:secp256r1:x448:secp521r1:secp384r1:ffdhe2048:ffdhe3072:ffdhe409
6:ffdhe6144:ffdhe8192
CIPHER is TLS_AES_256_GCM_SHA384
Secure Renegotiation IS supported
Hello TLS
```

10) Do you see any sign that the certificate has expired on the server side? (5pt)

No, there is no sign on the server side but the client side does have a sign.
Client side:

```
    Verify return code: 10 (certificate has expired)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
Hello TLS
```

11) Why does the server still accept messages from clients even though the certificate has

expired? (10pt)

This happens because openssl's server side does not validate its own certificate. In TLS connections the client is responsible for verifying whether a certificate is valid, trusted, or expired. Therefore the server still accepts messages from its clients even though the certificate has expired.

12) What is the cipher used for this website's TLS certificate? (5pt)

The cipher used is TLS_AES_256_GCM_SHA384.

13) What is the meaning of "Peer Signature Type"? (10pt)

The Peer Signature Type indicates the algorithm that the server (peer) uses to digitally sign its certificate to prove its identity to the client during the TLS handshake. In this certificate ECDSA was our signature type. ECDSA - Elliptic Curve Digital Signature Algorithm is used to sign this certificate.

14) Please provide the screenshot of the TLS message. You can submit multiple

screenshots to cover the entire TLS handshake. (5pt)

```
bash-4.4$ openssl s_client -connect www.google.com:443
CONNECTED(00000003)
depth=2 C = US, O = Google Trust Services LLC, CN = GTS Root R4
verify return:1
depth=1 C = US, O = Google Trust Services, CN = WE2
verify return:1
depth=0 CN = www.google.com
verify return:1
---
Certificate chain
 0 s:CN = www.google.com
   i:C = US, O = Google Trust Services, CN = WE2
 1 s:C = US, O = Google Trust Services, CN = WE2
   i:C = US, O = Google Trust Services LLC, CN = GTS Root R4
 2 s:C = US, O = Google Trust Services LLC, CN = GTS Root R4
   i:C = BE, O = GlobalSign nv-sa, OU = Root CA, CN = GlobalSign Root CA
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDlTCCAzygAwIBAgIQAnGD1KgleQcKPY9gnifN5TAKBggqhkjOPQQDAjA7MQsw
CQYDVQQGEwJVUzEeMBwGA1UEChMVR29vZ2xlIFRydXN0IFNlcnZpY2VzMQwwCgYD
VQQDEwNXRTIwHhcNMjUwMzMxMDg1NjI3WhcNMjUwNjIzMDg1NjI2WjAZMRcwFQYD
VQQDEw53d3cuZ29vZ2xlLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABBHj
B5fkcQxfYTjDVmvM4Jpr4RhjL+mH4yyk8lTvodX9BsFwTMwbaZ3AH7rPf9Pv6s3v
M9CBGWcwDkVZbDXS4NSjggJCMIICPjAOBgNVHQ8BAf8EBAMCB4AwEwYDVR0lBAww
CgYIKwYBBQUHAwEwDAYDVR0TAQH/BAIwADAdBgNVHQ4EFgQU3KP2XkzycUZRpFCY
kYv8IO8pcEUwHwYDVR0jBBgwFoAUdb7Ed66J9kQ3fc+xaB8dGuvcNFkwWAYIKwYB
BQUHAQEETDBKMCEGCCsGAQUFBzABhhVodHRwOi8vby5wa2kuZ29vZy93ZTIwJQYI
KwYBBQUHMAKGGWh0dHA6Ly9pLnBraS5nb29nL3dlMi5jcnQwGQYDVR0RBBIwEIIO
d3d3Lmdvb2dsZS5jb20wEwYDVR0gBAwwCjAIBgZngQwBAgEwNgYDVR0fBC8wLTAr
oCmgJ4YlaHR0cDovL2MucGtpLmdvb2cvd2UyL3h1enQzUFU5Rl93LmNybDCCAQUG
CisGAQQB1nkCBAIEgfYEgfMA8QB2AM8RVu7VLnyv84db2Wkum+kacWdKsBfsrAHS
W3fOzDsIAAABleuhkB4AAAQDAEcwRQIhAMf3jwButHnFnHo1aUx9e+EbNsgP2WzC
YyhM3o9H13J0AiAgLZv1kFt0po07tpllOvk/LAzx8RtO9l3IDHxs0q7AXQB3AKLj
```

```
YyhM3o9H13J0AiAgLZv1kFt0po07tpllOvk/LAzx8RtO9l3IDHxs0q7AXQB3AKLj
CuRF772tm3447Udnd1PXgluElNcrXhssxLlQpEfnAAABleuhk+MAAAQDAEgwRgIh
AIu72/WhD+8tyuBXYyJ7sqUhTXuurs4MLJIqDcT2Y6USAiEA0Dmz78Ap+gPbnUhJ
+UifxR8jQ2tBX7J27wfH6sbfl3YwCgYIKoZIzj0EAwIDRwAwRAIgZOShqs9njXez
5Wen/buqZWKZsXw57BPidSojHUJ5IhoCICRd5uCEGApzR5sG506AnVswrPKdMtiX
H7RqdOGixbXu
-----END CERTIFICATE-----
subject=CN = www.google.com

issuer=C = US, O = Google Trust Services, CN = WE2

---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: ECDSA
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 2804 bytes and written 392 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 256 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
```