

```
parallels@ubuntu-linux-20-04-desktop:~$ openssl s_client -connect twitter.com:443 -CApath /  
CONNECTED(00000003)  
depth=1 C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1  
verify error:num=20:unable to get local issuer certificate  
verify return:1  
depth=0 C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com  
verify return:1  
---  
Certificate chain  
 0 s:C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com  
  i:C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1  
 1 s:C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1  
  i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA  
---  
Server certificate  
-----BEGIN CERTIFICATE-----  
MIIFBTCCBPsgAwIBAgIQCxcca7CYoYZhkbnWP5XwSTAKBggqhkJOPQOAZBWmQsw  
CQYDVQQGEWJUUZEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMTAwLGYvDQVODQEydzEwdp  
Q2VydcBUFTMGSHlicmlkIEVDQyBTSEEzODQGMjAyMCBBQDEWEHhcNjMzMmAzMDAw  
MDAwMHcNMjYxOTUyZW5kaWQwR2FuIExyZW5jaXNjbzEWMBQGA1UEChMMNVhdpcHRlc  
iCWGSwSJAUEUMBIGA1UEAxMLdHdpdHRlcisjb20wNTATBgqchkhjOPQIBBggqhkJOp  
KREApTlttEG7OM8cus+37w3owHQYDVR0RBGFAGdg9XjselYGygjXob4j9XFIXn  
MCEGA1UdEQQGMBGCC3R3aXR0ZXIuY29tg93d3udHdpdHRlcisjb20wOGYyDVR0P  
AQH/BAAQAgeAMBGA1UdJQMwMBGGCSGAQUFBwwMBBgggrBgEFBQCDAjCBWWYDVR0f  
BFITMICQMEagRKBCbkBodHRwOi8vY3JsMySkawdpY2VydcSjyb20vRGlnaUNlcnRU  
TGFIeWJyaWRFRQONTSEEEzODQyMDIwQExLTEuY3JsMEagRKBCbkBodHRwOi8vY3Js  
NCScIAwlpY2VydcSjyb20vRGlnaUNlcnRUFGFTEuY3JsMEagRKBCbkBodHRwOi8vY3Js  
LTUyY3JsMD4GA1UdIAQ3MDUwMwYGVZ4EMAQICMcKcwJwYIKwYBBQUHAQEHWG2h0dHA6  
Ly93d3cuZGlnaWNlcnQuY29tLONOUzCBhQYIKwYBBQUHAQEETB3MCQGCCSGAUFB  
BzABhhndHRwOi8vb2Nzc5kawdpY2VydcSjyb20wTWYIKwYBBQUHMKGACQ2h0dHA6  
Ly9jYWllcnRzLnRpZ2ljZXJ0LmNvbS9EawdpQ2VydcFRMU0h5YnJpZEVDQ1NIQTm4  
IDlwMiBDQTEtMS5jcncwCQYDVR0TBAlwADCCAxBGCISGAQQB1nkCBAIEggFvBIIB  
awFPAHYAdv+IPqw+2SVRWmHM9Ye6NLskzbSP3GHCCP/mzAOxaOnQAAGFDxOBqAA  
BAMARZBFAiEAAnxmInk6eUD0CDPfAFmbhbIMPEyFYnnLfmo2wnDa/UClHPHE9A4  
IWpNiOs/CgwmmGe3gKQj7pqh7sstRMtljXx9fAhcASLDja9qmRZQP5woC+p0w6xxS  
ActNs3Yb2bu/jqnYhMAAAGFDXNQMQAABAAMASDBGAIEAmr2175fLCpjUTLO97tg8  
ZWmsyc5AucJM+UHAPThILM0CIQCUY5Gnk15yQ5SAbd9wogcv2PxDP14GRwxDMpgGnJ  
LLaAQ82AdDtD3U+LBmAtoswWwb+QDTn2E/D9Me9AA0tcM/h+tQXAAAbHXVzaI8A  
AAQDAECwRIQIG5MxpEzc0ynB3TR7S2J0BSxCjGXJOUSjsHsf/Y6LKACIQCLZbm+  
tJGWwxfQZ6viewbvrxc+ME0UFy9HJEgEwfntAKBggqhkJOPQOAWNNabDKAJBN  
i7q7j+zuqx/cTZtpykMc6vU5schbfICEZujVEIJ9DW0Q6ieIj30PB8U8eLCxzXPf4C  
HGj2ywKCTRG1THX7y883n04m4ftmqUBORBrLPGUJDJFMUOd+dJz6MPovZzhAOS6T  
PA==  
-----END CERTIFICATE-----  
subject=C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com  
  
issuer=C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1  
  
---  
No client certificate CA names sent  
Peer signing digest: SHA256  
Peer signature type: ECDSA  
Server Temp Key: X25519, 253 bits  
---  
SSL handshake has read 2754 bytes and written 383 bytes  
Verification error: unable to get local issuer certificate  
---  
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: 20 (unable to get local issuer certificate)  
---
```

```

parallel@ubuntu-linux-20-04-desktop:~$ openssl s_client -connect twitter.com:443 -CAfile /etc/ssls/certs/ca-certificates.crt
CONNECTED(00000003)
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
verify return:1
depth=1 C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1
verify return:1
depth=0 C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com
verify return:1
---
Certificate chain
 0 s:C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com
  i:C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1
 1 s:C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1
  i:C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFBtCCBBGsgAwIBAgIQcXxca7CYoYZhknbnWP5XwSTAKBgqhkJOPQQDAzBMWQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMRGlncmlnaUNlcnQgSjNjMTAwLGYyVQQDEydEawDp
Q2VyZCBUTFMgSHlicnlkIEVDQyBTSEEEZODQqMjAyMCBDQTEwHhcNMjMwMDAw
MDAwMmhmcmJQwMTAzMjM1OTU5WjBoMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2F5
aWZhcnVsZW50YTEwBQQA1UEBAxMNjU2FuIEZyYW5jaXNjb2EWMBOGA1UEChMNVHdpdHRl
ciwgSW5jLjEUMBIGA1UEAxMLDHdpdHRlcisjb20wMTATBgqhkJOPQIBggqhkJO
PQMBBwNCAATBaScGRXLMZlweibpgniMdyeJAPD5S8rrkHQAGF3VZgYqeSA2LSkc
he88OAaj1GR/Iw42ksCxMSeinFafMsvYbo4IDkDCCA4wwHwYDVR0jJBgwFoAUCrwiI
KREmpTltteg7OM8cus+37w3owHQYDVR00BBYEAFAGdUg9XjselYGygjXob4/9XFIXn
McCGA1UdeQQgMB6CC3R3axR0ZXIUy29tg93d3cudHdpdHRlcisjb20wDgYDVR0P
AQH/BAAQAgeAMB0GA1UdJQQMNBGGCCsGAQUFBwMBBggrBgEFBQCDAjCBmwYDVR0P
BIGTMICQMEagRKBCbkBodHRwOi8vY3JsMySkaldpY2VyZC5jb20vRGlnaUNlcnRU
TFNIewJyaWRfQ0NTSEEEZODQyMDIwQExLTEuY3JsMEagRKBCbkBodHRwOi8vY3Js
NC5kaWdpY2VyZC5jb20vRGlnaUNlcnRUTFNiellJyaWRfQ0NTSEEEZODQyMDIwQEx
LTEuY3JsMD0GA1UdIAQ3MDUwMmYyZG4EMAQICCKwJwYIKwYBBQUHAQEENG2h0dHA6
Ly93d3cuZGUzLnawNlcnQuY29tL0NQZUBhQyBhQyBhQyBhQyBhQyBhQyBhQyBhQyBh
BzABhhhodHRwOi8vb2Nzc5kaWdpY2VyZC5jb20wTWYIKwYBBQUHMAKGQ2h0dHA6
Ly9jYyYybmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJmcmJm
NDIwMjM1OTU5WjBoMQswCQYDVVR0TBAlwADCCAX8GCisGAQQB1nkCBAlEGgfVBIIB
awFAHyAdv+IPwq2+5VRwmHM9Ye6NLskzbSP3GHCCp/nZ0xaOnQAAAGFdXNq0gAA
BAMARzBFAtEAncXmnInk6eUd0CDpAFmbhbIMPEyfyNlmfo2wnDa/UCIHPEh9A4
WpNiOs/CgwnnGe3gKQj7pqh7sstRTLjXx9fAHcASLDja9qmRZQP5WoC+p0w6xxS
ActW3SYB2bu/qznYHMAAGFdXNqMQAABAMASDBGAIEAmr217SFCLPjuTLQ97tg8
Zlmsyc5AucJM+UHaPTHILM0CIQCUIY5qNk15yQ5SAbd9wogcV2Px0D14GRwxDmpGnJ
LLa0AQ8B2ADTD3U+LbmAToswMwb+QDtn2E/D9Me9AA0tcn/h+tQXAABHXVzaI8A
AAQDAECwRIgF5MxpEzc0ynB3tr7S2J0BsCxJgXJOUsjsHsTF/Y6LKACIQCLZbM+
tJGldw9xfqZ6VievbVrxIC+ME0Ufy9HJEgEwfntAKBgqhkJOPQQDAwNnAD8KAjBN
i7a71+zuax/cZ7tovKnC6viUSchBfiCEZuiVEIJD9W0N06ieiJ30PBUBELCzxNoF4C
PA==
-----END CERTIFICATE-----
subject=C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com

issuer=C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: ECDSA
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 2755 bytes and written 383 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)

```

**The first command trusts and use the system's default CA, while the second command trusts only the certificates provided in “ca-certificates.crt” . the first command gives Verify return code: 20 (unable to get local issuer certificate) while the second results in Verify return code: 0 (ok).**

2. What does the error (verify error) in the first command mean? Please explain.

The "verify error" in the initial command indicates that the system couldn't validate the received public key's authenticity. This is due to the absence of the necessary certificate for verification.

3. Copy the server certificate (beginning with -----BEGIN CERTIFICATE----- and ending with -----END CERTIFICATE-----) and store it as twitter\_com.cert. Use the command `openssl x509 -in twitter_com.cert -text` to show a text representation of the certificate content. Briefly explain what is stored in an X.509 certificate (i.e. data in each field).

1. **Version:** Specifies the X.509 version used. Here, it's version 3 (0x2 in hexadecimal).
2. **Serial Number:** A unique number assigned by the Certificate Authority (CA) during issuance. This helps in identifying the certificate uniquely among others issued by the same CA.
3. **Signature Algorithm:** The algorithm used by the CA to sign this certificate. In this case, it's `ecdsa-with-SHA384`.
4. **Issuer:** Provides details about the Certificate Authority (CA) that issued the certificate. Here, the issuer is "DigiCert Inc" and the specific certificate used for signing is "DigiCert TLS Hybrid ECC SHA384 2020 CA1".
5. **Validity:**
  - **Not Before:** The start date of the certificate's validity.
  - **Not After:** The expiration date of the certificate.
6. **Subject:** Details about the entity for which the certificate was issued. It indicates that this certificate is for twitter.com owned by "Twitter, Inc." based in San Francisco, California.
7. **Subject Public Key Info:** Contains the public key details:
  - **Public Key Algorithm:** Specifies the type of public key, here it's an elliptic curve public key (`id-ecPublicKey`).
  - **Public-Key:** The actual public key value.
  - **ASN1 OID & NIST CURVE:** Identify the specific elliptic curve used. Here, it's `prime256v1` or commonly known as P-256.
8. **X509v3 extensions:** A set of additional properties and metadata for the certificate:
  - **Authority Key Identifier & Subject Key Identifier:** These are identifiers that help in linking certificates in a chain.
  - **Subject Alternative Name:** Alternative names for which this certificate is valid. Here, it's valid for both `twitter.com` and `www.twitter.com`.
  - **Key Usage & Extended Key Usage:** Define purposes for which the public key can be used. This certificate is primarily for server and client authentication over TLS.
  - **CRL Distribution Points:** URLs where browsers can check if this certificate has been revoked.
  - **Certificate Policies:** Identifies the policy under which the certificate has been issued and a link (CPS) to a detailed Certification Practice Statement.
  - **Authority Information Access:** Provides links to the OCSP server for real-time revocation checking and to the CA's certificate.
  - **Basic Constraints:** Indicates if the certificate is a CA certificate. Here, it's not (`CA:FALSE`).
  - **CT Precertificate SCTs:** These are Signed Certificate Timestamps, evidence that the certificate has been logged in public Certificate Transparency logs. This helps in detecting mistakenly or maliciously issued certificates.
9. **Signature Algorithm:** This is repeated at the end, just before the actual signature. It indicates the algorithm used to create the signature below.
10. **Signature:** The digital signature generated by the CA. This is used to verify that the certificate was genuinely issued by the stated CA.

```

Certificate:
Data:
  Version: 3 (0x2)
  Serial Number:
    0b:1c:5c:6b:b0:98:a1:86:61:92:76:d6:3f:95:f0:49
  Signature Algorithm: ecdsa-with-SHA384
  Issuer: C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1
  Validity
    Not Before: Jan  3 00:00:00 2023 GMT
    Not After : Jan  3 23:59:59 2024 GMT
  Subject: C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com
  Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (256 bit)
    pub:
      04:c1:69:27:1a:19:1c:4b:30:b6:56:bd:e6:e9:82:
      78:8c:77:27:89:00:f7:79:4b:ca:eb:90:74:00:18:
      5d:d5:66:06:2a:7b:90:36:2e:c9:02:85:cf:34:38:
      08:f5:19:1f:d6:b3:8d:a4:b0:2c:4c:49:e8:a6:15:
      a7:cc:b2:f6:1b
    ASN1 OID: prime256v1
    NIST CURVE: P-256
  X509v3 extensions:
    X509v3 Authority Key Identifier:
      keyid:0A:BC:08:29:17:8C:A5:39:6D:7A:0E:CE:33:C7:2E:B3:ED:FB:C3:7A

    X509v3 Subject Key Identifier:
      01:9D:52:0F:57:8E:C7:A5:60:6C:A0:8D:7A:1B:E3:FF:57:7C:8C:67
    X509v3 Subject Alternative Name:
      DNS:twitter.com, DNS:www.twitter.com
    X509v3 Key Usage: critical
      Digital Signature
    X509v3 Extended Key Usage:
      TLS Web Server Authentication, TLS Web Client Authentication
    X509v3 CRL Distribution Points:

      Full Name:
        URI:http://crl3.digicert.com/DigiCertTLSHybridECCSHA3842020CA1-1.crl

      Full Name:
        URI:http://crl4.digicert.com/DigiCertTLSHybridECCSHA3842020CA1-1.crl

    X509v3 Certificate Policies:
      Policy: 2.23.140.1.2.2
      CPS: http://www.digicert.com/CPS

  Authority Information Access:
    OCSP - URI:http://ocsp.digicert.com
    CA Issuers - URI:http://cacerts.digicert.com/DigiCertTLSHybridECCSHA3842020CA1-1.crt

  X509v3 Basic Constraints:
    CA:FALSE
  CT Precertificate SCTs:
    Signed Certificate Timestamp:
      Version : v1 (0x0)
      Log ID  : 76:FF:88:3F:0A:B6:FB:95:51:C2:61:CC:F5:87:BA:34:
        B4:A4:CD:BB:29:DC:68:42:0A:9F:E6:67:4C:5A:3A:74
      Timestamp : Jan  3 02:26:16.250 2023 GMT
      Extensions: none
      Signature : ecdsa-with-SHA256
        30:45:02:21:00:9F:17:0C:9C:89:E4:E9:E5:1D:D0:20:
        E9:00:53:1B:86:10:48:30:F1:32:17:23:67:94:C7:E8:
        DB:09:C3:6B:F5:02:20:73:C4:87:D0:38:5A:93:48:A3:
        9F:C2:83:0C:27:19:ED:E0:29:08:FB:A6:A8:7B:B2:CB:
        51:31:32:E3:5F:1F:5F

    Signed Certificate Timestamp:
      Version : v1 (0x0)
      Log ID  : 48:B0:E3:6B:DA:A6:47:34:0F:E5:6A:02:FA:9D:30:EB:
        1C:52:01:CB:56:DD:2C:81:D9:BB:BF:AB:39:D8:84:73
      Timestamp : Jan  3 02:26:16.241 2023 GMT
      Extensions: none
      Signature : ecdsa-with-SHA256
        30:46:02:21:00:9A:BD:B5:ED:27:E5:08:F8:EE:4E:54:
        3D:EE:D8:3C:65:69:AC:C9:CE:40:B9:C2:4C:F9:41:DA:
        3D:38:48:94:CD:02:21:00:94:63:98:0D:93:5E:72:43:
        9B:00:6D:DF:70:A2:07:15:D8:FC:43:D7:81:91:C3:10:
        E6:A4:69:C9:94:B6:B4:01

    Signed Certificate Timestamp:
      Version : v1 (0x0)
      Log ID  : 3B:53:77:75:3E:2D:B9:80:4E:88:30:5B:06:FE:40:3B:
        67:D8:4F:C3:F4:C7:BD:00:0D:2D:72:6F:E1:FA:D4:17
      Timestamp : Jan  3 02:26:16.239 2023 GMT
      Extensions: none
      Signature : ecdsa-with-SHA256
        30:45:02:20:17:93:31:A4:4C:DC:D3:29:C1:DE:D4:7B:
        4B:62:74:06:C0:B1:26:05:C9:39:4B:23:B0:7B:13:7F:
        F6:3A:2C:A0:02:21:00:A5:65:B3:3E:B4:91:96:C3:DC:
        5F:A9:9E:95:89:EB:DB:56:BC:62:0B:E3:04:D1:47:F2:
        F4:72:44:80:4C:1F:9D

  Signature Algorithm: ecdsa-with-SHA384
    30:64:02:30:4d:8b:b8:3b:8f:ec:ee:ab:1f:dc:4d:9b:69:c8:
    a9:82:ea:f5:39:72:10:5f:d4:21:19:ba:35:44:20:9f:43:58:
    e4:3a:89:e8:89:dc:e3:fc:53:c7:8b:0b:1c:d7:a4:5e:02:30:
    68:f6:c9:62:82:4d:11:b5:b4:75:fb:cb:cf:37:9c:ee:26:e1:
    f4:e6:a9:40:4e:44:1a:eb:2c:f1:ae:0c:91:76:31:43:be:74:
    9c:fa:32:93:95:67:38:40:d3:9e:93:3c

```



4. From the information in exercise 3, is there an intermediate certificate? If yes, what purpose does it serve?

**Yes, there is an intermediate certificate. Its function is to validate the intermediate TLS Server, namely "DigiCert TLS Hybrid ECC SHA384 2020 CA1"**

5. Is there an intermediate CA, i.e. is there more than one organization involved in the certification? Say why you think so.

**Yes there is, base on the results from running openssl s\_client -connect twitter.com:443**

```
parallels@ubuntu-linux-20-04-desktop:~$ openssl s_client -connect twitter.com:443 -CAfile /etc/ssl/certs/ca-certificates.crt
CONNECTED(00000000)
depth=2 C = US, O = DigiCert Inc, OU = www.digicert.com, CN = DigiCert Global Root CA
verify return:1
depth=1 C = US, O = DigiCert Inc, CN = DigiCert TLS Hybrid ECC SHA384 2020 CA1
verify return:1
depth=0 C = US, ST = California, L = San Francisco, O = "Twitter, Inc.", CN = twitter.com
verify return:1
```

6. What is the role of ca-certificates.crt?

**The file "ca-certificates.crt" contains the trusted root certificates of various Certificate Authorities (CAs). It plays a pivotal role in the trust framework by containing the public keys of these trusted CAs. Our devices rely on these keys to verify certificates presented by websites or services. Essentially, if a certificate can be traced back to one of the trusted roots in this file, then the entity presenting the certificate is deemed trustworthy by our device.**

7. Explore the ca-certificates.crt. How many certificates are in there? Give the command/method you have used to count.

**openssl storeutl -noout -certs /etc/ssl/certs/ca-certificates.crt    Total found: 137**

8. Extract a root certificate from ca-certificates.crt. Use the openssl command to explore the details. Do you see any Issuer information? Please compare it to the details of twitter's certificate and the details of the intermediate certificate.

```
parallels@ubuntu-linux-20-04-desktop:~/Desktop/Parallels Shared Folders/Home/desktop$ openssl x509 -in '/home/parallels/Desktop/Parallels Shared Folders/Home/Desktop/class-lecture/2023S12110413-Computer-Security-Activity/activity5/certificates/root.cert' -text
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 6828503384748696800 (0x5ec3b7a6437fae0)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: CN = ACCVRAIZ1, OU = PKIACCV, O = ACCV, C = ES
    Validity
      Not Before: May  5 09:37:37 2011 GMT
      Not After : Dec 31 09:37:37 2030 GMT
    Subject: CN = ACCVRAIZ1, OU = PKIACCV, O = ACCV, C = ES
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public-Key: (4096 bit)
      Modulus:
        00:9b:a9:ab:bf:61:4a:97:af:2f:97:66:9a:74:5f:
        d0:d9:96:fd:cf:e2:e4:66:ef:1f:1f:47:33:c2:44:
        a3:df:9a:de:1f:b5:54:dd:15:7c:69:35:11:6f:bb:
        c8:0c:8e:6a:18:1e:d8:8f:d9:16:bc:10:48:36:5c:
        f0:63:b3:90:5a:5c:24:37:d7:a3:d6:cb:09:71:b9:
        f1:01:72:84:b0:7d:db:4d:80:cd:fc:d3:6f:c9:f8:
        da:b6:0e:82:d2:45:85:a8:1b:68:a8:3d:e8:f4:44:
        6c:bd:a1:c2:cb:03:be:8c:3e:13:00:84:df:4a:48:
        c0:e3:22:0a:e8:e9:37:a7:18:4c:b1:09:0d:23:56:
        7f:04:4d:d9:17:84:18:a5:c8:da:40:94:73:eb:ce:
        0e:57:3c:03:81:3a:9d:0a:a1:57:43:69:ac:57:6d:
        79:90:78:e5:b5:b4:3b:d8:bc:4c:8d:28:a1:a7:a3:
        a7:ba:02:4e:25:d1:2a:ae:ed:ae:03:22:b8:6b:20:
        0f:30:28:54:95:7f:e0:ee:ce:0a:66:9d:d1:40:2d:
        6e:22:af:9d:1a:c1:05:19:d2:6f:c0:f2:9f:f8:7b:
        b3:02:42:fb:50:a9:1d:2d:93:0f:23:ab:c6:c1:0f:
        92:ff:d0:a2:15:f5:53:09:71:1c:ff:45:13:84:e6:
        26:5e:f8:e0:88:1c:0a:fc:16:b6:a8:73:06:b8:f0:
        63:84:02:a0:c6:5a:ec:e7:74:df:70:ae:a3:83:25:
        ea:d6:c7:97:87:93:a7:c6:8a:8a:33:97:60:37:10:
        3e:97:3e:6e:29:15:d6:a1:0f:d1:88:2c:12:9f:6f:
        aa:a4:c6:42:eb:41:a2:e3:95:43:d3:01:85:6d:8e:
        bb:3b:f3:23:36:c7:fe:3b:e0:a1:25:07:48:ab:c9:
        89:74:ff:08:8f:80:bf:c0:96:65:f3:ee:ec:4b:68:
        bd:9d:88:c3:31:b3:40:f1:e8:cf:f6:38:bb:9c:e4:
        d1:7f:d4:e5:58:9b:7c:fa:d4:f3:0e:9b:75:91:e4:
        ba:52:2e:19:7e:d1:f5:cd:5a:19:fc:ba:06:f6:fb:
        52:a8:4b:99:04:dd:f8:f9:b4:8b:50:a3:4e:62:89:
        52:a8:4b:99:04:dd:f8:f9:b4:8b:50:a3:4e:62:89:
        f0:87:24:fa:83:42:c1:87:fa:d5:2d:29:2a:5a:71:
        7a:64:6a:d7:27:60:63:0d:db:ce:49:f5:8d:1f:90:
        89:32:17:f8:73:43:b8:d2:5a:93:86:61:d6:e1:75:
        0a:ea:79:66:76:88:4f:71:eb:04:25:d6:0a:5a:7a:
        93:e5:b9:4b:17:40:0f:b1:b6:b9:f5:de:4f:dc:e0:
        b3:ac:3b:11:70:60:84:4a:43:6e:99:20:c0:29:71:
        0a:c0:65
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      Authority Information Access:
        CA Issuers - URI:http://www.accv.es/fileadmin/Archivos/certificados/raizaccv1.crt
        OCSP - URI:http://ocsp.accv.es

      X509v3 Subject Key Identifier:
        D2:87:B4:E3:DF:37:27:93:55:F6:56:EA:81:E5:36:CC:8C:1E:3F:BD
      X509v3 Basic Constraints: critical
        CA:TRUE
      X509v3 Authority Key Identifier:
        keyid:D2:87:B4:E3:DF:37:27:93:55:F6:56:EA:81:E5:36:CC:8C:1E:3F:BD

      X509v3 Certificate Policies:
        Policy: X509v3 Any Policy
        User Notice:
          Explicit Text:
            CPS: http://www.accv.es/legislacion_c.htm

      X509v3 CRL Distribution Points:

        Full Name:
          URI:http://www.accv.es/fileadmin/Archivos/certificados/raizaccv1_der.crl

      X509v3 Key Usage: critical
        Certificate Sign, CRL Sign
      X509v3 Subject Alternative Name:
        email:accv@accv.es
    Signature Algorithm: sha1WithRSAEncryption
    97:31:02:9f:e7:fd:43:67:48:44:14:e4:29:87:ed:4c:28:66:
    d0:8f:35:da:4d:61:b7:4a:97:4d:b5:db:90:e0:05:2e:0e:c6:
    79:d0:f2:97:69:0f:bd:04:47:d9:be:db:b5:29:da:9b:d9:ae:
    a9:99:d5:d3:3c:30:93:f5:8d:a1:a8:fc:06:8d:44:f4:ca:16:
    95:7c:33:dc:62:8b:a8:37:f8:27:d8:09:2d:1b:ef:c8:14:27:
    20:a9:64:44:ff:2e:d6:75:aa:6c:4d:60:40:19:49:43:54:63:
    da:e2:cc:ba:66:e5:4f:44:7a:5b:d9:6a:81:2b:40:d5:7f:f9:
    01:27:58:2c:c8:ed:48:91:7c:3f:a6:00:cf:c4:29:73:11:36:
    de:86:19:3e:9d:ee:19:8a:1b:d5:b0:ed:8e:3d:9c:2a:c0:0d:
```

```

d8:3d:66:e3:3c:0d:bd:d5:94:5c:e2:a7:35:1b:04:00:f6:
3f:5a:8d:ea:43:bd:5f:89:1d:a9:c1:b0:cc:99:e2:4d:00:0a:
da:c9:27:5b:e7:13:90:5c:e4:f5:33:a2:55:6d:dc:e0:09:4d:
2f:b1:26:5b:27:75:00:09:c4:62:77:29:08:5f:9e:59:ac:b6:
7e:ad:9f:54:30:22:03:c1:1e:71:64:fe:f9:38:0a:96:18:dd:
02:14:ac:23:cb:06:1c:1e:a4:7d:8d:0d:de:27:41:e8:ad:da:
15:b7:b0:23:dd:2b:a8:d3:da:25:87:ed:e8:55:44:4d:88:f4:
36:7e:84:9a:78:ac:f7:0e:56:49:0e:d6:33:25:d6:84:50:42:
6c:20:12:1d:2a:d5:be:bc:f2:70:81:a4:70:60:be:05:b5:9b:
9e:04:44:be:61:23:ac:e9:a5:24:8c:11:80:94:5a:a2:a2:b9:
49:d2:c1:dc:d1:a7:ed:31:11:2c:9e:19:a6:ee:e1:55:e1:c0:
ea:cf:0d:84:e4:17:b7:a2:7c:a5:de:55:25:06:ee:cc:c0:87:
5c:40:da:cc:95:3f:55:e0:35:c7:b8:84:be:b4:5d:cd:7a:83:
01:72:ee:87:e6:5f:1d:ae:b5:85:c6:26:df:e6:c1:9a:e9:1e:
02:47:9f:2a:a8:6d:a9:5b:cf:ec:45:77:7f:98:27:9a:32:5d:
2a:e3:84:ee:c5:98:66:2f:96:20:1d:dd:d8:c3:27:d7:b0:f9:
fe:d9:7d:cd:d0:9f:8f:0b:14:58:51:9f:2f:8b:c3:38:2d:de:
e8:8f:d6:8d:87:a4:f5:56:43:16:99:2c:f4:a4:56:b4:34:b8:
61:37:c9:c2:58:80:1b:a0:97:a1:fc:59:8d:e9:11:f6:d1:0f:
4b:55:34:46:2a:8b:86:3b
-----BEGIN CERTIFICATE-----
MIIEH0zCCBugAwIBAgIIXsO3pkN/p0AwDQYJKoZIhvcNAQEFBQAwQjESMBAGA1UE
AwwJQUUNDVlJBSVoxMRAwDgYDVQQLADAdQ08lBQ0NMWQ9CwYDVQQKQARBQ0NMWQsw
CQYDVQQGEwJFZUzAeFw0xMTA1MDUwOTMzMzdaFw0zMDUyMzEwOTMzMzdaMEIxIjAq
BgNVBAMwCUFFDQ1ZSUlAmTEQMA4GA1UECwwHUETJQUINDVjENMAsGA1UECgwEQUND
VjELMAkGA1UEBhMCRVwggIiMA0GCSCqSISB3DQEBQUAAICDWAwwgIKAoICAQCb
qau/YUqKry+Xzp0X9DZLv3P4uRn7x8FRzPCRKPfnt4ftVTdFXxpNRfVuu8gMjmoY
HtiP2Ra8EEg2XPBjs58aXCQ316PWywLxufEBcoSwdftNgM3802/J+Nq2D0LSRYWo
G2IoPej0Ry9ocLLA76MPHMAHn9K5MDjIgro6TenGEyxQ0jVn8ETdkXhBilyNpA
lHPrzg5XPA0B0p0KoVdDaaxXbXmQe0W1tDvYvEyNKKGno6e6Ak4l05qu7a4Dirhr
IA8WKF5VF+DuzgpmndFALW4irS0awQUZ0m/A8p/4e7MCQvtQ0R0tkw8jq8bBD5L/
0KTV9VMCRz/RR0E5iZe+OCiHAr8FRaocwa48G0EAQDGWuzndN9wrq0DJerWx5eH
k6Fci0oz12A3ED6Xpm4pFdahD9GILBKfb6qkxLrQaLjLUPTAYVtjrs78yM2x/47
4KElB0iryYl0/wiPgL/AlmXz7uxLaL2dImXs0Dx6M/20Luc5NF/10Vym3z61PMO
n3WR5Lp5Lh1+0fXNWhn8ugb2+1Ko5S5kE3fj5tItQ085iifCHJPQ0QsGh+UtkSpa
cXpkatcnYGMN285J9Y0fKikyF/hzQ7j5Wp0GQydbhdQrqewZ2iE9x6wQ1QpaepPL
uUsXQA+xtmn13k/c4L0s0xwFyIRKQ26ZIMApCqRzAQIDAQABo4ICyzCCAscwfQYI
KwYBBQUHAQEETBVMwGCCsGAQUFBzAChk8odHRwOi8vd3d3LmFjY3YuZXh1bWZmLs
ZWFKbWl0L0FyY2hpdn9zL2NlcnRpZmZlYmRvcy9yYmV6YmVjZjEuY3J0MB8GCCS
GAQUFBzABhhNodHRwOi8vd3d3LmFjY3YuZXh1bWZmLsZWFKbWl0L0FyY2hpdn9z
VuqB5tBmJb4/vTAPBgNVHRMBAF8EBTADAQH/MB8GA1UdIwQYMBaAFNKHtOPfNyeT
VFZW60HlnsYMHj+9MIIBcwYDVVR0dBIBAgIICCAwYwggFiBgRVHSAAMIIBWDCASIG
CCsGAQUFBwICMIIBFB6CARAAQ0B1AHQAbwByAGkAZABhAGUATABkAGUATABDAGUA
cgB0AGkAZgBpACMAyQ0BjAGkA8wBpACAAUgBhA0B0AegAgAGQAZQAgAGwAYQAgAEFA
QwBDAFYATAAoAEFAZwBjAG4AYwBpACAAUgBhA0B0AegAgAGQAZQAgAGwAYQAgAEFA
7QBhACAAeQAgAEFAZQ0BjAG4AYwBpACAAUgBhA0B0AegAgAGQAZQAgAGwAYQAgAEFA
cgDZAG4Aa0BjAG4EALAAgAEFASQBGAUAAUgBhA0B0AegAgAGQAZQAgAGwAYQAgAEFA
QwBQAFMAIABlAG4AIABoAHQAdABwADoALwAvAHcAdwB3AC4AYQ0BjAG4MAdgAuAGUA
czAwBggrBgEFBQcCARYkaHR0cDovL3d3dy5hY2N2LmVzL2x1Z2LzBGFjaW9uX2Mu
aHRtMfUGA1UdHwROMeWwSgBIEoEaGRGh0dHA6Ly93d3cuYmV6YmVjZjEuY3J0MB8GCC
sGAQUFBzABhhNodHRwOi8vd3d3LmFjY3YuZXh1bWZmLsZWFKbWl0L0FyY2hpdn9z
aW4vQXJjaG12b3MvY2VydgG1mahNhZG9zL3JhaXphY2N2Mw9kZXIuY3J0MA4GA1Ud
DwEB/wQEAwIBBjAXBgNVHREEDAAQ0xhY2N2QGfjY3YuZXh1bWZmZmZmZmZmZmZmZmZm
BQADggIBA3cxAp/n/UNnSEQU5CmH7UwoZtCPNdpNYbdKl02125DgBS40xnnQ8pdp
D70ER9m+27Up2pvZrqmZ1dM8MJP1jaGo/AaNRPTKfPv8M9xi16g3+CfYCS0b78gU
JyCpZET/Lt21qnxNYEASUNUY9riZLpm5U9EeLvZaoErQNV/+QEnWCzI7UlrFD+m
AM/EXXMRnt6GGT6d7hmKG9W7Y49nCrAddg9ZuM8Db3VLFzi4qc1GwQA9j9aJepD
Vv+JHanBsMyZ4k0ActrJJ1vnESBc5PUzoLvt30AJTS+xxJLsndQAJxG3JKQhfnlms
tn6tn1QwIgPBHnFk/vk4CpYY3QIUrcPLBhwepH2NDd4nQeit2hW3sCPdK6jT2iWH
7ehVRE2I9Dz+hJp4rPcOVkk01jMl1oRQ0mwgEh0q1b688nCBpHBgvgW1m54ERLSh
I6zppSSMEYCUwKluUnSwdzRp+0xE5yGaba4VXhwOrPDYTKF7eiFKXevSUG7szA
h1x2sYVP1XgNce4hL60Xc16gwFy7ofmXx2utYXGJt/mwZrpHgJHnyqoba1bz+xF
d3+YJ5oyXsRjh07FmCYvliAd3dJdJ9ew+f7Zfc3Qn48LFFhRny+Lwzgt3uip1o2H
pPVWQxaZLPSkVrQ0uGE3ycJYgBugl6H8WY3pEFBRD0tVNEYqi4Y7
-----END CERTIFICATE-----
parallel@ubuntu-linux-20-04-desktop:~/Desktop/Parallels Shared Folders/Home/desktop$

```

The issuer and subject are identical, which is a distinction from the intermediate and Twitter certificates, where the issuer and subject represent different entities.

9. If the intermediate certificate is not in a PEM format (text readable), use the command to convert a DER file (.crt .cer .der) to PEM file.

**openssl x509 -inform der -in certificate.cer -out certificate.pem.**

(You need the pem file for exercise 10.)

10. From the given python code,<sup>1</sup> implement the certificate validation.

```

1  from OpenSSL import crypto
2  import pem
3
4
5  def verify(target_filename, intermediate_filenames, root_filename):
6      with open(target_filename, "r") as cert_file:
7          cert = cert_file.read()
8          int_certs = []
9          for filename in intermediate_filenames:
10             with open(filename, "r") as cert_file:
11                 int_certs.append(cert_file.read())
12             pems = pem.parse_file(root_filename)
13             trusted_certs = [str(mypem) for mypem in pems] + int_certs
14
15             verified = verify_chain_of_trust(cert, trusted_certs)
16             if verified:
17                 print("Certificate verified")
18             else:
19                 print("Certificate verification failed")
20
21
22  def verify_chain_of_trust(cert_pem, trusted_cert_pems):
23      certificate = crypto.load_certificate(crypto.FILETYPE_PEM, cert_pem)
24      store = crypto.X509Store()
25
26      for trusted_cert_pem in trusted_cert_pems:
27          trusted_cert = crypto.load_certificate(crypto.FILETYPE_PEM, trusted_cert_pem)
28          store.add_cert(trusted_cert)
29
30      store_ctx = crypto.X509StoreContext(store, certificate)
31      result = store_ctx.verify_certificate()
32      if result is None:
33          return True
34      else:
35          return False
36
37
38  print("Verifying Twitter certificate...")
39  verify("twitter_com.cert", ["int_twitter_com.cert"], "ca-certificates.cert")
40  print("Verifying Google certificate...")
41  verify(
42      "google_com.cert",
43      ["int_google_com.cert", "int2_google_com.cert"],
44      "ca-certificates.cert",
45  )
46  print("Verifying Chula certificate...")
47  verify("chula_ac_th.cert", ["int_chula_ac_th.cert"], "ca-certificates.cert")
48  print("Verifying Classdeedee certificate...")
49  verify("classdeedee.cert", ["int_classdeedee.cert"], "ca-certificates.cert")
50

```

```

pawankanjeam@Pawans-MacBook-Pro: ~/Desktop/class-lecture/2023S12110413-Computer-Security-Activity/activity5/for#10 $ main ? /opt/homebrew/bin/python3 /Users/pawankanjeam/Desktop/class-lecture/2023S
12110413-Computer-Security-Activity/activity5/for#10/certificate_validation.py
Verifying Twitter certificate...
Certificate verified
Verifying Google certificate...
Certificate verified
Verifying Chula certificate...
Certificate verified
Verifying Classdeedee certificate...
Certificate verified

```



11. Nowadays, there are root certificates for class 1 and class 3. What uses would a class 1 signed certificate have that a class 3 doesn't, and vice versa?

**Class 1 is a legacy root certificate encompassing both low and high-security validations, while Class 3 represents a more recent, elevated-security certificate. Class 1 is compatible with older browsers, but Class 3 caters to modern browsers with enhanced security requirements.**

12. Assuming that a Root CA in your root store is hacked and under the control of an attacker, and this is not noticed by anyone for months.

- A. What further attacks can the attacker stage? Draw a possible attack setup.
- B. In the attack you have described above, can we rely on CRLs or OCSP for protection? Please explain

**A. Attack Setup :**

If a Root CA is compromised:

- Fraudulent Certificates: Attackers can issue fake certificates for any domain.
- Man-in-the-Middle: They can intercept and alter communications without users realizing.
- Data Theft: Users, thinking they're on genuine sites, might give away personal data.
- Malware Distribution: Users could be redirected to malicious sites that look legitimate.
- Reputation Damage: Fake sites can spread misinformation.

**B. CRLs and OCSP :**

If the Root CA breach is undetected, CRLs and OCSP might not help much. There's often a delay in revocation, not all sites use the OCSP "must-staple" feature, and if the CA itself is compromised, its revocation data might be manipulated. Thus, relying solely on these tools in such a scenario would be inadequate.