CRY 2024

Laboratoire #4

June 2024 - Nathan Rayburn

1 Pourquoi devons nous transmettre une chaine de certificats dans les deux applications (email et TLS) ?

A chain of certificates is required in both situations to ensure the trustworthiness of the certificate being used. For the web server, each endpoint user receives the intermediate (RAYBURN-TLS) and client certificates (IP Certificat) and must use their trusted root certificate to validate the chain. This ensures everything is authentic and is essential for establishing a secure connection.

Concerning the e-mail application, the basics of validating a signature involve using the person's public key.

Thus, we need a valid certificate of the user's public key that is signed and issued by an intermediate CA. This ensures that the public key which was sent to the destination user is from the expected source user. Of course the end user will need the ROOT CA CRT, the intermediate CA and the client's CRT to validate the chain.

2 Comment avez-vous configuré nginx ? Donnez votre fichier de configuration.

The nginx server's configuration was set up to listen on both HTTP (80) and HTTPS (443). I used the Mozilla's intermediate guidelines as a template to configure the server and adapted what was asked in the lab. I had to remove the passphrase protection on my private key to ensure that nginx can read the key properly. I added the certificate chain for final + intermediate CA which is given to the client when establishing a connection. I left the original cipher settings since after reading them they all seemed secure. I added the index location for the root endpoint to know which file we want to display.

```
# generated 2024-06-19, Mozilla Guideline v5.7, nginx 1.17.7, OpenSSL 1.1.1k, intermediate configuration
# https://ssl-config.mozilla.org/#server=nginx&version=1.17.7&config=intermediate&openssl=1.1.1k&guideline=5.7

server {
    listen 80;
    listen [::]:80;

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    ssl_certificate /etc/ssl/mySSL/certs/chain_inter_final.crt;
    ssl_certificate_key /etc/ssl/mySSL/private/IP.key.unprotected;
    ssl_session_timeout 1d;
    ssl_session_cache shared:MozSSL:10m;  # about 40000 sessions
    ssl_session_tickets off;

    # curl https://ssl-config.mozilla.org/ffdhe2048.txt > /etc/ssl/mySSL/dhparam/dhparam
    ssl_dhparam /etc/ssl/mySSL/dhparam/dhparam;

    # intermediate configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-CHACHA20-POLY1305;
    ssl_prefer_server_ciphers on;

    # HSTS (ngx_http_headers_module is required) (63072000 seconds)
    add_header Strict-Transport-Security "max-age=63072000" always;

    location / {
        root /var/www/labo-crypto.com/public;
        index index.html;
    }

    # replace with the IP address of your resolver
    resolver 127.0.0.1;
}
```

Listing 1: Configuration nginx

3 Fournissez le résultat du scan de testssl sur votre serveur ainsi que des commentaires, si nécessaire.

Command used :

```
./testssl.sh --add-ca ../HEIG-VDRoot.crt 10.190.133.22:44314 > testssl_results_ca.txt
```

Listing 2: Command for testssl

```
SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      not offered
TLS 1.1    not offered
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   h2, http/1.1 (advertised)
ALPN/HTTP2 h2, http/1.1 (offered)
```

Listing 3: Testing protocols via sockets except NPN+ALPN

```
NULL ciphers (no encryption)                  not offered (OK)
Anonymous NULL Ciphers (no authentication)    not offered (OK)
Export ciphers (w/o ADH+NULL)                 not offered (OK)
LOW: 64 Bit + DES, RC[2,4] (w/o export)       not offered (OK)
Triple DES Ciphers / IDEA                     not offered
Obsolete CBC ciphers (AES, ARIA etc.)         not offered
Strong encryption (AEAD ciphers)              offered (OK)
```

Listing 4: Testing cipher categories

```
PFS is offered (OK)          TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 ECDHE-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-GCM-SHA384
                             ECDHE-RSA-CHACHA20-POLY1305 DHE-RSA-CHACHA20-POLY1305 TLS_AES_128_GCM_SHA256 ECDHE-RSA-AES128-GCM-SHA256
                             DHE-RSA-AES128-GCM-SHA256
Elliptic curves offered:     prime256v1 secp384r1 secp521r1 X25519 X448
DH group offered:            ffdhe2048
```

Listing 5: Testing robust (perfect) forward secrecy, (P)FS

```
Has server cipher order?     yes (OK) -- TLS 1.3 and below
Negotiated protocol          TLSv1.3
Negotiated cipher            TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Cipher order
    TLSv1.2:   ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-CHACHA20-POLY1305 DHE-RSA-AES128-GCM-SHA256
               DHE-RSA-AES256-GCM-SHA384 DHE-RSA-CHACHA20-POLY1305
    TLSv1.3:   TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256
```

Listing 6: Testing server preferences

```
TLS extensions (standard)     "renegotiation info/#65281" "EC point
    formats/#11" "next protocol/#13172" "supported versions/#43"
                              "key share/#51" "supported_groups/#10"
                                   "max fragment length/#1"
                              "application layer protocol
                                  negotiation/#16" "extended master
                                  secret/#23"
Session Ticket RFC 5077 hint no -- no lifetime advertised
SSL Session ID support        yes
Session Resumption            Tickets no, ID: yes
TLS clock skew                Random values, no fingerprinting
    possible
Signature Algorithm           SHA256 with RSA
Server key size               RSA 2048 bits
Server key usage              Digital Signature, Key Encipherment
Server extended key usage     TLS Web Server Authentication, TLS Web
     Client Authentication
Serial                        815C55CE4146FC12 (OK: length 8)
Fingerprints                  SHA1
                              B3FCB7BAA2AC22F7C4F1B05D3056B9D2B1B
                              81CEC
                              SHA256
                              93C00E66C923BFA3A46783E2DE0EF30E235
                              F7CB931EEE7DE1FFFDA5AB3383A40
Common Name (CN)              IP
subjectAltName (SAN)          10.190.133.22
Issuer                        RAYBURN-TLS (HEIG-VD from CH)
Trust (hostname)              Ok via SAN
Chain of trust                Ok
EV cert (experimental)        no
ETS/"eTLS", visibility info   not present
Certificate Validity (UTC)    398 >= 60 days (2024-06-18 14:37 -->
    2025-07-23 14:37)
# of certificates provided    2
Certificate Revocation List   --
OCSP URI                      --
                              NOT ok -- neither CRL nor OCSP URI
                                  provided
OCSP stapling                 not offered
OCSP must staple extension    --
DNS CAA RR (experimental)     not offered
Certificate Transparency      --
```

Listing 7: Testing server defaults (Server Hello)

```
HTTP Status Code              200 OK
HTTP clock skew               0 sec from localtime
Strict Transport Security     730 days=63072000 s, just this domain
Public Key Pinning            --
Server banner                 nginx/1.14.0 (Ubuntu)
Application banner            --
Cookie(s)                     (none issued at "/")
Security headers              --
Reverse Proxy banner          --
```

Listing 8: Testing HTTP header response @ ”/”

```
Heartbleed (CVE-2014-0160)                not vulnerable (OK), no
    heartbeat extension
CCS (CVE-2014-0224)                       not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (OK), no
    session ticket extension
ROBOT                                     Server does not support
    any cipher suites that use RSA key transport
Secure Renegotiation (RFC 5746)           supported (OK)
Secure Client-Initiated Renegotiation     not vulnerable (OK)
CRIME, TLS (CVE-2012-4929)                not vulnerable (OK)
BREACH (CVE-2013-3587)                    potentially NOT ok, "gzip
    " HTTP compression detected. - only supplied "/" tested
                                          Can be ignored for static
                                               pages or if no
                                              secrets in the page
POODLE, SSL (CVE-2014-3566)               not vulnerable (OK), no
    SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507)              No fallback possible (OK)
    , no protocol below TLS 1.2 offered
SWEET32 (CVE-2016-2183, CVE-2016-6329)    not vulnerable (OK)
FREAK (CVE-2015-0204)                     not vulnerable (OK)
DROWN (CVE-2016-0800, CVE-2016-0703)      not vulnerable on this
    host and port (OK)

LOGJAM (CVE-2015-4000), experimental      common prime with 2048
    bits detected: RFC7919/ffdhe2048 (2048 bits),
                                          but no DH EXPORT ciphers
BEAST (CVE-2011-3389)                     not vulnerable (OK), no
    SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental     not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808)        no RC4 ciphers detected (
    OK)
```

Listing 9: Testing vulnerabilities

```
Hexcode  Cipher Suite Name (OpenSSL)       KeyExch.   Encryption  Bits     Cipher Suite Name (IANA/RFC)
-----------------------------------------------------------------
x1302   TLS_AES_256_GCM_SHA384            ECDH 253   AESGCM      256      TLS_AES_256_GCM_SHA384
x1303   TLS_CHACHA20_POLY1305_SHA256      ECDH 253   ChaCha20    256      TLS_CHACHA20_POLY1305_SHA256
xc030   ECDHE-RSA-AES256-GCM-SHA384       ECDH 256   AESGCM      256      TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
x9f     DHE-RSA-AES256-GCM-SHA384         DH 2048    AESGCM      256      TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
xcca8   ECDHE-RSA-CHACHA20-POLY1305       ECDH 253   ChaCha20    256      TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
xccaa   DHE-RSA-CHACHA20-POLY1305         DH 2048    ChaCha20    256      TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
x1301   TLS_AES_128_GCM_SHA256            ECDH 253   AESGCM      128      TLS_AES_128_GCM_SHA256
xc02f   ECDHE-RSA-AES128-GCM-SHA256       ECDH 256   AESGCM      128      TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
x9e     DHE-RSA-AES128-GCM-SHA256         DH 2048    AESGCM      128      TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
```

Listing 10: Testing 370 ciphers via OpenSSL plus sockets against the server, ordered by encryption strength

```
Android 6.0                  TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Android 7.0 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 256 bit ECDH (P-256)
Android 8.1 (native)         TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256, 253 bit ECDH (X25519)
Android 9.0 (native)         TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Android 10.0 (native)        TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Android 11 (native)          TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Android 12 (native)          TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Chrome 79 (Win 10)           TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Chrome 101 (Win 10)          TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Firefox 66 (Win 8.1/10)      TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
Firefox 100 (Win 10)         TLSv1.3 TLS_AES_256_GCM_SHA384, 253 bit ECDH (X25519)
IE 6 XP                      No connection
IE 8 Win 7                   No connection
IE 8 XP                      No connection
```

Listing 11: Running client simulations (HTTP) via sockets

```
       IE 11 Win 7                  TLSv1.2 DHE-RSA-AES128-GCM-
           SHA256, 2048 bit DH  (ffdhe2048)
IE 11 Win 8.1                TLSv1.2 DHE-RSA-AES128-GCM-SHA256,
    2048 bit DH  (ffdhe2048)
IE 11 Win Phone 8.1          No connection
IE 11 Win 10                 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    256 bit ECDH (P-256)
Edge 15 Win 10               TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    253 bit ECDH (X25519)
Edge 101 Win 10 21H2         TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
Safari 12.1 (iOS 12.2)       TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
Safari 13.0 (macOS 10.14.6)  TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
Safari 15.4 (macOS 12.3.1)   TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
Java 7u25                    No connection
Java 8u161                   TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    256 bit ECDH (P-256)
Java 11.0.2 (OpenJDK)        TLSv1.3 TLS_AES_256_GCM_SHA384, 256
    bit ECDH (P-256)
Java 17.0.3 (OpenJDK)        TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
go 1.17.8                    TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
LibreSSL 2.8.3 (Apple)       TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    253 bit ECDH (X25519)
OpenSSL 1.0.2e               TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    256 bit ECDH (P-256)
OpenSSL 1.1.0l (Debian)      TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    253 bit ECDH (X25519)
OpenSSL 1.1.1d (Debian)      TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
OpenSSL 3.0.3 (git)          TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
Apple Mail (16.0)            TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256,
    256 bit ECDH (P-256)
Thunderbird (91.9)           TLSv1.3 TLS_AES_256_GCM_SHA384, 253
    bit ECDH (X25519)
```

Listing 12: Running client simulations (HTTP) via sockets

3.1 Conclusion

The setup and certificates generated for the lab have been validated through the `testssl` scan results. The server supports only secure protocols (TLS 1.2 and TLS 1.3), offers strong encryption ciphers, and enables Perfect Forward Secrecy (PFS). The certificate chain is valid, with the chain of trust intact and verified. The server configuration is robust against known vulnerabilities, ensuring a secure environment for both email and HTTPS connections. The server's response includes necessary security headers, confirming the proper setup.

Comments on testssl Output

- **Protocols Supported:**

```
    SSLv2       not offered (OK)
    SSLv3       not offered (OK)
    TLS 1       not offered
    TLS 1.1     not offered
    TLS 1.2     offered (OK)
    TLS 1.3     offered (OK): final
```

  Comment: The server supports only the secure TLS 1.2 and TLS 1.3 protocols, which is what we wanted.

- **Cipher Categories:**

```
    NULL ciphers (no encryption)                  not offered
        (OK)
    Anonymous NULL Ciphers (no authentication)    not offered
        (OK)
    Export ciphers (w/o ADH+NULL)                 not offered
        (OK)
    LOW: 64 Bit + DES, RC[2,4] (w/o export)       not offered
        (OK)
    Triple DES Ciphers / IDEA                     not offered
    Obsolete CBC ciphers (AES, ARIA etc.)         not offered
    Strong encryption (AEAD ciphers)              offered (OK)
```

  Comment: Only strong encryption cipher suites are offered, ensuring robust encryption.

- **Perfect Forward Secrecy (PFS):**

```
    PFS is offered (OK)          TLS_AES_256_GCM_SHA384
        TLS_CHACHA20_POLY1305_SHA256 ECDHE-RSA-AES256-GCM-
        SHA384 DHE-RSA-AES256-GCM-SHA384
                                 ECDHE-RSA-CHACHA20-POLY1305
                                     DHE-RSA-CHACHA20-POLY1305
                                      TLS_AES_128_GCM_SHA256
                                     ECDHE-RSA-AES128-GCM-
                                     SHA256
                                 DHE-RSA-AES128-GCM-SHA256
    Elliptic curves offered:     prime256v1 secp384r1
        secp521r1 X25519 X448
    DH group offered:            ffdhe2048
```

  Comment: PFS is enabled.

- **Server Preferences:**

```
    Has server cipher order?     yes (OK) -- TLS 1.3 and below
    Negotiated protocol          TLSv1.3
    Negotiated cipher            TLS_AES_256_GCM_SHA384, 253
        bit ECDH (X25519)
```

Comment: The server enforces a strong cipher order and prefers secure ciphers.

- **Server Defaults (Server Hello):**

```
    Common Name (CN)             IP
    subjectAltName (SAN)         10.190.133.22
    Issuer                       RAYBURN-TLS (HEIG-VD from CH)
    Trust (hostname)             Ok via SAN
    Chain of trust               Ok
```

Comment: The server's certificate chain is valid and correctly configured. The chain of trust is intact, verifying the authenticity and integrity of the certificates used.

- **HTTP Header Response:**

```
    HTTP Status Code             200 OK
    Strict Transport Security    730 days=63072000 s, just
        this domain
    Server banner                nginx/1.14.0 (Ubuntu)
```

Comment: The server responds with a proper HTTP status code.

- **Vulnerabilities:**

```
    Heartbleed (CVE-2014-0160)                not vulnerable (
        OK), no heartbeat extension
    CCS (CVE-2014-0224)                       not vulnerable (
        OK)
    Ticketbleed (CVE-2016-9244), experiment.  not vulnerable (
        OK), no session ticket extension
    ROBOT                                     Server does not
        support any cipher suites that use RSA key transport
    Secure Renegotiation (RFC 5746)           supported (OK)
    Secure Client-Initiated Renegotiation     not vulnerable (
        OK)
    CRIME, TLS (CVE-2012-4929)                not vulnerable (
        OK)
    BREACH (CVE-2013-3587)                    potentially NOT
        ok, "gzip" HTTP compression detected. - only supplied
        "/" tested
                                              Can be ignored
                                                  for static
                                                  pages or if
                                                  no secrets
                                                  in the page
    POODLE, SSL (CVE-2014-3566)               not vulnerable (
        OK), no SSLv3 support
```

Comment: The server is not vulnerable to various known attacks, indicating a secure setup.

4 Quelle durée de validité avez-vous choisie pour le certificat du serveur TLS ? Pourquoi ?

I chose a validity period of 400 days, which is slightly more than a year. This strikes a balance between security and convenience. It ensures security by avoiding long-term certificates that could become vulnerable over time. The certificate needs to be renewed annually, and the additional month provides a buffer period to generate and implement a new certificate without any rush.