

# TP Applications Internet sécurisées

(Veuillez reporter vos réponses directement sur ce document)

Noms et prénoms (binôme) : **GUARDIA Quentin**

A remettre : au plus tard le dimanche 18h00 à master2srs.dir@gmail.com

Indiquer « M1 CYBER TP7 SSL »

## 1. HTTPS

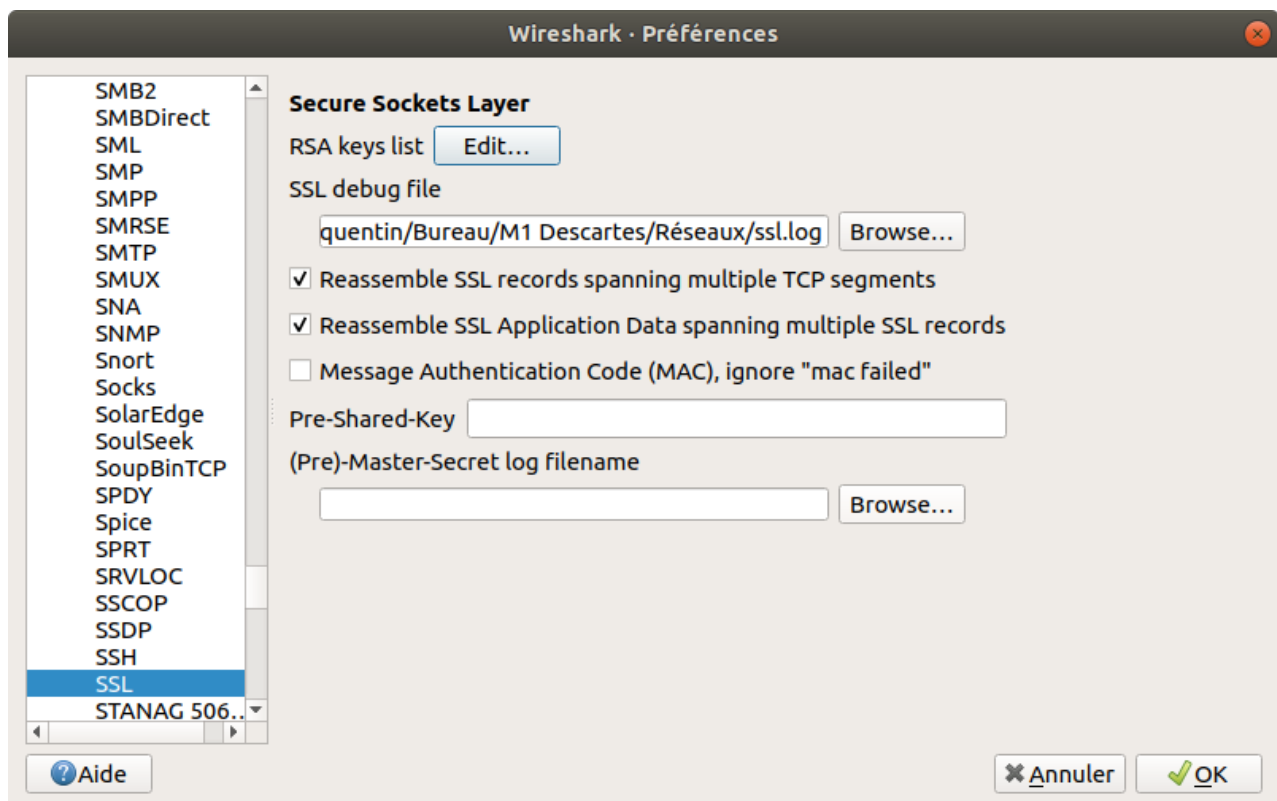
*Objectif : établir une session web sécurisée avec SSL/TLS, et analyser les échanges de messages et les différents protocoles employés.*

Connectez vous à votre compte webmail (ou bien en créer un sur le site [www.yahoo.fr](http://www.yahoo.fr) ou autre site équivalent supportant les transactions sécurisées SSL/TLS).

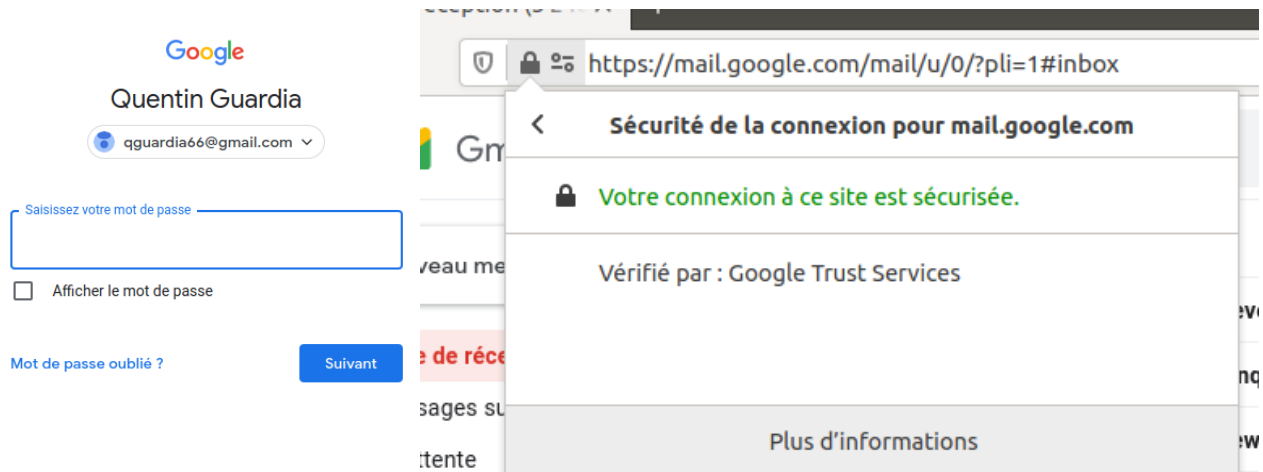
Utiliser le support de cours, les RFC 2818 et RFC 2246 ainsi que le logiciel Wireshark pour capturer les échanges entre votre client et le serveur webmail.

Au moyen de Wireshark, veuillez capturer et filtrer les échanges SSL/TLS dans un fichier log, et le nommer « ssl.log ».

Je fais en sorte d'enregistrer les logs SSL sous ssl.log via éditer > préférences > protocols > SSL, puis :

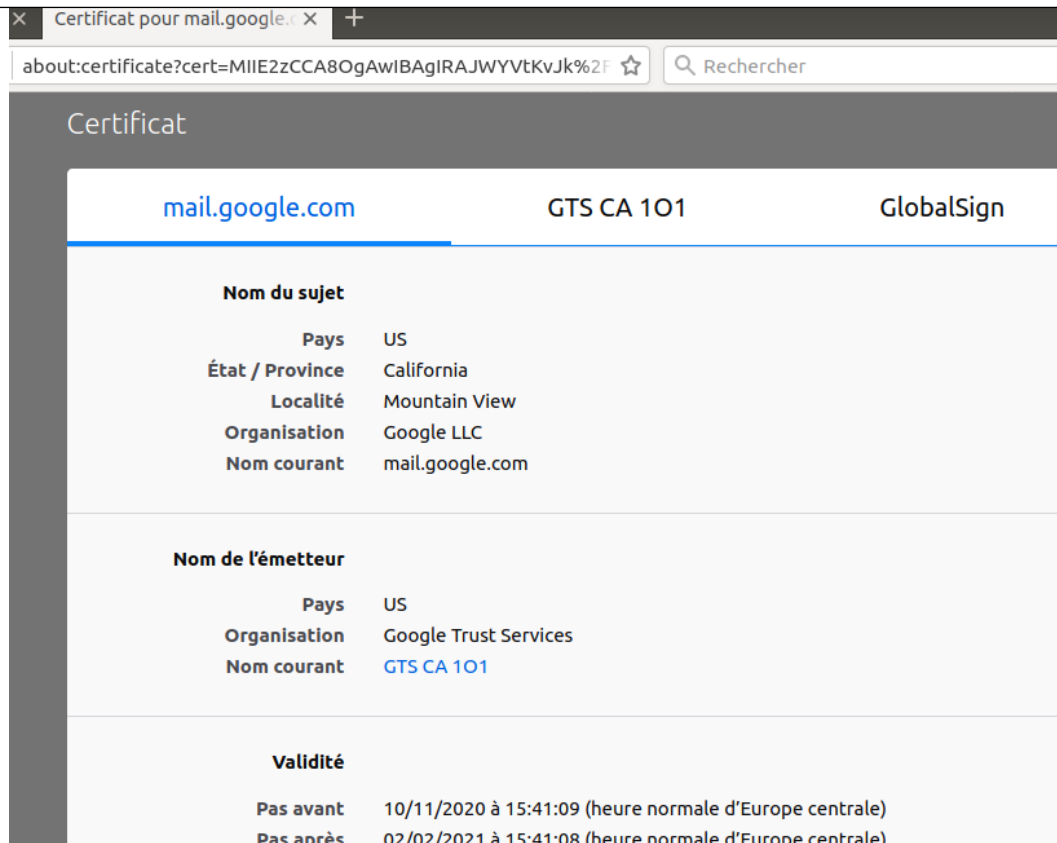


Je me connecte à mon compte gmail après avoir lancé la capture :



Et j'accède à certaines informations du serveur en cliquant sur le cadenas du navigateur (Mozilla Firefox). Je sélectionne alors « plus d'informations », « sécurité » et enfin « afficher le certificat ».





Veillez ensuite répondre aux questions suivantes :

### 6.1 - comment sont transmis votre login et mot de passe (chiffré, en clair, ...) ?

Tout est chiffré avec TLS. Comme le relève Wireshark :

*TLSv1.2 Record Layer: Application Data Protocol: http-over-tls*

7	0.8618110...	Zhejia...	Broadc...	ARP	60	Gratuitous ARP for 192.168.1.23 (Reply)
8	1.2800586...	192.16...	52.42...	TCP	66	58266 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3191754563 TSecr=...
9	1.4643514...	52.42...	192.16...	TCP	66	[TCP ACKed unseen segment] 443 → 58266 [ACK] Seq=1 Ack=2 Win=123 L...
10	1.5934485...	2a01:c...	2a00:1...	TLSv...	637	Application Data
11	1.5935174...	2a01:c...	2a00:1...	TLSv...	270	Application Data
12	1.5937252...	2a01:c...	2a00:1...	TLSv...	122	Application Data
13	1.6207871...	2a00:1...	2a01:c...	TCP	86	443 → 57592 [ACK] Seq=1 Ack=552 Win=391 Len=0 TSval=107394468 TSecr=...
14	1.6222648...	2a00:1...	2a01:c...	TCP	86	443 → 57592 [ACK] Seq=1 Ack=736 Win=402 Len=0 TSval=107394469 TSecr=...
Secure Sockets Layer						
TLSv1.2 Record Layer: Application Data Protocol: http-over-tls						
Content Type: Application Data (23)						
Version: TLS 1.2 (0x0303)						
Length: 546						
Encrypted Application Data: 792d70a5ae99924c1dde4f9c9273a641a6d5e32b3057b14e...						

Et effectivement, je ne retrouve pas mes identifiants en clair sur Wireshark.

### 6.2 - Combien d'aller-retours (TCP inclus) sont ils nécessaires pour poster un email avec un fichier attaché (toutes phases comprises)

- 1) SYN | retour : ACK (client puis serveur, donc x2)
- 3) ClientHello | retour : ServerHello, Certificate, ServerHelloDone

- 4) ClientKeyExchange, ChangerCipherSpec, Finished | retour : ChangerCipherSpec, Finished
- 5) HELO et code retour 250
- 6) MAIL FROM et code retour 250
- 7) RCPT TO et code retour 250
- 8) DATA et code retour 354
- 9) Autant de trames que nécessaire de type MIME pour envoyer le fichier. Puis code retour 250
- 10) QUIT et code retour 221

Je compte ainsi 10 aller-retours nécessaires à minima.

### 6.3 - Quel est votre identifiant de session (Session\_Id) ? :

L'identifiant de session se trouve dans le « Client Hello » et le « Server Hello ». Dans mon cas il s'agit de :

77:5d:8f:a9:3f:a2:9d:7b:3e:7a:f5:ca:68:df:fe:75:bd:e4:93:25:bd:36:2a:b1:f3:e1:b1:c7:70:32:b8:e0

100	2.6886876...	2a01:cb1d:840f:5300:1...	2a00:1450:4007:80a::2...	TLSv...	603	Client Hello
101	2.7156648...	2a00:1450:4007:80a::2...	2a01:cb1d:840f:5300:1...	TCP	86 443 → 38282	[ACK
102	2.7206475...	2a00:1450:4007:80a::2...	2a01:cb1d:840f:5300:1...	TLSv...	1294	Server Hello, Ch

Handshake Protocol: Client Hello	
Handshake Type:	Client Hello (1)
Length:	508
Version:	TLS 1.2 (0x0303)
Random:	1de6321a9663fbddec33c7bd59e771f774827af929879f487...
Session ID Length:	32
Session ID:	775d8fa93fa29d7b3e7af5ca68dffe75bde49325bd362ab1...
Cipher Suites Length:	36

28 20	77 5d 8f a9 3f a2 9d 7b 3e 7a f5 ca 68 df	( [w]..?..{>z..h.
fe 75 bd e4 93 25 bd 36 2a b1 f3 e1 b1 c7 70 32		..u...%·6 *.....p2
b8 e0 00 24 13 01 13 03 13 02 c0 2b c0 2f cc a9		..·\$.....+·/..
cc a8 c0 2c c0 30 c0 0a c0 09 c0 13 c0 14 00 9c		...·.0· .....

### 6.4 - Quelle est la fonction de hachage utilisée ? :

Les fonctions de hachage utilisées sont SHA1 et SHA256, comme on peut le voir sur la page des certificats de Firefox en capture dans l'introduction, à l'onglet mail.google.com :

Empreintes numériques	
SHA-256	B5:26:38:85:FA:DE:B1:08:D9:B6:D4:60:C8:97:E3:38:E6:C1:8A:47:72:1B:D6:A6:71:71:A8:43:BB:E9:D2:DF
SHA-1	D1:AF:BF:FE:4D:D7:E0:F6:A8:38:A6:49:05:8F:E9:82:07:FE:19:A0

L'authentification de message se fait par un hachage grâce à SHA256, comme on peut le voir sur la suite cryptographique utilisée :

```

2.3868295... 2a00:1450:4007:808::2... 2a01:cb1d:840f:5300:1... TLSv... 1294 Server Hello, Ch
Random: 922a399e5aa370bfe54d76bdb7350e9dc9d592af15ac29fe...
Session ID Length: 32
Session ID: 287489a79cca478d43a1c834c5ead071458a16e4be1c3117...
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
Compression Method: null (0)
Extensions Length: 46

```

Et le hachage utilisé dans les requêtes OCSF pour transmettre l'empreinte du nom de l'émetteur et celle de sa clé se fait par l'algorithme SHA1 :

```

835 11.917797... 2a01:cb1d:840f:5300:1... 2a00:1450:4007:810::2... OCSF 492 Request
Online Certificate Status Protocol
  tbsRequest
    requestList: 1 item
      Request
        reqCert
          hashAlgorithm (SHA-1)
            Algorithm Id: 1.3.14.3.2.26 (SHA-1)
            issuerNameHash: 424630c22719dhd70f08ffc73e5a65f663817hc

```

À noter que la signature du certificat se calcule avec SHA256 et RSA.

## 6.5 - Quelle est l'algorithme de chiffrement asymétrique qui sera utilisé ? :

Une courbe elliptique est utilisée, la Curve25519. Dans la page du certificat de mail.google.com, on peut voir :

Informations sur la clé publique	
Algorithme	Elliptic Curve
Taille de la clé	256
Courbe	P-256
Valeur publique	04:B2:48:6A:08:7D:1B:5B:C4:C9:36:85:C2:E0:D6:AB:E3:B1:6C:62

Et plus précisément sur Wireshark, où x25519 fait référence à Curve25519.

```

470 8.1382811... 2a01:cb1d:840f:5300:1... 2a00:1450:4007:812::2... TCP 86 37784 → 443 [ACK]
471 8.1394477... 192.168.1.88 216.58.198.195 TLSv... 583 Client Hello
Extension: key_share (len=107)
  Type: key_share (51)
  Length: 107
  Key Share extension
    Client Key Share Length: 105
    Key Share Entry: Group: x25519, Key Exchange length: 32
      Group: x25519 (29)
      Key Exchange Length: 32

```

Au début je cherchais des traces de RSA. Mais l'usage de RSA sert uniquement à la signature du certificat, combiné à SHA256.

## 6.6 - Identifier l'autorité qui à émis et qui certifie le certificat du serveur reçu par votre client :

Je retourne sur l'onglet de Firefox contenant les informations sur les certificats.

J'apprends que mail.google.com a un certificat émis et certifié par GTS CA 101 (Google Trust Services), qui est lui-même certifié par GlobalSign, une autorité de certification.

Nom courant	mail.google.com	Nom courant	GTS CA 101
<b>Nom de l'émetteur</b>		<b>Nom de l'émetteur</b>	
Pays	US	Unité organisationnelle	GlobalSign Root CA - R2
Organisation	Google Trust Services	Organisation	GlobalSign
Nom courant	GTS CA 101	Nom courant	GlobalSign

## 6.7 - identifier le numéro de série du certificat du serveur ? :

Le numéro de série du certificat du serveur est :

00:95:98:56:d2:af:26:4f:e5:05:00:00:00:00:7e:8c:ec

63	2.4046219...	2a01:cb1d:840f:5300:1...	2a00:1450:4007:810::2...	OCSP	493	[TCP Previous segment not captured] Request
64	2.4352094...	2a00:1450:4007:810::2...	2a01:cb1d:840f:5300:1...	TCP	86	[TCP ACKed unseen segment] 80 → 37990 [ACK]
65	2.4360466...	2a00:1450:4007:810::2...	2a01:cb1d:840f:5300:1...	OCSP	788	Response
66	2.4360893...	2a01:cb1d:840f:5300:1...	2a00:1450:4007:810::2...	TCP	86	37990 → 80 [ACK] Seq=409 Ack=703 Win=501 Le
requestList: 1 item						
▼ Request						
▼ reqCert						
▼ hashAlgorithm (SHA-1)						
Algorithm Id: 1.3.14.3.2.26 (SHA-1)						
issuerNameHash: 424630c22719dbde70f08ffc73e5a65f663817bc						
issuerKeyHash: 98d1f86e10ebcf9bec609f18901ba0eb7d09fd2b						
serialNumber: 0x00959856d2af264fe50500000007e8cec						
66	38	17	bc	04	14	98 d1 f8 6e 10 eb cf 9b ec 60 f8.....n.....
9f	18	90	1b	a0	eb	7d 09 fd 2b 02 11 00 95 98 56 .....}.+...V
d2	af	26	4f	e5	05	00 00 00 00 7e 8c ec ..&0....~...

Comme on peut le voir dans le paquet OCSP. Je vérifie qu'il s'agisse du numéro de série du bon certificat en allant sur l'onglet « mail.google.com » de la page des certificats du navigateur :

Divers	
<b>Numéro de série</b>	00:95:98:56:D2:AF:26:4F:E5:05:00:00:00:00:7E:8C:EC

## 6.8 - identifier la clé publique du serveur ? :

Toujours dans le about:certificate de Firefox :



Informations sur la clé publique	
Algorithme	Elliptic Curve
Taille de la clé	256
Courbe	P-256
Valeur publique	04:B2:48:6A:08:7D:1B:5B:C4:C9:36:85:C2:E0:D6:AB:E3:B1:6C:62:6E:28:94:65:86:42:C7:E5:81:C6:22:38:CB:BA:BA:61:60:7E:E1:0E:3D:39:49:12:74:39:2B:F6:08:7E:7C:72:9B:8F:C6:43:8B:FD:78:BE:68:22:14:37:BC

On peut voir le key exchange (échange de clé) sur Wireshark mais pas la valeur de la clé.

## 6.9 - Quelle est l'algorithme de chiffrement symétrique qui sera utilisé ? :

Parmi les suites cryptographiques proposées dans le Hello Client, le serveur a choisi le chiffrement TLS\_AES\_128\_GCM\_SHA256. Donc l'algorithme de chiffrement symétrique est AES en Galois/Counter Mode avec des blocs de 128bits.

2	2.4262413...	2a01:c...	2a00:1450...	TLSv1...	603	Client Hello
3	2.4540589...	2a00:1...	2a01:cb1d...	TCP	86	443 → 43642 [ACK] Seq=1 Ack=518 Win=66816 Len=0
4	2.4597988...	2a00:1...	2a01:cb1d...	TLSv1...	1294	Server Hello, Change Cipher Spec
5	2.4598224...	2a01:c...	2a00:1450...	TCP	86	43642 → 443 [ACK] Seq=518 Ack=1209 Win=64128 Len=0
6	2.4603071...	2a00:1...	2a01:cb1d...	TLSv1...	1294	Continuation Data
7	2.4603281...	2a01:c...	2a00:1450...	TCP	86	43642 → 443 [ACK] Seq=518 Ack=2417 Win=63744 Len=0

Handshake Protocol: Server Hello  
Handshake Type: Server Hello (2)  
Length: 118  
Version: TLS 1.2 (0x0303)  
Random: 27aebcc6495a4242d9d290b567e00815b6dad9361e497c8e...  
Session ID Length: 32  
Session ID: 8fc67756960bad302e49271123d155ca5e4faf2bf9f1eb11...  
Cipher Suite: TLS\_AES\_128\_GCM\_SHA256 (0x1301)  
Compression Method: null (0)  
Extensions Length: 46

Ce qui est confirmé par la capture de la fenêtre du navigateur (voir introduction), sous « détails techniques ».