Tutorial 7 (Authentication and Key Distribution)

In this tutorial, we will complete some practical exercises on digital certificates.

🔐 Introduction: What Are Digital Certificates?

Digital certificates are a core component of modern internet security.

They are used to:

- ✅ Authenticate the identity of websites and servers
- ✅ Establish encrypted channels (e.g. HTTPS/TLS)
- ✅ Enable trust without sharing secrets in advance

Most secure protocols today — including HTTPS, TLS, VPNs, and code signing — rely on digital certificates to prevent impersonation and man-in-the-middle attacks.

📄 What Is a Digital Certificate?

A **digital certificate** is a structured, signed document that binds a **public key** to a verified identity (such as a domain name or an organization).

It follows the **X.509 standard** and is typically issued by a **Certificate Authority (CA)**.

🔍 What Does a Certificate Contain?

A digital certificate includes key fields required to verify identity and establish secure connections. Here's a simplified overview:

Field	Description
Subject	The identity being certified (e.g., domain)
Issuer	The Certificate Authority (CA) that issued it
Public Key	Used to establish secure communication
Validity Period	Start and end time for certificate validity
Serial Number	Unique identifier for the certificate
Signature	CA's cryptographic signature over the contents

> 🔎 **Note:** Real certificates include additional fields (e.g., extensions, key usage, algorithms) that control behavior and compatibility across systems.

🧪 Common Formats You Will Encounter

- `.pem` — Base64-encoded certificate, used by OpenSSL, Apache, Nginx
- `.cer` — Can be **PEM or DER** (binary); used in Windows and Java environments
- `.crt` — Often identical to `.pem`; varies by convention

🎯 Why This Matters

If you work with secure websites, APIs, VPNs, or authentication protocols, you must understand how certificates:

- Prove identity
- Enable encryption
- Fail when expired, revoked, or tampered with

📘 In This Tutorial

You will:

- Examine and decode real certificate files
- Understand the differences between `.pem`, `.cer`, `.crt`, and `.der`
- Use OpenSSL to parse, validate, and convert certificates
- Learn how integrity and trust are established using certificate chains

1) Exploring Digital Certificates

a) Using OpenSSL

OpenSSL is a command-line tool that can be used to display the contents of an X.509 certificate. We provide a certificate file: `whoami.pem`. Use `openssl x509 -help` to find out how to display the certificate in a human-readable format. Then, answer the following questions:

- Who is the issuer? Look up the organization.
- Who is the subject (i.e., the certified entity)?
- What is the algorithm used for the public key?
- What is the key length?
- Are Certificate Revocation Lists (CRLs) defined?
- Is Online Certificate Status Protocol (OCSP) defined?
- What hash function is used for the signature?

In [11]:
```
!openssl x509 -in whoami.pem -text -noout
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            72:ea:64:de:cb:5f:12:bf:fe:29:04:ad:30:8b:dd:ca:9d:d9:e3:9e
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=BM, O=QuoVadis Limited, CN=QuoVadis Global SSL ICA G3
        Validity
            Not Before: Mar  9 05:53:39 2018 GMT
            Not After : Mar  9 06:03:00 2020 GMT
        Subject: C=AU, ST=New South Wales, L=Sydney, O=The University of S
ydney, OU=Information and Communications Technology, CN=*.sydney.edu.au
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:b2:28:4e:6d:7b:2e:c0:ef:dc:f9:4a:5f:a9:38:
                    ef:6d:37:3e:00:96:5a:bf:b7:91:01:c4:27:af:23:
                    82:f6:a3:15:fc:5f:eb:6c:59:42:a7:2a:a4:bd:22:
                    c4:8e:c8:dd:73:b9:68:a9:e0:58:35:04:ec:42:91:
                    26:4e:58:37:ea:14:cd:a6:c7:1b:26:fe:fe:43:91:
                    94:8f:50:56:99:ec:2a:d4:93:9b:08:81:a9:a2:63:
                    85:72:a8:54:0d:77:25:1c:62:1d:e0:97:e9:5a:99:
                    94:6c:e7:75:b2:dc:44:6f:83:c3:33:a7:a8:e1:91:
                    99:03:8e:c9:0f:a5:a4:68:7e:61:64:41:07:55:36:
                    32:e1:f6:a7:89:8a:d3:be:71:42:1d:08:a9:48:c2:
                    a7:96:88:1a:06:0a:d1:61:e4:60:2d:46:bd:44:0c:
                    2b:4e:eb:bc:6b:1e:2c:97:64:45:10:4d:76:84:0a:
                    32:89:1a:4a:1e:e5:16:7d:7c:c4:4b:0f:a0:64:1e:
                    b2:74:22:96:e3:df:c9:ad:43:0d:b4:72:5a:b9:c0:
                    4d:bc:cf:e1:72:ea:79:b2:37:f6:4e:cd:85:cf:ae:
                    c7:a4:8c:c5:4e:e2:ca:25:b0:60:1d:c7:b2:51:98:
                    22:b0:af:bb:fd:52:40:bd:1f:14:2f:2f:15:59:b7:
                    1a:e5
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Authority Key Identifier:
                B3:12:89:B5:A9:4B:35:BC:15:00:F0:80:E9:D8:78:87:F1:13:7C:7
6
            Authority Information Access:
                CA Issuers - URI:http://trust.quovadisglobal.com/qvsslg3.c
rt
                OCSP - URI:http://ocsp.quovadisglobal.com
            X509v3 Subject Alternative Name:
                DNS:*.sydney.edu.au, DNS:sydney.edu.au
            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.8024.0.2.100.1.1
                  CPS: http://www.quovadisglobal.com/repository
            X509v3 Extended Key Usage:
                TLS Web Client Authentication, TLS Web Server Authenticati
on
            X509v3 CRL Distribution Points:
                Full Name:
                  URI:http://crl.quovadisglobal.com/qvsslg3.crl

            X509v3 Subject Key Identifier:
                A9:65:57:67:6B:79:41:73:CA:83:9A:4D:DE:92:3E:68:C3:6E:93:3
D
```

```
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            CT Precertificate SCTs:
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : BB:D9:DF:BC:1F:8A:71:B5:93:94:23:97:AA:92:
7B:47:
                                38:57:95:0A:AB:52:E8:1A:90:96:64:36:8E:1E:
D1:85
                    Timestamp : Mar  9 06:03:39.513 2018 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:46:02:21:00:E1:B5:97:7F:1A:3C:BB:2C:DE:
64:A2:
                                44:97:C1:A4:62:01:D3:83:27:18:75:F2:22:FB:
93:A9:
                                C1:BC:93:F2:54:02:21:00:D6:7A:91:E4:DD:B3:
2C:9C:
                                30:95:DA:6F:EC:D9:E3:B0:89:FC:D5:9F:18:C3:
B5:9A:
                                E3:EA:13:35:BE:E2:70:BC
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : A4:B9:09:90:B4:18:58:14:87:BB:13:A2:CC:67:
70:0A:
                                3C:35:98:04:F9:1B:DF:B8:E3:77:CD:0E:C8:0D:
DC:10
                    Timestamp : Mar  9 06:03:39.549 2018 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:45:02:21:00:A0:A4:A6:1E:3B:AE:D6:6F:69:
11:10:
                                D4:75:C9:FC:EF:D2:50:1F:70:89:CC:06:95:3E:
FC:FF:
                                AC:B4:05:58:D9:02:20:0B:08:20:F8:7C:EB:8D:
E5:80:
                                83:43:67:D1:CF:F0:7E:A2:0E:A7:D5:0B:21:F4:
9B:41:
                                DD:21:3C:76:4C:12:35
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : 56:14:06:9A:2F:D7:C2:EC:D3:F5:E1:BD:44:B2:
3E:C7:
                                46:76:B9:BC:99:11:5C:C0:EF:94:98:55:D6:89:
D0:DD
                    Timestamp : Mar  9 06:03:40.220 2018 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:45:02:20:37:21:0C:C7:FD:25:D3:08:C0:C6:
B4:99:
                                08:7F:84:08:9F:0E:0D:13:0A:11:CE:B5:89:A9:
F2:98:
                                FC:AD:2F:E9:02:21:00:F4:CD:FF:0B:E2:0D:ED:
8F:57:
                                84:A0:42:DC:D2:31:0E:7F:C4:B2:06:84:3C:8D:
10:A2:
                                0E:A7:3F:7A:75:11:5E
                Signed Certificate Timestamp:
                    Version   : v1 (0x0)
                    Log ID    : 6F:53:76:AC:31:F0:31:19:D8:99:00:A4:51:15:
```

```
FF:77:
                                15:1C:11:D9:02:C1:00:29:06:8D:B2:08:9A:37:
D9:13
                    Timestamp : Mar  9 06:03:40.362 2018 GMT
                    Extensions: none
                    Signature : ecdsa-with-SHA256
                                30:44:02:20:38:35:1B:72:CB:A4:83:AE:80:BE:
66:29:
                                89:73:04:FF:B9:9C:87:E5:11:77:CD:7B:96:DE:
E1:D3:
                                3A:66:F7:2C:02:20:66:AB:EC:6C:94:74:C5:5A:
A2:CA:
                                6A:18:17:99:E3:93:7E:10:EB:35:62:99:55:CC:
0B:ED:
                                87:87:FB:76:AC:5C
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        5d:2d:93:00:91:74:81:ae:83:4f:70:6b:67:64:09:47:73:c2:
        a9:ee:0f:28:0e:68:0d:93:59:09:56:27:94:ba:d9:10:d7:d7:
        7c:3f:7b:5f:22:42:92:d0:ef:46:24:35:47:9a:40:17:2c:7b:
        80:c1:4b:5c:42:59:d8:70:da:b4:a8:4e:c0:4d:9f:b1:7d:bd:
        c3:55:84:67:bf:f4:d1:27:83:11:94:45:a5:88:1a:3f:20:d5:
        de:f5:72:cf:6f:0d:90:08:97:f1:19:55:f9:01:d5:ae:15:6d:
        58:3f:c6:36:5a:d0:6c:ad:d3:fc:58:c8:f6:a8:80:e1:2c:77:
        e3:4c:45:96:0e:be:e1:26:16:eb:d5:21:f2:4c:17:41:28:3e:
        18:0c:84:cb:81:3a:fb:af:3c:19:a8:fa:08:72:e8:de:e6:96:
        63:3d:14:5b:41:b9:30:1b:c2:7a:2b:99:f2:a4:cb:97:8a:b5:
        b9:6b:f6:bc:98:7d:23:53:22:07:58:d5:85:6e:c2:2b:5a:99:
        36:86:f8:54:4b:35:01:f2:99:9f:b5:a9:3b:dd:9e:27:78:e8:
        99:1e:ef:f7:31:2e:96:56:fd:30:85:b0:1a:4b:bf:e2:cc:87:
        4b:1e:e4:50:86:2b:74:3f:3b:35:50:49:4c:36:71:6f:57:b5:
        0b:0a:89:e6:5f:bd:5b:82:f5:01:aa:a7:3e:0c:86:7d:0a:9d:
        0b:7f:b1:96:5f:3e:da:af:90:de:e4:eb:cd:76:60:e3:a1:2f:
        af:4a:09:40:b1:0c:a7:9a:75:cb:fc:c5:54:0a:f7:04:fb:2f:
        b1:14:fb:af:4b:ce:74:33:e5:43:54:b5:0f:b9:51:5a:8b:ef:
        b0:9b:6b:78:70:6e:a1:36:35:37:f9:0a:f4:ab:e5:7f:99:da:
        fc:e3:d3:68:a9:0a:98:e4:41:7e:49:b5:95:d7:14:03:18:c1:
        15:0b:91:2c:21:98:bd:58:91:81:87:b0:1d:fa:49:5f:46:6f:
        01:d1:29:2c:24:31:ab:06:03:f2:d4:d6:2c:48:3c:d1:8a:38:
        22:70:f3:4e:17:57:21:37:3e:12:93:c3:1e:60:0d:53:90:ba:
        a2:cd:1c:4a:31:1c:72:77:7d:9a:e5:d8:ee:33:b7:14:1c:94:
        f0:7e:d3:c0:5b:d8:08:65:49:9c:25:1a:b3:6b:20:b0:d3:f7:
        fd:e1:9f:57:69:de:5f:c6:e6:5a:58:62:b9:f9:77:33:df:54:
        8d:a0:7f:a6:af:82:a4:4d:fa:bc:eb:05:6a:63:cb:25:0a:72:
        71:cc:f3:2f:05:36:58:cd:25:87:b6:45:13:fc:19:e2:6a:c3:
        3f:62:aa:2d:8f:7f:e4:07
```

Solution:

- **Who is the issuer?** QuoVadis Limited
- **Who is the subject (the certified entity)?** The University of Sydney
- **What is the algorithm of the public key?** RSA Encryption
- **What is the key length?** 2048 bits
- **Are CRLs defined?** Yes, a URL is given: http://crl.quovadisglobal.com/qvsslg3.crl
- **Is OCSP defined?** Yes, a URL is given: http://ocsp.quovadisglobal.com

- **What hash function is used for the signature?** SHA256

b) Let's try one more: `fred.cer`

Credit: Applied Crypto by Bill Buchanan

Examine the certificate and find the following information:

- Serial number
- Effective date
- Name (i.e., the Subject)
- Issuer
- What is the **CN** (Common Name) used for?
- What is the **OU** (Organizational Unit) used for?
- What is the **O** (Organization) used for?
- What is the **L** (Locality) used for?

> **Note:**
> The `fred.cer` file is in **DER** (Distinguished Encoding Rules) format, a binary encoding often used with `.cer` extensions.
> This differs from the `.pem` format we examined earlier, which is a **Base64-encoded** version of the certificate wrapped with `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` markers.
> PEM is human-readable; DER is not. Tools like `openssl` can be used to decode `.cer` files for analysis.

In [2]:
```
!openssl x509 -in fred.cer -text -noout
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 7350616 (0x702958)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=GB, ST=Lothian, L=Edinburgh, O=Nowhere, emailAddress=fre
d@home, OU=None, CN=Fred Smith
        Validity
            Not Before: Apr 24 20:18:42 2008 GMT
            Not After : Apr 24 20:18:42 2010 GMT
        Subject: C=GB, ST=Lothian, L=Edinburgh, O=Nowhere, emailAddress=fr
ed@home, OU=None, CN=Fred Smith
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (1024 bit)
                Modulus:
                    00:ca:62:fb:39:a9:a6:2a:78:db:14:4f:6a:f8:e1:
                    8c:90:bc:a8:97:f5:74:2e:cb:27:90:23:f3:f8:80:
                    5e:e5:a6:73:52:a4:f8:59:c1:0e:e2:18:54:a2:ce:
                    d8:d4:d5:79:98:c0:b7:7b:6a:b4:15:e8:17:48:d2:
                    3a:56:ea:3d:9d:ec:f2:35:2a:a5:88:77:eb:ba:de:
                    65:dc:02:26:cc:0c:d1:6e:c2:9b:ac:07:e0:b8:bc:
                    94:e6:fe:1e:30:77:aa:b7:af:05:1c:e6:4f:94:ea:
                    2c:fa:e6:5b:e1:cd:76:d7:18:1a:72:bd:4d:7e:46:
                    9a:8f:7a:b0:14:f5:e5:f1:e1
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Netscape SSL Server Name:
                www.abylonsoft.de
    Signature Algorithm: sha1WithRSAEncryption
    Signature Value:
        af:e1:b7:cf:a8:66:32:4a:22:89:31:e3:05:5a:89:25:87:9f:
        ad:40:1b:ec:3b:72:b4:e7:22:51:62:58:d9:ed:30:8a:97:5e:
        ac:06:7f:d1:31:5e:20:ce:f7:57:75:ec:80:8b:69:fe:ab:b0:
        4d:c4:a7:d9:43:15:26:fd:f5:89:26:ab:bf:23:0a:fd:e1:0a:
        95:da:2d:e0:b2:f8:91:21:5e:d0:16:68:9e:53:e5:2a:69:5b:
        0a:5e:02:e6:79:a5:47:ed:d5:c0:26:6a:bb:0f:49:b6:ec:3f:
        b3:21:f2:e9:f2:ad:f0:a6:e4:0e:79:e6:63:18:97:6b:aa:6c:
        bd:a4
```

Solution:

- Serial number: 7350616
- Effective date: Apr 24 20:18:42 2008 GMT
- Name (i.e., The Subject): Fred Smith
- Issuer:
- What does CN stand for: Common Name
- What does OU stand for: Organisational Unit
- What does O stand for: Organisation
- What does L stand for: Location

Read about the various attributes here

2) Exploring Digital Certificates in the Internet

`openssl` can also be used to download and inspect certificates from a web server. The command
`openssl s_client -connect x.com:443` will display all certificates used by `x.com`.

Run the tool to inspect the certificate configurations of the following organisations and answer the questions below:

- Which Certificate Authorities (CAs) are used by `microsoft.com`, `x.com`, and `my.gov.au`?
  Note: It is sufficient to inspect the first few lines of the output.

- Are intermediate certificates used? Speculate why.
  Note: Again, the first few lines of the output should be enough.

Tip: After running the command, you may need to stop it manually using `Ctrl+C` (in the command line) or the **Stop** button if using a Jupyter notebook cell.

In [3]:
```
!openssl s_client -connect x.com:443
```

```
Connecting to 172.66.0.227
CONNECTED(00000003)
depth=2 C=US, O=Internet Security Research Group, CN=ISRG Root X1
verify return:1
depth=1 C=US, O=Let's Encrypt, CN=E7
verify return:1
depth=0 CN=x.com
verify return:1
---
Certificate chain
 0 s:CN=x.com
   i:C=US, O=Let's Encrypt, CN=E7
   a:PKEY: EC, (prime256v1); sigalg: ecdsa-with-SHA384
   v:NotBefore: May 28 06:29:59 2026 GMT; NotAfter: Aug 26 06:29:58 2026 G
MT
 1 s:C=US, O=Let's Encrypt, CN=E7
   i:C=US, O=Internet Security Research Group, CN=ISRG Root X1
   a:PKEY: EC, (secp384r1); sigalg: sha256WithRSAEncryption
   v:NotBefore: Mar 13 00:00:00 2024 GMT; NotAfter: Mar 12 23:59:59 2027 G
MT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDwzCCA0mgAwIBAgISBcQ8XddaC/qrc44DYLjb0SpGMAoGCCqGSM49BAMDMDIx
CzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQswCQYDVQQDEwJF
NzAeFw0yNjA1MjgwNjI5NTlaFw0yNjA4MjYwNjI5NThaMBAxDjAMBgNVBAMTBXgu
Y29tMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAECBT65HDzYEoA7EYLwBngIE/6
Q2UlbM7GnLAX7KHdZhrariNInd8CwVWf585JWB2Owe0uJrs4efY5LT+AdctFwaOC
Al8wggJbMA4GA1UdDwEB/wQEAwIHgDATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNV
HRMBAf8EAjAAMB0GA1UdDgQWBBSPjCORvL8EpGL1BheQf3OvsBqGXzAfBgNVHSME
GDAWgBSuSJ7chx1EoG/aouVgdAR4wpwAgDAyBggrBgEFBQcBAQQmMCQwIgYIKwYB
BQUHMAKGFmh0dHA6Ly9lNy5pLmxlbmNyLm9yZy8wXwYDVR0RBFgwVoIMKi5saXZl
LnguY29tgg0qLndhdGNoLnguY29tggcqLnguY29tghVjZG4uc3luZGljYXRpb24u
eC5jb22CEGNsaWNrLnNlbmQueC5jb22CBXguY29tMBMGA1UdIAQMMAowCAYGZ4EM
AQIBMC0GA1UdHwQmMCQwIqAgoB6GHGh0dHA6Ly9lNy5jLmxlbmNyLm9yZy83MS5j
cmwwggELBgorBgEEAdZ5AgQCBIH8BIH5APcAdQDIo8R/x7OtuTVrAT9qehJt4zpO
Q6XGRvmXrTl1mR3PmgAAAZ5tfBdTAAAEAwBGMEQCIHW1RXCzLBC8ggXO1z6DVzGP
aQuf/ythmeM2t1J9poGiAiB8I5OD5b1utxSBW69avm166shcRDlLcJd92E34M2Vx
7QB+AGz+UBlDqF6pFrxS0TPk3Mke8UEcfSWEINFzgJ4YGOs6AAABnm18GfkACAAA
BQANu1aZBAMARzBFAiEA7gYXTjDN153ram5Wkbns3eaO5XfVGwq3K9Uj/RlCpgQC
IFx+8k0+VkffKjLxEejy9dJyfUKUoc0W2rWOyyG4cdueMAoGCCqGSM49BAMDA2gA
MGUCMDokR9h1j8vm/GljlbZAKs6JxfCXkbDnXi7Gue8hLDnA0VmHbfK5xHjzNZDa
stsKuwIxAMUxpuIpSFD7we4oSyx7B+vJF9phxYVkObyNyvcLNAcvzQTmIOe/0C6r
hiAuaHfDGw==
-----END CERTIFICATE-----
subject=CN=x.com
issuer=C=US, O=Let's Encrypt, CN=E7
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: ecdsa_secp256r1_sha256
Negotiated TLS1.3 group: X25519MLKEM768
---
SSL handshake has read 3484 bytes and written 1627 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Protocol: TLSv1.3
Server public key is 256 bit
This TLS version forbids renegotiation.
```

```
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
^C
```

In [5]:
```
!openssl s_client -connect microsoft.com:443
```

```
Connecting to 150.171.109.25
CONNECTED(00000003)
depth=3 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root
G2
verify return:1
depth=2 C=US, O=Microsoft Corporation, CN=Microsoft TLS RSA Root G2
verify return:1
depth=1 C=US, O=Microsoft Corporation, CN=Microsoft TLS G2 RSA CA OCSP 04
verify return:1
depth=0 C=US, ST=WA, L=Redmond, O=Microsoft Corporation, CN=microsoft.com
verify return:1
---
Certificate chain
 0 s:C=US, ST=WA, L=Redmond, O=Microsoft Corporation, CN=microsoft.com
   i:C=US, O=Microsoft Corporation, CN=Microsoft TLS G2 RSA CA OCSP 04
   a:PKEY: RSA, 2048 (bit); sigalg: sha384WithRSAEncryption
   v:NotBefore: Jun 23 09:04:17 2026 GMT; NotAfter: Dec 20 09:04:17 2026 G
MT
 1 s:C=US, O=Microsoft Corporation, CN=Microsoft TLS G2 RSA CA OCSP 04
   i:C=US, O=Microsoft Corporation, CN=Microsoft TLS RSA Root G2
   a:PKEY: RSA, 4096 (bit); sigalg: sha384WithRSAEncryption
   v:NotBefore: Aug  1 20:02:59 2025 GMT; NotAfter: Jun  3 20:02:59 2029 G
MT
 2 s:C=US, O=Microsoft Corporation, CN=Microsoft TLS RSA Root G2
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   a:PKEY: RSA, 4096 (bit); sigalg: sha384WithRSAEncryption
   v:NotBefore: May 21 00:00:00 2025 GMT; NotAfter: Jun 19 23:59:59 2029 G
MT
 3 s:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: Aug  1 12:00:00 2013 GMT; NotAfter: Jan 15 12:00:00 2038 G
MT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIZADCCFuigAwIBAgITQwBnSEqrc7PZSzYHDwAAAGdISjANBgkqhkiG9w0BAQwF
ADBXMQswCQYDVQQGEwJVUzEeMBwGA1UEChMVTWljcm9zb2Z0IENvcnBvcmF0aW9u
MSgwJgYDVQQDEx9NaWNyb3NvZnQgVExTIEcyIFJTQSBDQSBPQ1NQIDA0MB4XDTI2
MDYyMzA5MDQxN1oXDTI2MTIyMDA5MDQxN1owZDELMAkGA1UEBhMCVVMxCzAJBgNV
BAgTAldBMRAwDgYDVQQHEwdSZWRtb25kMR4wHAYDVQQKExVNaWNyb3NvZnQgQ29y
cG9yYXRpb24xFjAUBgNVBAMTDW1pY3Jvc29mdC5jb20wggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCmz046aPoEIwCmrGQN8FTJ2QgG6dcoE4xcd4wVvmp5
Utx8uFy1rz2BaGFK5PBYsXeKEXlpR10+CCF4sLTIeGj8liOa5GHQ8TqlVt4D/h8b
SGxnWaYL/B0X3exO/8pbQZULKB1BySZ9bt43iWYM8qny/jcENi7Jxox7mrpvOm78
8o7hsx+NebzMUrtaP7Q1Peo5Zcz/4sXe0WlJuftGXXl2kvROLvt3rbACZPmb162N
LuSmNkGtGKypEVQiVesOrDIuQdb6PIIAw93e0wi5eeqqOMVcRb7JC30KzlxwUt87
HR5QIj2t9p8dV8MrL2TJuX0zUCJ3SH8YRWPQmBM9WTxxAgMBAAGjghS2MIIUsjCC
AX4GCisGAQQB1nkCBAIEggFuBIIBagFoAHYA1219ENGn9XfCx+lf1wC/+YLJM1pl
4dCzAXMXwMjFaXcAAAGe88Jb0wAABAMARzBFAiBmkexcJZpj74Y8YHTP28rTMy/6
039cInfna0OToeFJ+wIhAM9B6ljx+QywhdNYPsMy+Idq7fkaW/Pb08JEueWqwt0s
AHYAwjF+V0UZo0XufzjespBB68fCIVoiv3/Vta12mtkOUs0AAAGe88JcGAAABAMA
RzBFAiAvmlRY2//rikYKQATzAmtOSfmDaKhjTGuC9OQOa/y49wIhANLOlQvDsteI
1x1IxkPwTY/GncQP4RXWoQzYxYpbgb/lAHYAyKPEf8ezrbk1awE/anoSbeM6TkOl
xkb5l605dZkdz5oAAAGe88JcGAAABAMARzBFAiEAsc50LZ+AM50wuTWOCvYe3YyF
HEzUHoDdA+ZdwVukNGUCIDDcv9FyamJN+FaI/lSWU43/bHmGgw6afShIFa9e7FZp
MBsGCSsGAQQBgjcVCgQOMAwwCgYIKwYBBQUHAwEwPAYJKwYBBAGCNxUHBC8wLQYl
KwYBBAGCNxUIh73XG4Hn60aCgZ0ujtAMh/DaHV2D6cNngrSjTAIBZAIBIDCCAQsG
CCsGAQUFBwEBBIH+MIH7MGEGCCsGAQUFBzAChlVodHRwOi8vd3d3Lm1pY3Jvc29m
dC5jb20vcGtpb3BzL2NlcnRzL01pY3Jvc29mdCUyMFRMUyUyMEcyJTIwUlNBJTIw
```

Q0ElMjBPQ1NQJTIwMDQuY3J0MGcGCCsGAQUFBzAChltodHRwOi8vY2Fpc3N1ZXJz
Lm1pY3Jvc29mdC5jb20vcGtpb3BzL2NlcnRzL01pY3Jvc29mdCUyMFRMUyUyMEcy
JTIwUlNBJTIwQ0ElMjBPQ1NQJTIwMDQuY3J0MC0GCCsGAQUFBzABhiFodHRwOi8v
b25lb2NzcC5taWNyb3NvZnQuY29tL29jc3AwHQYDVR0OBBYEFNBNdhtsyboq5+vN
stdYL79qiJp4MA4GA1UdDwEB/wQEAwIFoDCCD/MGA1UdEQSCD+owgg/mgg1taWNy
b3NvZnQuY29tgg9zLm1pY3Jvc29mdC5jb22CEGdhLm1pY3Jvc29mdC5jb22CEWFl
cC5taWNyb3NvZnQuY29tghFhZXIubWljcm9zb2Z0LmNvbYIRZ3J2Lm1pY3Jvc29m
dC5jb22CEWh1cC5taWNyb3NvZnQuY29tghFtYWMubWljcm9zb2Z0LmNvbYIRbWti
Lm1pY3Jvc29mdC5jb22CEXBtZS5taWNyb3NvZnQuY29tghFwbWkubWljcm9zb2Z0
LmNvbYIRcnNzLm1pY3Jvc29mdC5jb22CEXNhci5taWNyb3NvZnQuY29tghF0Y28u
bWljcm9zb2Z0LmNvbYISZnVzZS5taWNyb3NvZnQuY29tghJpZWFrLm1pY3Jvc29m
dC5jb22CEm1hYzIubWljcm9zb2Z0LmNvbYISbWNzcC5taWNyb3NvZnQuY29tghJv
cGVuLm1pY3Jvc29mdC5jb22CEnNob3AubWljcm9zb2Z0LmNvbYISc3B1ci5taWNy
b3NvZnQuY29tghNpdHByby5taWNyb3NvZnQuY29tghNtYW5nby5taWNyb3NvZnQu
Y29tghNtdXNpYy5taWNyb3NvZnQuY29tghNweW1lcy5taWNyb3NvZnQuY29tghNz
dG9yZS5taWNyb3NvZnQuY29tghRhZXRoZXIubWljcm9zb2Z0LmNvbYIUYWxlcnRz
Lm1pY3Jvc29mdC5jb22CFGRlc2lnbi5taWNyb3NvZnQuY29tghRnYXJhZ2UubWlj
cm9zb2Z0LmNvbYIUZ2lnamFtLm1pY3Jvc29mdC5jb22CFG1zY3RlYy5taWNyb3Nv
ZnQuY29tghRvbmxpbmUubWljcm9zb2Z0LmNvbYIUc3RyZWFtLm1pY3Jvc29mdC5j
b22CFWFmZmxpbmsubWljcm9zb2Z0LmNvbYIVY29ubmVjdC5taWNyb3NvZnQuY29t
ghVkZXZlbG9wLm1pY3Jvc29mdC5jb22CFWRvbWFpbnMubWljcm9zb2Z0LmNvbYIV
ZXhhbXBsZS5taWNyb3NvZnQuY29tghVtYWRlaXJhLm1pY3Jvc29mdC5jb22CFW1z
ZG5pc3YubWljcm9zb2Z0LmNvbYIVbXNwcmVzcy5taWNyb3NvZnQuY29tghV3d3cu
YWVwLm1pY3Jvc29mdC5jb22CFXd3dy5hZXIubWljcm9zb2Z0LmNvbYIVd3d3YmV0
YS5taWNyb3NvZnQuY29tghZidXNpbmVzcy5taWNyb3NvZnQuY29tghZlbXByZXNh
cy5taWNyb3NvZnQuY29tghZsZWFybmluZy5taWNyb3NvZnQuY29tghZtc2Rud2lr
aS5taWNyb3NvZnQuY29tghZvcGVubmVzcy5taWNyb3NvZnQuY29tghZwaW5wb2lu
dC5taWNyb3NvZnQuY29tghZzbmFja2JveC5taWNyb3NvZnQuY29tghZzcG9uc29y
cy5taWNyb3NvZnQuY29tghZzdGF0aW9ucS5taWNyb3NvZnQuY29tghdhaXN0b3Jp
ZXMubWljcm9zb2Z0LmNvbYIXY29tbXVuaXR5Lm1pY3Jvc29mdC5jb22CF2NyYXds
bXNkbi5taWNyb3NvZnQuY29tghdpb3RzY2hvb2wubWljcm9zb2Z0LmNvbYIXbWVz
c2VuZ2VyLm1pY3Jvc29mdC5jb22CF21pbmVjcmFmdC5taWNyb3NvZnQuY29tghhi
YWNrb2ZmaWNlLm1pY3Jvc29mdC5jb22CGGVudGVycHJpc2UubWljcm9zb2Z0LmNv
bYIYaW90Y2VudHJhbC5taWNyb3NvZnQuY29tghhwaW51bmJsb2NrLm1pY3Jvc29m
dC5jb22CGHJlcm91dGU0NDMubWljcm9zb2Z0LmNvbYIZY29tbXVuaXRpZXMubWlj
cm9zb2Z0LmNvbYIZZXhwbG9yZS1zbWIubWljcm9zb2Z0LmNvbYIZZXhwcmVzc2lv
bnMubWljcm9zb2Z0LmNvbYIZb25kZXJuZW1lcnMubWljcm9zb2Z0LmNvbYIZdGVj
aGFjYWRlbXkubWljcm9zb2Z0LmNvbYIZdGVycmFzZXJ2ZXIubWljcm9zb2Z0LmNv
bYIaY29tbXVuaXRpZXMyLm1pY3Jvc29mdC5jb22CGmNvbm5lY3RldmVudC5taWNy
b3NvZnQuY29tghpkYXRhcGxhdGZvcm0ubWljcm9zb2Z0LmNvbYIaZW50cmVwcmVu
ZXVyLm1pY3Jvc29mdC5jb22CGmh4ZC5yZXNlYXJjaC5taWNyb3NvZnQuY29tghpt
c3BhcnRuZXJpcmEubWljcm9zb2Z0LmNvbYIabXlkYXRhaGVhbHRoLm1pY3Jvc29m
dC5jb22CGm9lbWNvbW11bml0eS5taWNyb3NvZnQuY29tghpyZWFsLXN0b3JpZXMu
bWljcm9zb2Z0LmNvbYIad3d3LmZvcm1zcHJvLm1pY3Jvc29mdC5jb22CG2Z1dHVy
ZWRlY29kZWQubWljcm9zb2Z0LmNvbYIbdXBncmFkZWNlbnRlci5taWNyb3NvZnQu
Y29tghxsZWFybmFuYWx5dGljcy5taWNyb3NvZnQuY29tghxvbmxpbmVsZWFybmlu
Zy5taWNyb3NvZnQuY29tgh1idXNpbmVzc2NlbnRyYWwubWljcm9zb2Z0LmNvbYId
Y2xvdWQtaW1tZXJzaW9uLm1pY3Jvc29mdC5jb22CHXN0dWRlbnRwYXJ0bmVycy5t
aWNyb3NvZnQuY29tgh5hbmFseXRpY3NwYXJ0bmVyLm1pY3Jvc29mdC5jb22CHmJ1
c2luZXNzcGxhdGZvcm0ubWljcm9zb2Z0LmNvbYIeZXhwbG9yZS1zZWN1cml0eS5t
aWNyb3NvZnQuY29tgh5rbGVpbnVudGVybmVobWVuLm1pY3Jvc29mdC5jb22CHnBh
cnRuZXJjb21tdW5pdHkubWljcm9zb2Z0LmNvbYIfZXhwbG9yZS1tYXJrZXRpbmcu
bWljcm9zb2Z0LmNvbYIfaW5ub3ZhdGlvbmNvbnRlc3QubWljcm9zb2Z0LmNvbYIf
cGFydG5lcmluY2VudGl2ZXMubWljcm9zb2Z0LmNvbYIfcGhvZW5peGNhdGFsb2d1
YXQubWljcm9zb2Z0LmNvbYIfc3prb2x5cHJ6eXN6bG9zY2kubWljcm9zb2Z0LmNv
bYIfd3d3LnBvd2VyYXV0b21hdGUubWljcm9zb2Z0LmNvbYIgc3VjY2Vzc2lvbnBs
YW5uaW5nLm1pY3Jvc29mdC5jb22CImx1bWlhY29udmVyc2F0aW9uc3VrLm1pY3Jv
c29mdC5jb22CI3N1Y2Nlc3Npb25wbGFubmluZ3VhdC5taWNyb3NvZnQuY29tgiRi
dXNpbmVzc21vYmlsaXR5Y2VudGVyLm1pY3Jvc29mdC5jb22CJXNreXBlYW5kdGVh
bXMuZmFzdHRyYWNrLm1pY3Jvc29mdC5jb22CJ3d3dy5taWNyb3NvZnRkbGFwYXJ0

```
bmVyb3cubWljcm9zb2Z0LmNvbYIoY29tbWVyY2lhbGFwcGNlcnRpZmljYXRpb24u
bWljcm9zb2Z0LmNvbYIpd3d3LnNreXBlYW5kdGVhbXMuZmFzdHRyYWNrLm1pY3Jv
c29mdC5jb22CImNlb2Nvbm5lY3Rpb25zLmV2ZW50Lm1pY3Jvc29mdC5jb22CGGJp
ejRhZnJpa2EubWljcm9zb2Z0LmNvbYIWY2FzaGJhY2subWljcm9zb2Z0LmNvbYIa
d3d3LmNhc2hiYWNrLm1pY3Jvc29mdC5jb22CE3Zpc2lvLm1pY3Jvc29mdC5jb22C
F2luc2lkZW1zci5taWNyb3NvZnQuY29tgh9kZXZlbG9wZXJ2ZWxvY2l0eWFzc2Vz
c21lbnQuY29tgiN3d3cuZGV2ZWxvcGVydmVsb2NpdHlhc3Nlc3NtZW50LmNvbYIK
Z2VhcnM1LmNvbYIOd3d3LmdlYXJzNS5jb22CFHd3dy5nZWFyc3RhY3RpY3MuY29t
ghBnZWFyc3RhY3RpY3MuY29tghFtMTIubWljcm9zb2Z0LmNvbYIMc2VlaW5nYWku
Y29tghh5b3VyY2hvaWNlLm1pY3Jvc29mdC5jb22CGW12dGQuZXZlbnRzLm1pY3Jv
c29mdC5jb22CFWltYWdpbmUubWljcm9zb2Z0LmNvbYIQbWljcm9zb2Z0LmNvbS5h
dYIUd3d3Lm1pY3Jvc29mdC5jb20uYXWCFmR5bmFtaWNzLm1pY3Jvc29mdC5jb22C
G3Bvd2VycGxhdGZvcm0ubWljcm9zb2Z0LmNvbYIXcG93ZXJhcHBzLm1pY3Jvc29m
dC5jb22CG3Bvd2VyYXV0b21hdGUubWljcm9zb2Z0LmNvbYIgcG93ZXJ2aXJ0dWFs
YWdlbnRzLm1pY3Jvc29mdC5jb22CGHBvd2VycGFnZXMubWljcm9zb2Z0LmNvbYIf
dGVzdC5pZGVhcy5mYWJyaWMubWljcm9zb2Z0LmNvbYIRc2RzLm1pY3Jvc29mdC5j
b22CFXBwZS5zZHMubWljcm9zb2Z0LmNvbYIbd3d3Lm1pY3Jvc29mdDM2NWNvcGls
b3QuY29tghB3d3cuamNsYXJpdHkuY29tght0ZWNoaW5ub3ZhdG9yc3Nwb3RsaWdo
dC5jb22CH3d3dy50ZWNoaW5ub3ZhdG9yc3Nwb3RsaWdodC5jb22CCmNvcGlsb3Qu
YWmCFWdldGxpY2Vuc2luZ3JlYWR5LmNvbYIZd3d3LmdldGxpY2Vuc2luZ3JlYWR5
LmNvbYIUanBuLmRlbHZlLm9mZmljZS5jb22CFGF1cy5kZWx2ZS5vZmZpY2UuY29t
ghRpbmQuZGVsdmUub2ZmaWNlLmNvbYIUa29yLmRlbHZlLm9mZmljZS5jb22CFmNv
YnJhLm1lLm1pY3Jvc29mdC5jb22CF3d3dy5idXNpbmVzc2NlbnRyYWwuY29tghNi
dXNpbmVzc2NlbnRyYWwuY29tghxtc2FpZGF0YXN0dWRpby5vZmZpY2VwcGUubmV0
ghppZGVhcy5mYWJyaWMubWljcm9zb2Z0LmNvbYIMd3d3LmNwdC5saW5rgghjcHQu
bGlua4IMeWFycC5kb3QubmV0ghNtaWNyb3NvZnRzdHJlYW0uY29tghd3d3cubWlj
cm9zb2Z0c3RyZWFtLmNvbYIXd2ViLm1pY3Jvc29mdHN0cmVhbS5jb22CE2Rpc2Nv
dmVyLmNvcGlsb3QuYWmCC2NvcGlsb3QuY29tgg93d3cuY29waWxvdC5jb22CFGRp
c2NvdmVyLmNvcGlsb3QuY29tghtyZXNlYXJjaGZvcnVtLm1pY3Jvc29mdC5jb22C
H2Nkbi50ZWNoY29tbXVuaXR5Lm1pY3Jvc29mdC5jb20wDAYDVR0TAQH/BAIwADCB
8QYDVR0fBIHpMIHmMIHjoIHgoIHdhmxodHRwOi8vd3d3Lm1pY3Jvc29mdC5jb20v
cGtpb3BzL2NybC9wYXJ0aXRpb24vTWljcm9zb2Z0JTIwVExTJTIwRzIlMjBSU0El
MjBDQSUyME9DU1AlMjAwNF9QYXJ0aXRpb24wMDAwMS5jcmyGbWh0dHA6Ly9jcmwy
Lm1pY3Jvc29mdC5jb20vcGtpb3BzL2NybC9wYXJ0aXRpb24vTWljcm9zb2Z0JTIw
VExTJTIwRzIlMjBSU0ElMjBDQSUyME9DU1AlMjAwNF9QYXJ0aXRpb24wMDAwMS5j
cmwwZgYDVR0gBF8wXTAIBgZngQwBAgIwUQYMKwYBBAGCN0yDfQEBMEEwPwYIKwYB
BQUHAgEWM2h0dHA6Ly93d3cubWljcm9zb2Z0LmNvbS9wa2lvcHMvRG9jcy9SZXBv
c2l0b3J5Lmh0bTAfBgNVHSMEGDAWgBRUDLzsGPd98QPihL40ZERnz3UfZTATBgNV
HSUEDDAKBggrBgEFBQcDATANBgkqhkiG9w0BAQwFAAOCAgEAABsQp/NAFQ210tL+
liFNrgZZVX8nE2S9V6xioFSxV35knsVdDditONCbFA/JisTDyd/ofJ6PU/08K0yc
lyBIIrpUfCAr3te7iWQnXf397RS+kB79G/wBWb72RBNpfA+6st5xUU4Djk6xwfSB
kaZHEH1v2G5E1ie+ZaOieNtIvbomX4jWNcxvLYaN5rOlsmIz5W36HMBaEQ5lXsYx
H0qJYUoEJdWn4is2hhzYi0J7ehR3n4ousir/5ZeR/MQ69N93xFZdN8Yt8U1henAy
hJegIpg/7yhz0PJTAGwPiyAnb/uW4aKyXe2jHmQZhHcrwQlFvd3h7y1uE6jWxmJS
aFhlPxwr66p3qtNpxcyNoW3fpn/zB2cd6ZFkc4rBVeO5VZfN3ZlVqY9L4y5m62oB
IKJbz4XWgAhaXWq6u6QrL7VDM+1IuIVMkkGdycUxQ5PIfigENmz7rJZez1GA0w8B
nI7s8eA7zLUWhyCiab9cNod1JambBvzsrhw3BSaeJdEe3XY91fKAheIwaS9npoAv
uO9nTjYC7t1TV7ccQf7EAdcc0behWjzcd+8ILNMWYbwk5TjI9DNnuoYu8jXWFtx/
rIYfGe4SrtEMJ6CY26cHP7m2yrpCJtMHwV2Pbl1b0UfKT4kN+cGlZJ56PYNAwlzj
KP7WFm/mz8wjVIlEJZr1STNy8WU=
-----END CERTIFICATE-----
subject=C=US, ST=WA, L=Redmond, O=Microsoft Corporation, CN=microsoft.com
issuer=C=US, O=Microsoft Corporation, CN=Microsoft TLS G2 RSA CA OCSP 04
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: rsa_pss_rsae_sha256
Peer Temp Key: ECDH, prime256v1, 256 bits
---
SSL handshake has read 11420 bytes and written 2003 bytes
```

```
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Protocol: TLSv1.3
Server public key is 2048 bit
This TLS version forbids renegotiation.
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: A1EAB97C6861E0ED33F83E631293A790A865AB37B303F3ACA585DE148C
6F32E7
    Session-ID-ctx:
    Resumption PSK: 88A076F0BD4F535A3A3121FDE2F49D008A8CF7802A6C16F12A556F
398B882430829B37D0B11A3A8CD31026DCDC58B5EB
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 36000 (seconds)
    TLS session ticket:
    0000 - 53 59 44 32 38 2d 30 37-31 33 54 30 30 3a 30 30   SYD28-0713T0
0:00
    0010 - 90 db 1c f8 c0 4d c6 d7-55 5a 2a df e4 e9 3a 82   .....M..UZ
*...:.
    0020 - a5 b6 35 2d e1 f3 18 cb-84 1a d0 af 32 f8 0c 75   ..5-........
2..u
    0030 - 24 98 94 ae c2 8c ed 34-58 f3 43 91 dd af 1e 2b   $......4X.
C....+
    0040 - ea 6c 80 64 d5 62 67 73-c5 05 19 a1 7b 68 d4 a5   .l.d.bgs....
{h..
    0050 - 8b c2 55 f6 2c f1 60 8a-3e dd 3e 89 a1 51 0b 6b   ..U.,.`.>.>..
Q.k
    0060 - b2 30 af d7 d1 e1 6d d6-4b c5 3d cd 56 88 81 c0   .0....m.K.=.
V...
    0070 - e6 40 d3 fc 82 9a ac 1f-05 9c 35 45 32 7a 92 f6   .@........5E2
z..
    0080 - 74 32 e2 d7 27 40 bf 20-d7 d4 4b 4f c6 9a 19 95   t2..'@. ..K
O....
    0090 - ac d7 9a 51 66 60 da a7-40 14 ab 55 5d 10 cd bd   ...Qf`..@..
U]...
    00a0 - 27 8e 1a 3c 1b d4 39 66-67 c6 93 85 e8 e1 91 53   '..<..9f
g......S
    00b0 - 48 53 04 62 7f 95 ba 80-e0 10 81 db a8 f2 59 09   HS.
b..........Y.
    00c0 - fa 53 9c d9 33 59 b8 50-1e 36 18 2b 65 28 f1 52   .S..3Y.P.6.+e
(.R
    00d0 - 36 d9 f8 ea 05 70 bd 97-7c 16 98 1f 19 1a a6 c6   6....p..
|.......
    00e0 - 1f d4 29 76 8e 7b 50 a4-08 02 ab fb 9c 8f 0e 9d   ..)v.
{P.........

    Start Time: 1783915368
    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: 80CF80452ED4F0E27823C15BF4EE49B1159DAAE9B0115B525F2C870D15
61C386
    Session-ID-ctx:
    Resumption PSK: 2FB14A3981B378ACDC8C7FA55E87619FC1FA30128A6BD293203B63
C32659E95E61EFB41029713347C3EDA513EDA394EC
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 36000 (seconds)
    TLS session ticket:
    0000 - 53 59 44 32 38 2d 30 37-31 33 54 30 30 3a 30 30   SYD28-0713T0
0:00
    0010 - 79 61 c7 e3 74 e3 c1 b0-c6 c6 39 9a b0 3c 0f 73   ya..t.....9..
<.s
    0020 - 5a 6f 19 b9 e0 33 d8 a6-47 cc 9b c3 a4 04 48 21   Zo...3..
G.....H!
    0030 - f1 95 a1 02 7d 95 d7 30-ad 98 b6 f9 b6 5c 7a f0   ....}..
0.....\z.
    0040 - 3f 16 ac ec a8 7d d4 01-30 e8 54 eb c0 7b d2 fa   ?....}..0.T..
{..
    0050 - 33 46 b6 50 aa 13 77 48-9b 9d 9a ac fb 28 ec 62   3F.P..wH.....
(.b
    0060 - c8 db cf ba 8d 40 7d 24-82 30 fd f1 f0 82 62 97   .....@}$.
0....b.
    0070 - d3 57 d1 92 1f b5 36 31-7a 40 e4 68 e5 5c fa 8e   .W....61z@.
h.\..
    0080 - 83 83 b5 ba 1a b6 21 2c-ed ea 9b b7 cb 7c 1c bc   ......!,.....
|..
    0090 - 56 3b d6 6e 0c 1b 5e 22-05 49 e8 0d 99 9f 0e 40   V;.n..^".
I.....@
    00a0 - 45 9c a3 6f 09 ba 44 e7-5a 55 47 ac 91 e7 42 1c   E..o..D.ZU
G...B.
    00b0 - 9d ef 02 10 66 ab 16 41-43 9a aa 46 7f 32 d2 a9   ....f..AC..F.
2..
    00c0 - eb aa 1b 69 7f b4 ed 88-86 23 2c dc 77 19 f3 97   ...i.....#,.
w...
    00d0 - df b0 3f 91 f4 f5 85 67-59 13 74 11 4e 2c e9 65   ..?....gY.t.
N,.e
    00e0 - db e7 9d 52 bb 00 59 ec-1d e3 54 06 d2 0a 77 f5   ...R..Y...
T...w.

    Start Time: 1783915368
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
closed
```

In [6]:
```
!openssl s_client -connect my.gov.au:443
```

```
Connecting to 3.175.115.9
CONNECTED(00000003)
depth=2 C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root
G2
verify return:1
depth=1 C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA
1
verify return:1
depth=0 C=AU, ST=Australian Capital Territory, L=Greenway, O=Services Aust
ralia, CN=*.my.gov.au
verify return:1
---
Certificate chain
 0 s:C=AU, ST=Australian Capital Territory, L=Greenway, O=Services Austral
ia, CN=*.my.gov.au
   i:C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: May 21 00:00:00 2026 GMT; NotAfter: Dec  5 23:59:59 2026 G
MT
 1 s:C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: Mar 30 00:00:00 2021 GMT; NotAfter: Mar 29 23:59:59 2031 G
MT
 2 s:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   i:C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root G2
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: Aug  1 12:00:00 2013 GMT; NotAfter: Jan 15 12:00:00 2038 G
MT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIHCzCCBfOgAwIBAgIQA0Sxjbab0N1LhZdPsdmq7zANBgkqhkiG9w0BAQsFADBZ
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMTMwMQYDVQQDEypE
aWdpQ2VydCBHbG9iYWwgRzIgVExTIFJTQSBTSEEyNTYgMjAyMCBDQTEwHhcNMjYw
NTIxMDAwMDAwWhcNMjYxMjA1MjM1OTU5WjB6MQswCQYDVQQGEwJBVTElMCMGA1UE
CBMcQXVzdHJhbGlhbiBDYXBpdGFsIFRlcnJpdG9yeTERMA8GA1UEBxMIR3JlZW53
YXkxGzAZBgNVBAoTElNlcnZpY2VzIEF1c3RyYWxpYTEUMBIGA1UEAwwLKi5teS5n
b3YuYXUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCJ8Z9jScWPaz8X
7eHQyNKAdlryAZL4tqc3/wvRTIQpa8PIhT7qjiCAxA+zTwLV3St4PttmN/3h1PAh
2OW0EEqQh/hAeYGaB1ceHt6nQa4FWps+LnO8aQXdPzI4N7XpsrOC8QD0QJaMXQm0
wluvM3W1ejTXWDgFcR76hpn8didg9LxVoSUGoBYLigjv3y2pFka9r2wKgFw9Jm9Q
Ums1UyfJLUGf4gKCbkKWj4AP02SgKWU4984Q1ESkByoUswyMCOWIFPAV/g16YvNy
JJztXXNLKlb6np5NimEXaXoLZy/OgQn+Cf5Mm/UGTO32+0L/8JKl7R0K53sHHLsC
WlH3jHErAgMBAAGjggOsMIIDqDAfBgNVHSMEGDAWgBR0hYDAZsffN97PvSk3qgMd
vu3NFzAdBgNVHQ4EFgQUBgDka4hfbjayE20OWJxsyhErBoMwRwYDVR0RBEAwPoIL
Ki5teS5nb3YuYXWCEGF1dGhvci5teS5nb3YuYXWCEiouYXV0aG9yLm15Lmdvdi5h
dYIJbXkuZ292LmF1MD4GA1UdIAQ3MDUwMwYGZ4EMAQICMCkwJwYIKwYBBQUHAgEW
G2h0dHA6Ly93d3cuZGlnaWNlcnQuY29tL0NQUzAOBgNVHQ8BAf8EBAMCBaAwEwYD
VR0lBAwwCgYIKwYBBQUHAwEwgZ8GA1UdHwSBlzCBlDBIoEagRIZCaHR0cDovL2Ny
bDMuZGlnaWNlcnQuY29tL0RpZ2lDZXJ0R2xvYmFsRzJUTFNSU0FTSEEyNTYyMDIw
Q0ExLTEuY3JsMEigRqBEhkJodHRwOi8vY3JsNC5kaWdpY2VydC5jb20vRGlnaUNl
cnRHbG9iYWxHMlRMU1JTQVNIQTI1NjIwMjBDQTEtMS5jcmwwgYcGCCsGAQUFBwEB
BHsweTAkBggrBgEFBQcwAYYYaHR0cDovL29jc3AuZGlnaWNlcnQuY29tMFEGCCsG
AQUFBzAChkVodHRwOi8vY2FjZXJ0cy5kaWdpY2VydC5jb20vRGlnaUNlcnRHbG9i
YWxHMlRMU1JTQVNIQTI1NjIwMjBDQTEtMS5jcnQwDAYDVR0TAQH/BAIwADCCAXwG
CisGAQQB1nkCBAIEggFsBIIBaAFmAHUAwjF+V0UZo0XufzjespBB68fCIVoiv3/V
ta12mtkOUs0AAAGeSVogtQAABAMARjBEAiApU7Mrb0fGYDXqDbJXAM5eHOq8LYVg
m/MNRNjHX6cSfQIgK4kq1O9p841hse0IFCaFGBhqwxpAgJ/vuGme0IQpTxYAdgDX
bX0Q0af1d8LH6V/XAL/5gskzWmXh0LMBcxfAyMVpdwAAAZ5JWiEwAAAEAwBHMEUC
```

```
IQC3hh83+qisJYDNEinivZy6klwYG3YqhJE6J4SiTuQkLAIgemO0DOQZX6J5TwZW
y6rQDuYwD3VbpNq0NCTjfktuLGoAdQCUTkOH+uzB74HzGSQmqBhlAcfTXzgCAT9y
Z31VNy4Z2AAAAZ5JWiCpAAAEAwBGMEQCIDjJI/feRLiFAKYqpJI/glX7vAzBO14O
FC+cEznWN6LkAiBm9zR3Tfyu52OuaZT7ou91NnA7Oy4muKZdMxQ8tk+8cTANBgkq
hkiG9w0BAQsFAAOCAQEAtEDjTPCZiepPe7lw2xB9d+o9C51Dzol1yhN77U0a3vQB
aZeR9nw4vLFUe5/l6R6C7UZPLo8df8D3sgevEyujnoBfXiz2WHY/161RDaKNrE0x
ZWH4IlrDpEPYtlNBOv918wFcuWShUs8u9ZjV1nksnZJv1pTYckXQ1y8Ka/joo7sC
P+pkgrdWLofxouTdD1fx6OMKfmnEkplR2R3yvvD1NLaMRlRwx7atiAMFDJ5ChjVS
tMwvLdgVO3VTJLZbOK5Fm8r1M9rOvLy5iga1qT/J9T1GFZf68RNTDyzO8p17+ght
yjSWto1D8A4FcSYpPZmf1iCn5OtOhvKpz0siLFF3jw==
-----END CERTIFICATE-----
subject=C=AU, ST=Australian Capital Territory, L=Greenway, O=Services Aust
ralia, CN=*.my.gov.au
issuer=C=US, O=DigiCert Inc, CN=DigiCert Global G2 TLS RSA SHA256 2020 CA1
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: rsa_pss_rsae_sha256
Negotiated TLS1.3 group: X25519MLKEM768
---
SSL handshake has read 5591 bytes and written 1615 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_128_GCM_SHA256
Protocol: TLSv1.3
Server public key is 2048 bit
This TLS version forbids renegotiation.
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_128_GCM_SHA256
    Session-ID: 52D383689556CDEDB3C8101728F2EFF36344897A15DF8ECAE0B00D3E33
B0FD22
    Session-ID-ctx:
    Resumption PSK: C03297E6A930A3554FAEF47C2A3CB25961686E0A1AD9619F037D14
8848B85372
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 78512 (seconds)
    TLS session ticket:
    0000 - 01 31 37 38 33 39 30 37-34 39 34 30 30 30 00 00   .178390749400
0..
    0010 - 00 6c 33 6a 9d 24 89 7f-7a 09 0d 65 91 1e 8e c5   .l3j.$..z..
e....
    0020 - f9 0f 06 d3 2b 94 68 86-24 c1 ad d6 17 4c 56 2a   ....+.h.$....
LV*
    0030 - 47 6c 2f 6b 07 13 f1 21-b0 b1 96 04 db 3c 90 3b   Gl/k...!.....
<.;
    0040 - 6d e4 1f 97 c8 47 3e d0-78 8c 23 69 d5 4d 45 56   m....G>.x.#i.
MEV
    0050 - 20 8f 19 2e e4 c9 13 39-fc cc 65 6b 24 ed 07 b8    ......9..ek
$...
```

```
    0060 - e3 de 6b c7 c1 e4 a9 eb-8a 25 f4 78 01 7d 84 4b   ..k......%.
x.}.K
    0070 - 1b f9 a7 7d 3b 1c aa c8-04 f1 77 17 50 2b 41 de   ...};.....w.P
+A.
    0080 - 03 5f 52 73 f0 66 24 30-60 58                     ._Rs.f$0`X

    Start Time: 1783915381
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
^C
```

Solution:

These answers may vary.

CAs

- x.com – Let's Encrypt
- microsoft.com - Microsoft Azure RSA TLS Issuing CA 08
- my.gov.au - Entrust OV

To print the Certificate Authority (CA) of a website's certificate, you can use the following command:

`!openssl s_client -connect x.com:443 -showcerts </dev/null 2>/dev/null | openssl x509 -noout -issuer`

Solution:

Yes, this is now enforced by standards. The main reason is to keep the main CA private key offline, and in day-to-day use, work with the intermediate certificate. This helps prevent the loss of the main private key in case of a CA compromise.

For example, for x.com there are multiple levels:

- Level 0: Leaf certificate issued to `x.com`
- Level 1: Intermediate certificate – Let's Encrypt
- Level 2: Root certificate – ISRG Root X1

In [13]:
```
!openssl s_client -connect x.com:443 -showcerts </dev/null 2>/dev/null |
```

```
CN=x.com
error 20 at 0 depth lookup: unable to get local issuer certificate
error stdin: verification failed
```

3) Issues in Certificates

One possible cause of the infamous invalid certificate error is a mismatch between the domain name listed in the certificate and the actual domain name of the website.

Using your browser, try to visit:

- http://testfire.net/
- https://testfire.net/

(This is a test website maintained by IBM for demonstrating common web application vulnerabilities and website defects.)

The first connection is not encrypted, so you will be able to visit the website without any issues. However, the second attempt uses encryption, and you will receive a certificate error: `NET::ERR_CERT_COMMON_NAME_INVALID`.

a) Use openssl and try to find the problem

In [ ]:
```
!openssl s_client -connect testfire.net:443
```

```
Connecting to 65.61.137.117
CONNECTED(00000003)
depth=2 C=US, ST=New Jersey, L=Jersey City, O=The USERTRUST Network, CN=US
ERTrust RSA Certification Authority
verify return:1
depth=1 C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sect
igo RSA Domain Validation Secure Server CA
verify return:1
depth=0 CN=demo.testfire.net
verify error:num=10:certificate has expired
notAfter=Jun 21 23:59:59 2026 GMT
verify return:1
depth=0 CN=demo.testfire.net
notAfter=Jun 21 23:59:59 2026 GMT
verify return:1
---
Certificate chain
 0 s:CN=demo.testfire.net
   i:C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo
RSA Domain Validation Secure Server CA
   a:PKEY: RSA, 2048 (bit); sigalg: sha256WithRSAEncryption
   v:NotBefore: May 21 00:00:00 2025 GMT; NotAfter: Jun 21 23:59:59 2026 G
MT
 1 s:C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo
RSA Domain Validation Secure Server CA
   i:C=US, ST=New Jersey, L=Jersey City, O=The USERTRUST Network, CN=USERT
rust RSA Certification Authority
   a:PKEY: RSA, 2048 (bit); sigalg: sha384WithRSAEncryption
   v:NotBefore: Nov  2 00:00:00 2018 GMT; NotAfter: Dec 31 23:59:59 2030 G
MT
 2 s:C=US, ST=New Jersey, L=Jersey City, O=The USERTRUST Network, CN=USERT
rust RSA Certification Authority
   i:C=GB, ST=Greater Manchester, L=Salford, O=Comodo CA Limited, CN=AAA C
ertificate Services
   a:PKEY: RSA, 4096 (bit); sigalg: sha384WithRSAEncryption
   v:NotBefore: Mar 12 00:00:00 2019 GMT; NotAfter: Dec 31 23:59:59 2028 G
MT
 3 s:C=GB, ST=Greater Manchester, L=Salford, O=Comodo CA Limited, CN=AAA C
ertificate Services
   i:C=GB, ST=Greater Manchester, L=Salford, O=Comodo CA Limited, CN=AAA C
ertificate Services
   a:PKEY: RSA, 2048 (bit); sigalg: sha1WithRSAEncryption
   v:NotBefore: Jan  1 00:00:00 2004 GMT; NotAfter: Dec 31 23:59:59 2028 G
MT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIGLDCCBRSgAwIBAgIQCFC26PmXdevJdonrg/txbTANBgkqhkiG9w0BAQsFADCB
jzELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEYMBYGA1UEChMPU2VjdGlnbyBMaW1pdGVkMTcwNQYDVQQD
Ey5TZWN0aWdvIFJTQSBEb21haW4gVmFsaWRhdGlvbiBTZWN1cmUgU2VydmVyIENB
MB4XDTI1MDUyMTAwMDAwMFoXDTI2MDYyMTIzNTk1OVowHDEaMBgGA1UEAxMRZGVt
by50ZXN0ZmlyZS5uZXQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCR
7mJPxC2MMpWbIgFA2g8r91LVX52B4Ir9I4KYLXhFC89+baLgaIilMuRmzrF2P55y
zSlJHA57URJ8Hx00wlljwbkCk3T/UqUiY0/ksQz4pzHa+B05M26tDZtB4Fg85Sin
iypiDh6A7O52zwSQ4YU165Nb6K1hA0SLe+m63/kLKWwcIDZUS8phZE+IKcZzsBO8
XLQVVf6zNtGg1/1GTPEJLOGdriRd9QNnfkU3kyRSmTex92mGYxfGLvATLlY+botR
G+b8Yi3NibLDs34Y43Oy463TzFwmvD2nRZQyb1OXpaj0y4x/K6p59i3AesZ4FITD
/tLDAN6e4DrKQ8aPjtBlAgMBAAGjggL0MIIC8DAfBgNVHSMEGDAWgBSNjF7EVK2K
4Xfpm/mbBeG4AY1h4TAdBgNVHQ4EFgQUD+bm67/wgO329JuDKpo8icXfEewwDgYD
```

```
VR0PAQH/BAQDAgWgMAwGA1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEG
CCsGAQUFBwMCMEkGA1UdIARCMEAwNAYLKwYBBAGyMQECAgcwJTAjBggrBgEFBQcC
ARYXaHR0cHM6Ly9zZWN0aWdvLmNvbS9DUFMwCAYGZ4EMAQIBMIGEBggrBgEFBQcB
AQR4MHYwTwYIKwYBBQUHMAKGQ2h0dHA6Ly9jcnQuc2VjdGlnby5jb20vU2VjdGln
b1JTQURvbWFpblZhbGlkYXRpb25TZWN1cmVTZXJ2ZXJDQS5jcnQwIwYIKwYBBQUH
MAGGF2h0dHA6Ly9vY3NwLnNlY3RpZ28uY29tMIIBfwYKKwYBBAHWeQIEAgSCAW8E
ggFrAWkAdwCWl2S/VViXrfdDh2g3CEJ36fA61fak8zZuRqQ/D8qpxgAAAZbzE8bg
AAAEAwBIMEYCIQCYhdWq5m8sBYpWr58ItIC4Fyr3jcAeJVHj+SvEX3mapwIhAPfb
XiymVcy9/ysVWisYOU4dfBvcECLJJ0CtkUwBCNY0AHYAGYbUxyiqb/66A294Kk0B
karOLXIxD67OXXBBLSVMx9QAAAGW8xPGigAABAMARzBFAiA3Xh4AYvc99nfmXOtS
Ms2S1VbHWg7g0zWGfJ1kRh1UQQIhAIpaL1Oo+CU4qh3varw/LQOEYk8scAbqO7GA
UbYW37+AAHYADleUvPOuqT4zGyyZB7P3kN+bwj1xMiXdIaklrGHFTiEAAAGW8xPG
igAABAMARzBFAiBOfqOUPhhd9pi8T0/96VAfVmUs8l7LAQXr3hxhzQoucAIhAIX2
g17jbpsnWTp4ImDmzuVAKDqzpMUl8qcS7U7IztlBMBwGA1UdEQQVMBOCEWRlbW8u
dGVzdGZpcmUubmV0MA0GCSqGSIb3DQEBCwUAA4IBAQAcS7hS4Cg/m4LwdTjwydMl
o9bPEpcKxJcL6lMB06eK3qSKp0cvDSWLBRFxAM9PaxjKTzKuTBjExN1oYaTj2XN6
Z+qwKPgjS4gVuKV2bW1r9VjnL6KMcmQ+/Tp5D83duJG7hZjw+E6HjV1Ml0jdJmGN
SgxtYV6lrA4PNYA5zMTFYBr7ijLPANp7mwFpOhcF9B+1wO3gO5J7PTRKmEO9GjA6
3DEaVHrDH6onmoEDCfiCHFrLlr6xCtw44EbY9HDVw2mM8GFcKgrfQuEJjqW0JEpP
+caySon48gkx/FjOwrZEdxPajGZ7TTJgYYOZzDUorWfuZEbzFRtlqA7iWVlnHfK9
-----END CERTIFICATE-----
subject=CN=demo.testfire.net
issuer=C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Secti
go RSA Domain Validation Secure Server CA
---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: rsa_pkcs1_sha256
Peer Temp Key: ECDH, prime256v1, 256 bits
---
SSL handshake has read 6131 bytes and written 1680 bytes
Verification error: certificate has expired
---
New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Protocol: TLSv1.2
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-AES256-GCM-SHA384
    Session-ID: 6A547DD78622B2BCFB1FC25966AEDF0350297FF73E724196910F912789
FE1C9C
    Session-ID-ctx:
    Master-Key: 3CDC49D76EE0FC13AEF88D518939334A66BAB498024D11A1A012EB5AC2
D5F2688426543B94AE5FD267255D1E56E371B1
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1783922135
    Timeout   : 7200 (sec)
    Verify return code: 10 (certificate has expired)
    Extended master secret: yes
---
```

Solution:

> You will see the subject in the certificate is `demo.testfire.net`, which is different from `testfire.net` that we are trying to visit. Such mismatches are not allowed by modern browsers.

b) Use SSL Labs to Verify Your Answer

SSL Labs is a useful tool for troubleshooting SSL certificate issues. Try running a scan for https://testfire.net and review the results.

Then, run a scan for another site such as www.sydney.edu.au for comparison.

> **Solution:**
>
> You will see SSL complaining about the same issue for `testfire.net`. The scan for `www.sydney.edu.au` may take some time, but it will not generate any errors.

Verifying the Certificate Chain of Trust with OpenSSL

Before we verify the certificate chain using `openssl`, it's important to understand what the **chain of trust** means in the context of TLS/SSL certificates.

🔐 What is a Certificate Chain of Trust?

When you connect securely to a website (e.g., `https://example.com`), your browser or client needs to verify that the server's TLS certificate is **authentic** and **issued by a trusted authority**.

This verification doesn't happen in isolation. Instead, it relies on a *chain of trust* that connects the server's certificate to a **trusted root certificate authority (CA)** that your system already knows about.

The chain typically includes:

1. **Leaf certificate (Server certificate)** – belongs to the website (e.g., `x.com`)
2. **Intermediate certificate(s)** – issued by a trusted root CA, used to sign the server certificate
3. **Root certificate** – issued by a trusted Certificate Authority, pre-installed on your system

If any certificate in the chain is missing or invalid, verification will fail.

🧪 Using OpenSSL to Verify the Chain of Trust

We can use OpenSSL tools to manually inspect and verify the certificate chain returned by a server.

We'll perform the following:

1. **Fetch the full certificate chain from the server**
2. **Extract the certificates**
3. **Use OpenSSL's `verify` command to validate the chain**

We'll see this in action using the domain `x.com` in the steps below.

4) Use `openssl` to verify the chain of trust for `x.com`

You will need to locate the root certificate store on your Ubuntu system.

> **Solution:**
>
> You can use the following command to connect to `x.com` and retrieve the full certificate chain:
>
> `openssl s_client -connect testfire.net:443 -CApath /etc/ssl/certs -showcerts`
>
> For verified certificate chains you will see the output `Verify return code: 0 (ok)`

5) Expired Certificates

Since September 2020, major browser vendors such as Google, Apple, and Microsoft have enforced stricter rules regarding certificate validity. Publicly rooted digital certificates that expire more than 13 months (or 398 days) from their creation date are no longer accepted in modern web browsers. This marked a significant shift from earlier practices, where certificates were often valid for 8–10 years. The move was intended to enhance security across the web ecosystem.

a) Explain how this policy improves the security of websites and users.

b) What practical or usability challenges may have arisen during the early stages of this transition?

> **Solution:**
>
> **a)** Reduces the possible exploitation window of compromised certificates; acts as a partial solution to the broken certificate revocation process; allows certificate owners to upgrade to stronger encryption algorithms and phase out weaker ones more frequently.
>
> **b)** Many organizations without automated certificate renewal mechanisms may face service outages. Frequent certificate maintenance can become tedious,

> especially in IoT environments where there are a large number of distributed devices.

5) Homework 📚

Please read about the following two concepts:

- **Self-Signed Certificates**
- **SSL Proxies**

> **Solution:**
>
> **Self-Signed Certificates**
> A self-signed certificate is one that is not signed by any CA—neither private nor public. Instead, the certificate is signed with its own private key, rather than being issued by a public or private CA. Normally, a certificate's subject public key is signed by the CA's private key. In a self-signed certificate, the subject's public key is signed by its own private key. Self-signed certificates are useful for testing and within internal networks.
>
> Learn more about self-signed certificate risks
>
> ---
>
> **SSL Proxies**
> SSL proxies use a form of Man-in-the-Middle (MITM) attack where the administering organization installs a trusted certificate on devices within their network.
>
> More about SSL proxies and HTTPS filtering

In [ ]: