

# Analyse du Handshake SSL/TLS avec Wireshark

Nadhir Boukhechem

L'objectif de ce TP est de d'analyser un échange SSL/TLS en capturant le trafic réseau, de comprendre les étapes principales de l'authentification et de l'établissement de session, et de saisir le rôle du certificat dans la sécurité des communications.

- Lancez Wireshark.
- Sélectionnez l'interface réseau active.
- Lancez la capture de paquets.
- Pour éviter d'être submergé, appliquez le filtre de capture au port 443 (port standard du HTTPS/TLS)
- Dans votre navigateur, videz complètement le cache et l'historique de la dernière heure. Ceci garantit que le navigateur ne réutilisera pas une session TLS existante.
- Ouvrez une fenêtre de navigation et accédez à un site sécurisé de votre choix.
- Une fois la page chargée, arrêtez immédiatement la capture Wireshark.
- Appliquez le filtre de visualisation ssl ou tls dans Wireshark. Vous devriez voir une séquence de paquets commençant par Client Hello.

11001	286.069484477	10.25.32.85	52.108.240.63	TLSv1.3	75 Application Data
11002	286.071165477	10.25.32.85	52.108.240.63	TLSv1.3	1412 Application Data
11005	286.111110339	52.108.240.63	10.25.32.85	TLSv1.3	166 Application Data
11032	290.410432969	52.108.50.37	10.25.32.85	TLSv1.3	98 Application Data
11082	292.976268415	10.25.32.85	104.21.11.198	QUIC	1294 Initial, DCID=33d7f07966e3e5ad, SCID=d0e98d, PKN: 5, CRYPTO
11095	293.039999384	10.25.32.85	104.21.11.198	TLSv1.3	517 Client Hello (SNI=cloudflare-ech.com)
11103	293.085209352	104.21.11.198	10.25.32.85	TLSv1.3	3444 Server Hello, Change Cipher Spec, Application Data
11107	293.087356349	10.25.32.85	104.21.11.198	TLSv1.3	130 Change Cipher Spec, Application Data
11109	293.089099611	10.25.32.85	104.21.11.198	TLSv1.3	158 Application Data
11112	293.098833754	104.21.11.198	10.25.32.85	TLSv1.3	587 Application Data, Application Data
11114	293.099439221	10.25.32.85	104.21.11.198	TLSv1.3	97 Application Data
11488	295.194389508	10.25.32.85	52.108.50.37	TLSv1.3	102 Application Data

- Développez le premier paquet de la séquence, étiqueté Client Hello.

```

▼ Frame 11095: 517 bytes on wire (4136 bits), 517 bytes captured (4136 bits) on interface
  Section number: 1
    ► Interface id: 0 (wlp3s0)
      Encapsulation type: Ethernet (1)
        Arrival Time: Dec 8, 2025 09:50:22.392247120 CET
        UTC Arrival Time: Dec 8, 2025 08:50:22.392247120 UTC
        Epoch Arrival Time: 1765183822.392247120
        [Time shift for this packet: 0.000000000 seconds]
        [Time delta from previous captured frame: 0.000023340 seconds]
        [Time delta from previous displayed frame: 0.063730969 seconds]
        [Time since reference or first frame: 293.039999384 seconds]
        Frame Number: 11095
        Frame Length: 517 bytes (4136 bits)
        Capture Length: 517 bytes (4136 bits)
        [Frame is marked: False]
        [Frame is ignored: False]
  
```

- Question 1 : Quel est le protocole supporté par votre navigateur

Réponse : Le protocole supporté par le navigateur est TLSv1.3 et v1.2

- Question 2 : Identifiez la liste des Cipher Suites proposées par le client. Ces suites définissent les algorithmes de chiffrement, de hachage et d'échange de clés que le client supporte.

Réponse :

```

Cipher Suites Length: 34
▼ Cipher Suites (17 suites)
  Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
  Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
  Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
  Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
  Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
  Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
Compression Methods Length: 1

```

- Localisez le paquet de réponse du serveur, étiqueté Server Hello.

11082	292.970208415	10.25.32.85	104.21.11.198	QUIC	1294	Initial, DCID=3307107900e3e3au, SCID=00e980, PNM: 0, CK
11095	293.039999384	10.25.32.85	104.21.11.198	TLSv1.3	517	Client Hello (SNI=cloudflare-ech.com)
11103	293.085209352	104.21.11.198	10.25.32.85	TLSv1.3	3444	Server Hello, Change Cipher Spec, Application Data
11107	293.087356349	10.25.32.85	104.21.11.198	TLSv1.3	130	Change Cipher Spec, Application Data
11109	293.089099611	10.25.32.85	104.21.11.198	TLSv1.3	158	Application Data

- Question 3 : Quelle est la Cipher Suite que le serveur a choisie parmi celles proposées par le client ?

Réponse : La Cipher Suite que le serveur a choisie est : TLS\_AES\_128\_GCM\_SHA256 (0x1301)

- Développez le paquet Certificate qui est souvent inclus dans le paquet suivant. (Vous pouvez taper dans la barre Wireshark : **tls.handshake.type == 11**)

```

tls.handshake.type == 11
No.    Time    Source    Destination    Protocol    Length    Info
9103 196.878023144 199.232.169.91 10.25.32.85 TLSv1.2 1749 Certificate, Server Key Exchange, Server Hello

Handshake Protocol: Certificate
  Handshake Type: Certificate (11)
  Length: 2659
  Certificates Length: 2656
  Certificates (2656 bytes)
    Certificate Length: 1360
    Certificate [truncated]: 3082054c30820434a003020102021205cb296189cace9cc
      signedCertificate
        algorithmIdentifier (sha256WithRSAEncryption)
          Padding: 0
          encrypted [truncated]: 6889a00c94ce21e29cfb54bab5cd86f4589bda287456630
        Certificate Length: 1290
      Certificate [truncated]: 30820506308202eea003020102021100c212324b70a9b49
        signedCertificate
          algorithmIdentifier (sha256WithRSAEncryption)
            Padding: 0
            encrypted [truncated]: 8f75d009cf6a7648653292deb544c88576f415848c02bf
  
```

- Question 4 : Examinez le certificat numérique du serveur. Identifiez la clé publique du serveur et le nom de l'Autorité de Certification (CA) qui a émis le certificat.

Réponse :

```

Certificate [truncated]: 30820506308202eea003020102021100c212324b70a9b49
  signedCertificate
    version: v3 (2)
    serialNumber: 0x00c212324b70a9b49171dc40f7e285263c
    signature (sha256WithRSAEncryption)
    issuer: rdnSequence (0)
    validity
    subject: rdnSequence (0)
    subjectPublicKeyInfo
      algorithm (rsaEncryption)
        Algorithm Id: 1.2.840.113549.1.1.1 (rsaEncryption)
      subjectPublicKey [truncated]: 3082010a0282010100da982874adbe94fe
        modulus: 0x00da982874adbe94fe3be01ee2e54b75ab2c127feda703327e3
        publicExponent: 65537
    extensions: 0 items
  
```

```

Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)
  issuer: rdnSequence (0)
    rdnSequence: 3 items (id-at-commonName=ISRG Root X1,id-at-organizationName=Inte
      RDNSSequence item: 1 item (id-at-countryName=US)
      RDNSSequence item: 1 item (id-at-organizationName=Internet Security Research
      RDNSSequence item: 1 item (id-at-commonName=ISRG Root X1)
        RelativeDistinguishedName item (id-at-commonName=ISRG Root X1)
    validity
  
```

ISRG Root X1 est la racine de l'Autorité de Certification de Let's Encrypt.

- Expliquez en quoi le certificat est essentiel à l'étape d'authentification du serveur.

Réponse : Le certificat est essentiel au client pour vérifier l'identité du serveur en s'assurant qu'il a été émis par une Autorité de Certification conforme. Il contient également la clé publique du serveur, qui sert à établir une connexion chiffrée et authentifiée.

- Question 5 : Quel est l'objectif de l'étape Key Exchange qui suit le certificat ?

Réponse : L'étape Key Exchange sert à ce que le client et le serveur se mette en accord sur la clé secrète. Cette clé sera utilisée pour protéger toutes les données échangées pendant la connexion.

- Pourquoi la suite des paquets après le Handshake est-elle illisible dans Wireshark ?

Réponse : La suite des paquets après le Handshake est illisible dans Wireshark car désormais les données sont chiffrées avec la clé de session. Car après le passage du Key Exchange, les connexions sont désormais chiffrées.

- Expliquez brièvement le concept de Perfect Forward Secrecy (PFS)

Réponse : Le Perfect Forward Secrecy est un système de clé temporaire unique pour chaque session, assurant que les anciens échanges restent confidentiels en cas de compromission de la clé privée du serveur.

#### Question de réflexion :

Les pare-feux peuvent-ils filtrer les paquets HTTPS ? Si oui, expliquez comment, en détaillant autant que possible.

Oui, les pare-feux sont en capacités de filtrer les paquets HTTPS, mais de manière indirecte car le pare-feu ne peut pas directement lire le contenu des paquets.

Il est possible de filtrer grâce aux informations lisibles comme l'adresse ip, le port, le nom de domaine, la version du protocole, par la taille ou la fréquence des paquets.

Il est tout de même possible d'utiliser une méthode DPI , (se place entre le client et serveur, en tant que proxy)

il permet de déchiffrer, inspecter, et re-chiffrer.