



Si on se concentre sur le contenu d'une entête HTTP (brute) et HTTPS ("en clair", telle que vue par le browser une fois le déchiffrement effectué), quelle différence y a-t-il ?

en HTTP, ce n'est pas chiffré tandis que HTTPS les données sont chiffrées.

Donner les ports standards d'écoute des serveurs HTTP et HTTPS

HTTP : 80

HTTPS : 443

## 1.2 Quelles applications utilisent le TLS

Pourquoi ne chiffre-t-on pas l'intégralité d'un paquet contenant du HTTP plutôt qu'uniquement le protocole applicatif ?

Faire cela permet d'éviter une surcharge inutile sur le réseau.

Quelle solution alternative peut-on adopter pour chiffrer les paquets (et non pas simplement le message applicatif) ?

On peut utiliser un VPN s'adaptant au protocole IPsec.

Protocole	Composition	Fonction
SMTPS	SMTP + TLS	Sécurisation des emails via TLS
FTPS	FTP + TLS	Transfert sécurisé de fichiers via TLS
LDAPS	LDAP + TLS	Accès sécurisé au service LDAP via TLS
POP3S	POP3 + TLS	Récupération sécurisée des emails via TLS
SIPS	SIP + TLS	Sécurisation des sessions de protocole SIP via TLS

## 1.3 Les 2 grandes phases d'une communication par TLS

Pour quelle(s) raison(s) TLS utilise-t-il 2 procédés de chiffrement différents (asymétrique puis symétrique) ?

TLS utilise un procédé de chiffrement asymétrique pour créer une clé secrète partagée, puis utilise un chiffrement symétrique pour sécuriser les données échangées.

Lancer Wireshark pendant quelques secondes et se connecter à la page “lemonde.fr” pour enregistrer le trafic généré. Filtrer les enregistrements en renseignant le critère “TLS” et tâcher de retrouver les 2 grandes phases d’une communication par TLS :

18.196.126.151	172.20.37.44	TLSv1.2	1454	Server Hello
18.196.126.151	172.20.37.44	TLSv1.2	1417	Certificate, Server Key Exchange, Server Hello Done
185.64.190.78	172.20.37.44	TLSv1.3	117	Application Data
172.20.37.44	185.255.84.152	TLSv1.3	118	Change Cipher Spec, Application Data
172.20.37.44	18.196.126.151	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
172.20.37.44	185.255.84.152	TLSv1.3	146	Application Data
172.20.37.44	18.196.126.151	TLSv1.2	153	Application Data
172.20.37.44	185.255.84.152	TLSv1.3	1291	Application Data
172.20.37.44	18.196.126.151	TLSv1.2	861	Application Data
172.20.37.44	213.227.153.220	TLSv1.3	1046	Application Data
172.20.37.44	18.193.96.13	TLSv1	596	Client Hello

Toujours sur Firefox, aller sur le site “lemonde.fr” et afficher les informations de sécurité (cadenas / connexion sécurisée / plus d’informations). Dans la section “détails techniques”, que signifie la suite “TLS\_.....\_SHA256” et de quoi est-elle constituée ?

#### Détails techniques

Connexion chiffrée (clés TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256, 128 bits, TLS 1.2)

La page actuellement affichée a été chiffrée avant d’avoir été envoyée sur Internet.

“tls\_ecdhe\_rsa\_with\_aes\_128\_gcm\_sha256” indique une suite de chiffrement utilisant le protocole TLS avec un échange de clés basé sur ECDHE et RSA, un chiffrement symétrique AES-128 en mode GCM, et une intégrité des données assurée par l’algorithme de hachage SHA-256

Se concentrer sur un paquet estampillé “TLSv1.3 - Application Data” “Rentrer” dans la couche la plus haute et commenter les champs contenus :

## 1.4 Qu’est-ce qu’un certificat SSL /TLS ?

Un certificat contient les éléments suivants (4 éléments les plus importants) :

Les 4 éléments les plus importants :

Nom du titulaire du certificat

Numéro de certificat

Date d’émission et date d’expiration

Signature numérique

Quel élément du certificat permet de commencer une communication sécurisée avec le système associé ?

C'est le certificat SSL/TLS de navigateur, c'est ce qui permet de commencer une communication sécurisée.

Un site Web peut-il générer lui-même son propre certificat ? Quelle conséquence cela a-t-il ?

Oui c'est un certificat auto-signé et il n'est pas vérifié par une autorité de certification tiers. On aura l'avertissement de sécurité, : lorsqu'un utilisateur tente de se connecter à un site web utilisant un certificat auto-signé, la plupart des navigateurs afficheront un avertissement

## 1.5 Qu'est-ce qu'une autorité de certification (CA) et une infra à clé publique(PKI) ?

Que doit faire un client pour vérifier l'authenticité d'une signature de certificat ?

L'authenticité d'un certificat repose sur la confiance dans l'autorité de certification (CA) qui a émis le certificat, donc il faut effectuer la vérification de la signature

Comment peut-on être sûr que le CA en question est digne de confiance ?

On peut regarder ça quand on va dans le cadenas et juger de la confiance :

Lecteur du certificat : \*.google.com

The screenshot shows a browser's certificate viewer interface. At the top, there are two tabs: 'Général' (selected) and 'Détails'. Below the tabs, the text 'Émis pour' (Issued for) is followed by a table of certificate details. The table has two columns: the field name and the value. The first row shows 'Nom commun (CN)' with the value '\*.google.com', which is underlined in red. The second row shows 'Organisation (O)' with the value '<Ne fait pas partie du certificat>'. The third row shows 'Unité d'organisation (OU)' with the value '<Ne fait pas partie du certificat>'.

Nom commun (CN)	<u>*.google.com</u>
Organisation (O)	<Ne fait pas partie du certificat>
Unité d'organisation (OU)	<Ne fait pas partie du certificat>

## 2.2 Premières manipulations et observation du trafic

Récupérer la capture wireshark effectuée sur HMI2 et montrer la structure des échanges ci-dessous :

No.	Time	Source	Destination	Protocol	Len
1	0.000000000	02:42:ac:19:00:05	Broadcast	ARP	
2	0.000107954	02:42:ac:19:00:06	02:42:ac:19:00:05	ARP	
3	0.000119918	172.25.0.5	172.25.0.6	TCP	
4	0.000196432	172.25.0.6	172.25.0.5	TCP	
5	0.000228150	172.25.0.5	172.25.0.6	TCP	
6	0.000375891	172.25.0.5	172.25.0.6	TCP	
7	0.000627969	172.25.0.6	172.25.0.5	TCP	
8	0.000735644	172.25.0.5	172.25.0.6	TCP	
9	0.000952665	172.25.0.6	172.25.0.5	TCP	
10	0.000972419	172.25.0.5	172.25.0.6	TCP	
11	5.104278750	02:42:ac:19:00:06	02:42:ac:19:00:05	ARP	
12	5.104297485	02:42:ac:19:00:05	02:42:ac:19:00:06	ARP	

Récupérer la capture wireshark effectuée sur HMI1 et montrer la structure des échanges ci-dessous :

172.25.0.4	172.25.0.3	TCP	74 10023 → 51374 [SYN, A
172.25.0.3	172.25.0.4	TCP	66 51374 → 10023 [ACK] S
172.25.0.3	172.25.0.4	TLSv1.2	583 Client Hello
172.25.0.4	172.25.0.3	TCP	66 10023 → 51374 [ACK] S
172.25.0.4	172.25.0.3	TLSv1.2	4162 Server Hello
172.25.0.3	172.25.0.4	TCP	66 51374 → 10023 [ACK] S
172.25.0.4	172.25.0.3	TLSv1.2	563 Certificate
172.25.0.3	172.25.0.4	TCP	66 51374 → 10023 [ACK] S
172.25.0.3	172.25.0.4	TLSv1.2	4162 Certificate, Client K
172.25.0.4	172.25.0.3	TCP	66 10023 → 51374 [ACK] S
172.25.0.3	172.25.0.4	TLSv1.2	207 Certificate Verify
172.25.0.4	172.25.0.3	TLSv1.2	1348 New Session Ticket, C
172.25.0.3	172.25.0.4	TLSv1.2	100 Application Data

Sur quel port s'effectuent les communications entre client et serveur ?

Port 51374 et 10023

Y a-t-il eu des échanges entre le client et le CA comme on aurait pu s'y attendre ?  
Pourquoi ?

Oui on peut voir qu'il y a eu un échange entre le client et le CA.

Avec le script client\_ssl, tenter d'envoyer un message HMI1 → PLC2 (server\_ssl).  
Recueillir la capture WS et les messages sur les 2 systèmes. Conclure

```
admin@plc2:~$ ./server
Connection from ('172.25.0.3', 52206)
Received: [hex data]
=5</Aih9876' '# [hex data] )% [hex data] @?>3210 [hex data] EDCB
[hex data]
[hex data]
```

Le message est chiffré je peux pas voir le contenu

Avec le script client\_ssl, tenter d'envoyer un message HMI2 → PLC1 (server\_ssl).  
Recueillir la capture WSet les messages sur les 2 systèmes. Conclure

```
admin@plc1:~$ ./server_ssl
SSL error, ignore connection request from ('172.25.0.5',
[hex data])
```

Erreur SSL

## 2.3 Génération des clés et des certificats et pour HMI2 et PLC2 sur le CA

### 1. Génération d'une clé RSA privée pour le serveur PLC2

```
admin@ca:~/ca$ openssl genrsa -out intermediate/private/plc2.example.com.key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....
.....+++
.....+++
e is 65537 (0x10001)
admin@ca:~/ca$
```

```
admin@ca:~/ca$ chmod 400 intermediate/private/plc2.example.com.key.pem
admin@ca:~/ca$
```

### 2. Génération d'une requête de signature pour l'AC ( PLC2 )

```
admin@ca:~/ca$ openssl req -config intermediate/openssl.cnf -key intermediate/private/plc2.example.com.key.pem -subj '/CN=plc2.example.com/O=Example./C=US/ST=CA' -new -sha256 -out intermediate/csr/plc2.example.com.csr.pem
admin@ca:~/ca$
```

### 3. Signature de la requête par l'AC et obtention d'un certificat signé ( PLC2 )

```
admin@ca:~/ca$ openssl ca -batch -config intermediate/openssl.cnf -extensions server_cert -days 375 -notext -md sha256 -in intermediate/csr/plc2.example.com.csr.pem -out intermediate/certs/plc2.example.com.cert.pem
Using configuration from intermediate/openssl.cnf
Check that the request matches the signature
Signature ok
```

### 1. Génération d'une clé RSA privée pour le serveur HMI2

```
admin@ca:~/ca$ openssl genrsa -out intermediate/private/hmi2.example.com.key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
admin@ca:~/ca$ chmod 400 intermediate/private/hmi2.example.com.key.pem
admin@ca:~/ca$
```

### 2. Génération d'une requête de signature pour l'AC ( HMI2 )

```
admin@ca:~/ca$ openssl req -config intermediate/openssl.cnf -key intermediate/private/hmi2.example.com.key.pem -subj '/CN=hmi2.example.com/O=Example./C=US/ST=CA' -new -sha256 -out intermediate/csr/hmi2.example.com.csr.pem
admin@ca:~/ca$
```

### 3. Signature de la requête par l'AC et obtention d'un certificat signé ( HMI2 )

```
admin@ca:~/ca$ openssl ca -batch -config intermediate/openssl.cnf -days 375 -notext -md sha256 -in intermediate/csr/hmi2.example.com.csr.pem -out intermediate/certs/hmi2.example.com.cert.pem
Using configuration from intermediate/openssl.cnf
Check that the request matches the signature
Signature ok
```

Entrez les commandes utilisées ici :

```
openssl genrsa -out intermediate/private/plc2.example.com.key.pem 2048
chmod 400 intermediate/private/plc2.example.com.key.pem
```

```
openssl req -config intermediate/openssl.cnf -key
intermediate/private/plc2.example.com.key.pem -subj
'/CN=plc2.example.com/O=Example./C=US/ST=CA' -new -sha256 -out
intermediate/csr/plc2.example.com.csr.pem
```

```
openssl ca -batch -config intermediate/openssl.cnf -extensions server_cert -days 375
-notext -md sha256 -in intermediate/csr/plc2.example.com.csr.pem -out
intermediate/certs/plc2.example.com.cert.pem
```

Est-il nécessaire pour un client d'obtenir un certificat ?

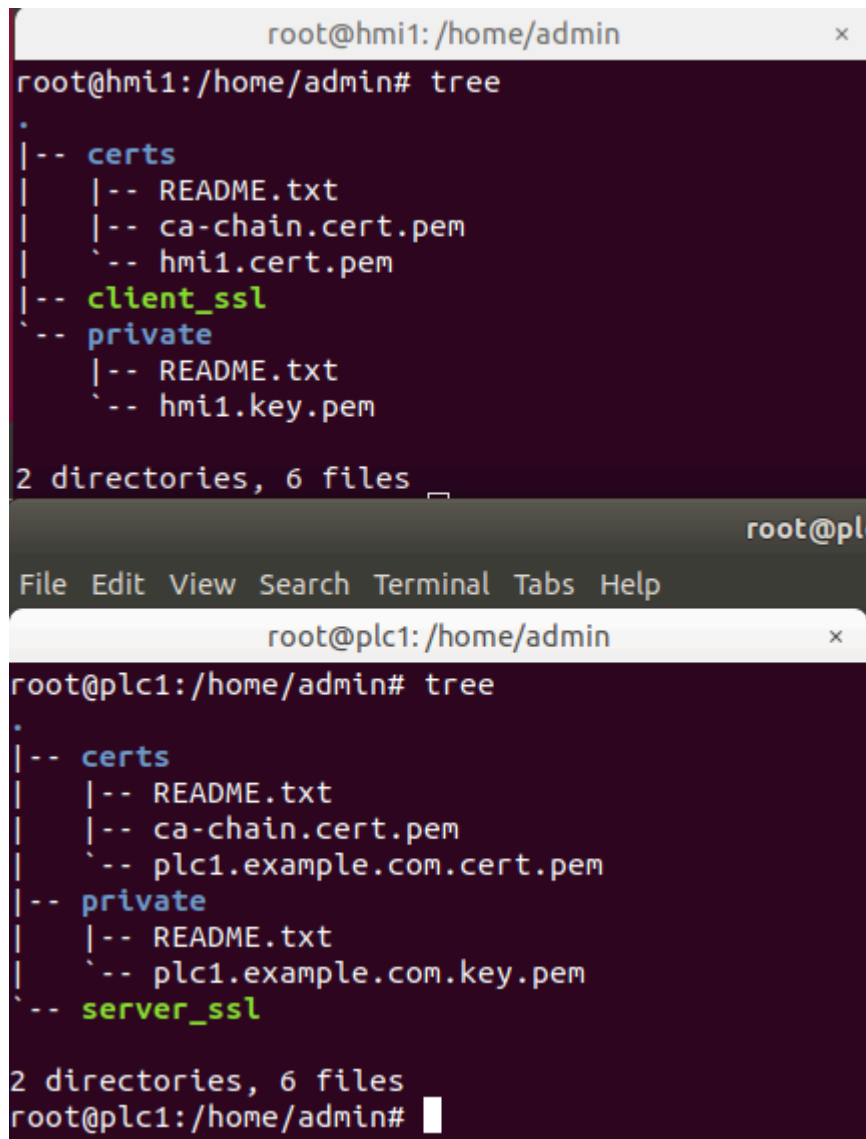
Non, c'est pour avoir accès à un environnement sécurisé et ne pas avoir cette erreur : Votre connexion n'est pas sécurisé

Est-ce le rôle du CA de générer les certificats (non signés) ?

La génération d'un certificat non signé peut être effectuée par n'importe qui sans aucune vérification. Donc n'importe qui peut se faire passer pour n'importe qui en utilisant un certificat non signé.

## 2.4 Transfert des clés et des certificats sur les hôtes HMI2 et PLC2

Faire un tree des répertoires “home” de PLC1 et HMI1 pour faire un état des lieux des types de fichiers contenus dans les répertoires



The image shows two terminal windows side-by-side. The top window is titled 'root@hmi1: /home/admin' and displays the output of the 'tree' command. It shows a directory structure with 'certs' and 'private' subdirectories. The 'certs' directory contains 'README.txt', 'ca-chain.cert.pem', and 'hmi1.cert.pem'. The 'private' directory contains 'README.txt' and 'hmi1.key.pem'. The bottom window is titled 'root@plc1: /home/admin' and also displays the output of the 'tree' command. It shows a similar directory structure with 'certs' and 'private' subdirectories. The 'certs' directory contains 'README.txt', 'ca-chain.cert.pem', and 'plc1.example.com.cert.pem'. The 'private' directory contains 'README.txt' and 'plc1.example.com.key.pem'. Both windows indicate '2 directories, 6 files'.

```
root@hmi1: /home/admin
root@hmi1:/home/admin# tree
.
|-- certs
|   |-- README.txt
|   |-- ca-chain.cert.pem
|   `-- hmi1.cert.pem
|-- client_ssl
`-- private
    |-- README.txt
    `-- hmi1.key.pem

2 directories, 6 files

root@plc1: /home/admin
root@plc1:/home/admin# tree
.
|-- certs
|   |-- README.txt
|   |-- ca-chain.cert.pem
|   `-- plc1.example.com.cert.pem
|-- private
|   |-- README.txt
|   `-- plc1.example.com.key.pem
`-- server_ssl

2 directories, 6 files
root@plc1:/home/admin#
```

Effectuer des transferts par SCP des clés (fichiers .key.pem) et des certificats signés (fichiers .cert.pem) vers les hôtes concernés. Copier les commandes SCP ci-dessous :



Effectuer des transferts par SCP des clés (fichiers .key.pem) et des certificats signés (fichiers .cert.pem) vers les hôtes concernés. Copier les commandes SCP ci-dessous

```
admin@172.25.0.6's password:
plc2.example.com.key.pem                                100% 1675      1.6KB/s   00:00
admin@ca:~/ca/intermediate/private$ scp hmi2.key.pem admin@172.25.0.5:
admin@172.25.0.5's password:
Permission denied, please try again.
admin@172.25.0.5's password:
Permission denied, please try again.
admin@172.25.0.5's password:
hmi2.key.pem                                            100% 1679      1.6KB/s   00:00
admin@ca:~/ca/intermediate/private$ scp hmi2.key.pem admin@172.25.0.5:private

admin@ca:~/ca/intermediate/certs$ scp ca-chain.cert.pem admin@172.25.0.6:certs
admin@172.25.0.6's password:
ca-chain.cert.pem                                    100% 3904      3.8KB/s   00:00
admin@ca:~/ca/intermediate/certs$ scp ca-chain.cert.pem admin@172.25.0.5:certs
admin@172.25.0.5's password:
ca-chain.cert.pem                                    100% 3904      3.8KB/s   00:00
admin@ca:~/ca/intermediate/certs$ scp hmi2.cert.pem admin@172.25.0.5:certs
admin@172.25.0.5's password:
hmi2.cert.pem                                         100%    0      0.0KB/s   00:00
admin@ca:~/ca/intermediate/certs$ scp plc2.example.com.cert.pem admin@172.25.0.6:certs
admin@172.25.0.6's password:
plc2.example.com.cert.pem                            100% 1834      1.8KB/s   00:00
```

## 2.5 Tests de communications sécurisées client-serveur

**HMI1 → PLC1**

**HMI1 → PLC2**

**HMI2 → PLC1**

**HMI2 → PLC2**

dans toutes les communications on a une requete et une réponse d'un certificat valide grâce au protocole TLS.