Activity V : Public Key Infrastructure

By Saenyakorn Siangsanoh 6232035721 สามารถดู Resource เต็ม ๆ ได้ที่ 2110413-COMP-SECURITY Activity 6

Table of Contents

Question 1

From the two given openssl commands, what is the difference?

Answer

เนื่องจากคอมของผู้ใช้เป็น MacOS Monterey version 12.7 จึงทำให้เวลา openssl หา cert ไม่เจอมันจะแอบไปหาจาก Keychain ทำให้ verify ok ตลอด

Question 2

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

Answer

อาจจะแปลได้หลายความหมายเช่น

- certificate อาจจะเป็น self-signed
- certificate อาจจะไม่ได้ sign โดย CA ที่เรา trust

Question 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).

Answer

ผลลัพธ์ที่ได้

```
Certificate:
    Data:
```

```
        Version: 3 (0x2)
        Serial Number:
            04:7c:20:d4:45:96:b4:97:87:65:bd:84:ef:5e:2e:76
    Signature Algorithm: ecdsa-with-SHA384
        Issuer: C=US, O=DigiCert Inc, CN=DigiCert TLS Hybrid ECC SHA384
2020 CA1
        Validity
            Not Before: Feb  2 00:00:00 2022 GMT
            Not After : Feb  1 23:59:59 2023 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:
                    04:7f:c0:c6:83:a5:e8:f2:9b:bd:bb:97:3b:b6:bc:
                    4c:73:8e:23:33:98:31:ab:96:9e:ea:b6:05:0b:88:
                    77:2b:c5:64:d2:24:ec:46:16:f5:a0:c6:67:12:fd:
                    62:69:9d:cc:e7:87:68:1a:c5:1f:10:8e:b9:20:14:
                    67:11:25:bf:b9
                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:

01:59:A3:CA:D4:41:E2:D8:40:BD:F9:8C:8B:13:C8:76:76:2D:0C:34
            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: critical
                CA:FALSE
            1.3.6.1.4.1.11129.2.4.2:

...j.h.w.....|.....=..>.j.g)]...$...4.......~.........H0F.!...8...c0I9.,LU
..../...c.........x.!..#.l..;...@B.....W..r.U..@ ......v.5.....lW...LmB...'
&Q.?.*....;.L...~.........G0E. B.v.../...S.N.?
w..!...j-.*F..*Q&=Xwa..q...l..Q.&..+..u..sw...P.c.......Jy-.g.......y6....
.~.........F0D. r.:g..YuQ.N+e......"....F....dH7. 4..~I>....C...a
...!S...
    Signature Algorithm: ecdsa-with-SHA384
         30:66:02:31:00:f8:7e:04:ca:41:96:06:08:a0:b6:69:2c:31:
         d2:2a:f4:6b:fa:45:5e:25:1b:6c:29:fe:3c:b0:c3:9e:96:83:
         47:f0:6e:e0:8d:3c:9f:07:7e:39:72:af:d2:ad:c2:67:64:02:
         31:00:a5:40:f0:aa:9b:a3:b8:07:70:44:d4:de:5b:e2:c6:17:
         e0:cd:a4:b2:4f:e8:95:90:63:c4:d9:1d:5e:7c:79:7f:cf:57:
         8b:e9:15:6d:e2:40:cb:c7:04:68:41:fa:59:39
-----BEGIN CERTIFICATE-----
MIIFcTCCBPagAwIBAgIQBHwg1EWWtJeHZb2E714udjAKBggqhkjOPQQDAzBWMQsw
CQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMTAwLgYDVQQDEydEaWdp
Q2VydCBUTFMgSHlicmlkIEVDQyBTSEEzODQgMjAyMCBDQTEwHhcNMjIwMjAyMDAw
MDAwWhcNMjMwMjAxMjM1OTU5WjBoMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2Fs
aWZvcm5pYTEWMBQGA1UEBxMNU2FuIEZyYW5jaXNjbzEWMBQGA1UEChMNVHdpdHRl
ciwgSW5jLjEUMBIGA1UEAxMLdHdpdHRlci5jb20wWTATBgcqhkjOPQIBBggqhkjO
PQMBBwNCAAR/wMaDpejym727lzu2vExzjiMzmDGrlp7qtgULiHcrxWTSJOxGFvWg
xmcS/WJpncznh2gaxR8QjrkgFGcRJb+5o4IDkjCCA44wHwYDVR0jBBgwFoAUCrwI
KReMpTlteg7OM8cus+37w3owHQYDVR0OBBYEFAFZo8rUQeLYQL35jIsTyHZ2LQw0
MCcGA1UdEQQgMB6CC3R3aXR0ZXIuY29tgg93d3cudHdpdHRlci5jb20wDgYDVR0P
AQH/BAQDAgeAMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjCBmwYDVR0f
BIGTMIGQMEagRKBChkBodHRwOi8vY3JsMy5kaWdpY2VydC5jb20vRGlnaUNlcnRU
TFNIeWJyaWRFQ0NTSEEzODQyMDIwQ0ExLTEuY3JsMEagRKBChkBodHRwOi8vY3Js
NC5kaWdpY2VydC5jb20vRGlnaUNlcnRUTFNIeWJyaWRFQ0NTSEEzODQyMDIwQ0Ex
LTEuY3JsMD4GA1UdIAQ3MDUwMwYGZ4EMAQICMCkwJwYIKwYBBQUHAgEWG2h0dHA6
Ly93d3cuZGlnaWNlcnQuY29tL0NQUzCBhQYIKwYBBQUHAQEEeTB3MCQGCCsGAQUF
BzABhhhodHRwOi8vb2NzcC5kaWdpY2VydC5jb20wTwYIKwYBBQUHMAKGQ2h0dHA6
Ly9jYWNlcnRzLmRpZ2ljZXJ0LmNvbS9EaWdpQ2VydFRMU0h5YnJpZEVDQ1NIQTM4
NDIwMjBDQTEtMS5jcnQwDAYDVR0TAQH/BAIwADCCAX4GCisGAQQB1nkCBAIEggFu
BIIBagFoAHcArfe++nz/EMiLnT2cHj4YarRnKV3PsQwkyoWGNOvcgooAAAF+vB6h
AAAABAMASDBGAiEA19g4sdiGYzBJOd4sTFWGzbz7LxaEtmMWG/K34bUGHngCIQCe
I+Zs6Rc78Z3FQEKRx6AGr1e0gHLfVZi0QCAuwpTbfwB2ADXPGRu/sWxXvw+tTG1C
y7u2JyAmUeo/4SrvqAPDO9ZMAAABfrweoNMAAAQDAEcwRQIgQvl2vdwUL8IR7xI6
U/W9Xy7MHdOFDRFIElOhTu4/d5QCIQDw6Wot/ipG5/kqUSY9WHdhvPdxGYmlbBmo
URomLgkr1AB1ALNzdwfhhFD4Y4bWBancEQlKeS2xZwwLh9zwAw55NqWaAAABfrwe
oPEAAAQDAEYwRAIgcp86Z42gWXVRm04rZdnq5g8WpiK3jcUbRpfWs/lkSDcCIDQS
AX5JPvuQ/KVDzYmkYQ2+1QoPNVLMDRCcCSFTjBgLMAoGCCqGSM49BAMDA2kAMGYC
MQD4fgTKQZYGCKC2aSwx0ir0a/pFXiUbbCn+PLDDnpaDR/Bu4I08nwd+OXKv0q3C
Z2QCMQClQPCqm6O4B3BE1N5b4sYX4M2ksk/olZBjxNkdXnx5f89Xi+kVbeJAy8cE
aEH6WTk=
-----END CERTIFICATE-----
```

Field ที่สำคัญได้แก่

- Authority Information Access - เป็นข้อมูลที่บอกว่า certificate นี้ออกโดยใคร
- X509v3 Subject Alternative Name - เป็นข้อมูลที่บอกว่า certificate นี้ใช้กับ domain ใดบ้าง
- Signature Algorithm - เป็นข้อมูลที่บอกว่า certificate นี้เข้ารหัสด้วยวิธีใด
- Validity คือ วันและเวลาที่เรายังสามารถใช้ certificate นี้ได้

Question 4

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

Answer

เพื่อไม่ให้ Root CA ได้ติดต่อกับ Client โดยตรง เพื่อไม่ให้ Privte Key ของ Root CA หลุดออกไป

Question 5

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

Answer

เมื่อลองสำรวจผลลัพธ์ที่ได้ในข้อ 3 จะสังเกตว่า Authority Information Access มี CA Issuer คือ http://cacerts.digicert.com/DigiCertTLSHybridECCSHA3842020CA1-1.crt

เมื่อลอง inspect โดยใช้ certificate ด้านบนจะพบว่า CA Issuer เป็น Digicert Root CA แล้ว

ดังนั้นหมายความว่ามี intermediate CA เพียงแค่หนึ่งเดียว

Question 6

What is the role of ca-certificates.crt?

Answer

ไฟล์ที่ทำหน้าที่เก็บ certificate ของ root CA ทั้งหมดที่เราเชื่อ

Question 7

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

Answer

ใช้ command นี้เพื่อนับจำนวน certificate ทั้งหมด ซึ่งมีทั้งหดม 127 certificates

```
cat ./ca-certificates.cert | grep "BEGIN CERTIFICATE" | wc -l
```

Question 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.

Answer

จากที่สังเกตจุดหลัก ๆ ที่จะแตกต่างกันก็คือ X509v3 extensions **นั่นคือ**

ของ DigicertRootCA

```
X509v3 extensions:
    X509v3 Key Usage: critical
        Digital Signature, Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
        CA:TRUE
    X509v3 Subject Key Identifier:
        03:DE:50:35:56:D1:4C:BB:66:F0:A3:E2:1B:1B:C3:97:B2:3D:D1:55
    X509v3 Authority Key Identifier:
        keyid:03:DE:50:35:56:D1:4C:BB:66:F0:A3:E2:1B:1B:C3:97:B2:3D:D1:55
```

ของ Digicert

```
X509v3 extensions:
    X509v3 Basic Constraints: critical
        CA:TRUE, pathlen:0
    X509v3 Subject Key Identifier:
        0A:BC:08:29:17:8C:A5:39:6D:7A:0E:CE:33:C7:2E:B3:ED:FB:C3:7A
    X509v3 Authority Key Identifier:
        keyid:03:DE:50:35:56:D1:4C:BB:66:F0:A3:E2:1B:1B:C3:97:B2:3D:D1:55

    X509v3 Key Usage: critical
        Digital Signature, Certificate Sign, CRL Sign
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
    Authority Information Access:
        OCSP - URI:http://ocsp.digicert.com
        CA Issuers -
URI:http://cacerts.digicert.com/DigiCertGlobalRootCA.crt

    X509v3 CRL Distribution Points:

        Full Name:
            URI:http://crl3.digicert.com/DigiCertGlobalRootCA.crl

    X509v3 Certificate Policies:
        Policy: 2.16.840.1.114412.2.1
```

```
        Policy: 2.23.140.1.1
        Policy: 2.23.140.1.2.1
        Policy: 2.23.140.1.2.2
        Policy: 2.23.140.1.2.3
```

และของ Twitter

```
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:
        01:59:A3:CA:D4:41:E2:D8:40:BD:F9:8C:8B:13:C8:76:76:2D:0C:34
    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: critical
        CA:FALSE
```

ซึ่งจะสังเกตว่า DigicertRootCA จะไม่มี field `Authority Information Access` ซึ่งเป็น field ที่บอกว่า certificate ออกโดยใคร

นอกจากนี้สำหรับ intermediate และ root จะมี `X509v3 Authority Key Identifier` เหมือนกัน

และมากไปกว่านั้น root certificate จะมี `Subject` เป็น `Subject: C=US, O=DigiCert Inc, OU=www.digicert.com, CN=DigiCert Global Root CA` ซึ่งก็คือตัวเอง แปลว่าเป็น certificate ที่ sign โดยตัวเอง ในขณะที่ field นี้ใน certificate อื่นจะเป็นตัวที่เอาไปบอกว่า certificate นี้ sign โดยใคร

Question 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.)

Answer

รัน command

```
openssl x509 -inform der -in intermediate.cert -out intermediate.pem
```

Question 10

From the given python code, implement the certificate validation

```
from OpenSSL import crypto
import pem


def verify(target_filename, intermediate_filenames, root_filename):
    with open(target_filename, 'r') as cert_file:
        cert = cert_file.read()
    int_certs = []
    for filename in intermediate_filenames:
        with open(filename, 'r') as cert_file:
            int_certs.append(cert_file.read())
    pems = pem.parse_file(root_filename)
    trusted_certs = []
    for mypem in pems:
        trusted_certs.append(str(mypem))
    trusted_certs += int_certs
    verified = verify_chain_of_trust(cert, trusted_certs)
    if verified:
        print('Certificate verified')


def verify_chain_of_trust(cert_pem, trusted_cert_pems):
    certificate = crypto.load_certificate(crypto.FILETYPE_PEM, cert_pem)
    # Create and fill a X509Store with trusted certs
    store = crypto.X509Store()
    for trusted_cert_pem in trusted_cert_pems:
        trusted_cert = crypto.load_certificate(
            crypto.FILETYPE_PEM, trusted_cert_pem)
        store.add_cert(trusted_cert)

    # Create a X590StoreContext with the cert and trusted certs
```

```
    # and verify the the chain of trust
    store_ctx = crypto.X509StoreContext(store, certificate)
    # Returns None if certificate can be validated
    result = store_ctx.verify_certificate()
    if result is None:
        return True
    else:
        return False
```

Use your program to verify the certificates of: `Twitter`, `Google`, `www.chula.ac.th`, `classdeedee.cloud.cp.eng chula.ac.th`

Answer

ผลลัพธ์ที่ได้พบว่า certificate ถูก verified ได้ทั้งหมด

```
Verfiying Twitter certificate...
Certificate verified
Verfiying Google certificate...
Certificate verified
Verfiying Chula certificate...
Certificate verified
Verfiying Classdeedee certificate...
Certificate verified
```

โดย verifying chain เป็นดังนี้

```
print("Verfiying Twitter certificate...")
verify("twitter_com.cert", ["int_twitter_com.cert"], "ca-
certificates.cert")

print("Verfiying Google certificate...")
verify("google_com.cert", ["int_google_com.cert",
       "int_google_com_2.cert"], "ca-certificates.cert")

print("Verfiying Chula certificate...")
verify("chula_ac_th.cert", ["int_chula_ac_th.cert"], "ca-
certificates.cert")

print("Verfiying Classdeedee certificate...")
verify("classdeedee.cert", ["int_classdeedee.cert"], "ca-
certificates.cert")
```

Question 11

Nowaday, 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?

Answer

Class 1 จะมีระดับความปลอดภัยที่ต่ำ เนื่องจากการตรวจสอบจะใชเพียงแต่ email เท่านั้น แต่ในขณะเดียวกันมันสามารถ support browser เก่า ๆ ได้ดีกว่า class 3 ที่มีการ verification ที่รัดกุมกว่า และปลอดภัย แต่ยุ่งยากกว่า

Question 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

What further attacks can the attacker stage? Draw a possible attack setup.

Answer

นั่นแปลว่าเราสามารถ sign certificate ของใครสักคนจากถูกเป็นผิด อละจากผิดเป็นถูกได้

ซึ่งการ sign จากถูกไปผิดทำให้ browser ไม่สามารถ verify ได้และทำให้ browser ไม่เชื่อใน service นั้นอีกต่อไป ใขณะที่ hacker อาจจะสร้าง malicious website จากนั้น sign certificate ให้ website ตัวเองแล้วหลอกให้ user ติดกับว่า browser เชื่อถือ website นี้ ซึ่ง website นั้นอาจจะทำการ install malware หรืออาจจะ phishing เพื่อเอาข้อมูลเราไปได้เหมือนกัน

In the attack you have described above, can we rely on CRLs or OCSP for protection? Please explain

Answer

certificate อะไรที่ถูกเผิกถอน เพราะ credential ที่ให้ Issuer บางอย่าง Issuer ไม่เชื่อ ก็จะไม่ถูกเผิกถอนอีกต่อไป (ถ้า hacker ต้องการ) ดังนั้น CRLs และ OCSP ก็จะไม่มีประโยชน์เลย