



TP 3.3-1

Certificats Web et SSH

Rendu attendu

Une infrastructure fonctionnelle complète, montrée à votre intervenant pour notation à chaud.

Il est conseillé de faire une lecture complète de ce support, avant de commencer les actions.

Objectifs de ce TP

L'objectif de ce travail pratique est de vous initier à la manipulation de différents types de certificats, notamment les certificats Web et SSH. La configuration de ces certificats vise à renforcer la sécurité de votre infrastructure informatique. D'une part, elle assure le chiffrement des données en transit, garantissant ainsi la confidentialité des échanges. D'autre part, elle permet de restreindre l'accès au serveur SSH en autorisant uniquement une machine spécifique à s'y connecter, éliminant ainsi la nécessité de saisir un mot de passe à chaque connexion.

Table des matières

Préambule.....	2
1. Visualisation du flux en clair	3
2. Mise en place du certificat TLS	6
1) Génération des certificats.....	7
2) Configuration d'Apache2	8
3) Redirection automatique vers un flux HTTPS	11
4) Amélioration de la sécurité.....	12
5) Mise en place du certificat dans le client Web	12
Modification du fichier HOSTS.....	12
Régénération des certificats	13
Qualifier ce certificat comme certificat de confiance (Google Chrome).....	15
6) Lecture des flux chiffrés	16
3. Clé SSH	16
1) Connexion SSH Standard.....	17
2) Connexion par clé SSH	17
Génération de la clé.....	18
Dépose de la clé sur le serveur	18



Pour réaliser ce TP, vous devrez [posséder le logiciel de virtualisation VirtualBox](#) et l'utiliser.

Assurez-vous d'avoir le logiciel sur votre poste, et d'être à l'aise dans son utilisation.

Dans le cas contraire, n'hésitez pas à vous [référer à la documentation](#) du logiciel.

Les fichiers .iso pour ce TP sont généralement fournis par votre intervenant.

Préambule

Pour ce travail, nous allons continuer sur les bases posées lors du **TP chapitre 1 : « Applications PowerShell et Sauvegardes »**.

Pour cela, démarrez votre VM Debian Apache2 + PHP et vérifiez que votre site soit toujours fonctionnel. Si ce n'est pas le cas, troublshootez. Aucune modification Hardware n'est nécessaire. Une Debian et un site en parfait état de marche sont nécessaires pour ce TP.

Pour rappels, utilisez SSH pour plus de facilité ; utilisez la commande `ip a` pour savoir l'adresse IP de votre machine. En renseignant l'URL de votre Debian : http://<IP_Debian> vous devriez avoir ce rendu :

Connexion au compte

Bienvenue sur la plateforme de gestion des utilisateurs, sans plus attendre, connectez-vous pour avoir les informations de votre compte.

Adresse e-mail :

Mot de passe :

Connexion

Votre directeur, Pierre MARTINEZ, souhaite améliorer la sécurité de ce site de gestion des comptes, actuellement en production. Il vous demande plusieurs choses : Est-ce vrai que les connexions HTTP ne sont pas sécurisées ? Ne pouvez-vous pas mettre une meilleure méthode de connexion SSH ?

Vous vous mettez immédiatement au travail !

1. Visualisation du flux en clair

Si nous disons que les flux HTTP ne sont pas du tout confidentiels, ce n'est pas pour rien ! Vous allez le voir par vous-même.

Il existe plusieurs outils qui permettent de capturer les trames réseau transitant dans un LAN. Nous allons aujourd'hui en utiliser deux : TCPDUMP (CLI) et [WireShark](#) (GUI).

Partie cours

Capturer des trames réseau consiste à intercepter et à enregistrer les paquets de données qui transitent sur un réseau informatique. Chaque paquet de données contient des informations telles que l'adresse source et de destination, le type de protocole utilisé, ainsi que les données proprement dites.

En capturant ces trames, on peut analyser le trafic réseau pour diverses raisons, telles que le dépannage, la surveillance de la performance du réseau, la détection d'anomalies ou de tentatives d'intrusion.

- À l'aide de TCPDUMP, capturez le flux réseau entre votre poste hôte et votre VM Debian, lors de l'envoi d'informations, via la mire de connexion (pas besoin de renseigner un mot de passe valide).

- Installez TCPDUMP si votre Debian ne le possède pas :

```
apt update -y && apt install tcpdump -y
```

Aides

- AB Pour voir l'adresse IP de votre machine hôte, ouvrez un invite de commande Windows puis renseignez la commande `ipconfig` ;
- AB Pour voir l'adresse IP de votre serveur Debian et le nom de votre interface réseau, renseignez la commande `ip a` ;
- AB La commande TCPDUMP se forme ainsi :

```
tcpdump -i <interface> -n host <IP_Source> and host <IP_Destination> and port 80
```

Rappel : Utilisez le mot « sudo » devant la commande si vous ne travaillez pas avec « root ».

Étapes

1. Entrez la commande TCPDUMP et lancez la capture ;
2. Entrez un login / mot de passe sur votre site (pas nécessairement valide) ;
3. Envoyez les données depuis votre site Web : cliquez sur « Connexion » ;
4. Arrêtez la capture TCPDUMP avec **Ctrl+C**.

Après ces actions effectuées, vous devriez avoir ce type de résultats :

```
root@tp-ps-svgr:~# tcpdump -i enp0s3 -n host 192.168.1.50 and host 192.168.1.63 and port 80
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
10:58:52.598216 IP 192.168.1.50.52485 > 192.168.1.63.80: Flags [S], seq 4050478268, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], ack 4050478269, win 64240, options [mss 1460,nop,nop,sackOK]
10:58:52.598245 IP 192.168.1.63.80 > 192.168.1.50.52485: Flags [S.], seq 2606898711, ack 4050478269, win 64240, options [mss 1460,nop,nop,sackOK]
10:58:52.598601 IP 192.168.1.50.52485 > 192.168.1.63.80: Flags [.], ack 1, win 513, length 0
10:58:52.598602 IP 192.168.1.50.52485 > 192.168.1.63.80: Flags [.], ack 1, win 513, length 645: HTTP: POST /login.php HTTP/1.1
10:58:52.598626 IP 192.168.1.63.80 > 192.168.1.50.52485: Flags [.], ack 646, win 501, length 0
10:58:52.600053 IP 192.168.1.50.52486 > 192.168.1.63.80: Flags [S], seq 4275603512, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sackOK], ack 995867886, win 64240, options [mss 1460,nop,nop,sackOK]
10:58:52.600053 IP 192.168.1.63.80 > 192.168.1.50.52486: Flags [S.], seq 995867886, ack 4275603513, win 64240, options [mss 1460,nop,nop,sackOK]
10:58:52.600375 IP 192.168.1.50.52486 > 192.168.1.63.80: Flags [.], ack 1, win 513, length 0
10:58:52.604865 IP 192.168.1.63.80 > 192.168.1.50.52485: Flags [P.], seq 1:535, ack 646, win 501, length 534: HTTP: HTTP/1.1 200 OK
10:58:52.621403 IP 192.168.1.50.52485 > 192.168.1.63.80: Flags [P.], seq 646:1095, ack 535, win 511, length 449: HTTP: GET /login.css HTTP/1.1
10:58:52.622925 IP 192.168.1.63.80 > 192.168.1.50.52485: Flags [P.], seq 535:1312, ack 1095, win 501, length 777: HTTP: HTTP/1.1 200 OK
10:58:52.663005 IP 192.168.1.50.52485 > 192.168.1.63.80: Flags [.], ack 1312, win 508, length 0
10:58:57.627421 IP 192.168.1.63.80 > 192.168.1.50.52485: Flags [F.], seq 1312, ack 1095, win 501, length 0
10:58:57.627688 IP 192.168.1.50.52485 > 192.168.1.63.80: Flags [.], ack 1313, win 508, length 0
^C
14 packets captured
14 packets received by filter
0 packets dropped by kernel
root@tp-ps-svgr:~#
```

Si vous avez ce résultat, vous avez réussi à sniffer votre réseau correctement lors de la connexion.

Oui ! Sniffer un réseau aussi simplement, ça fait peur ! Et vous n'avez pas encore tout vu ! 😊

Explications (via le Screenshot ci-dessus)

- AB Ligne 1 : Renseignement de la commande initiale ;
- AB Ligne 7 : Envoie des éléments renseignés à notre page de traitement : login.php ;
- AB Ligne 12 : Connexion HTTP success → Code 200 OK ;
- AB Ligne 13 : Récupération du fichier CSS de la page « login » (chargement des éléments en cache) ;
- AB Ligne 14 : Connexion HTTP success → Code 200 OK.

L'affichage de TCPDUMP ne nous permet pas de visualiser plus d'informations. Vous allez pourtant voir qu'il cache bien son jeu !

Pour voir davantage d'informations, il faut exporter la sortie de TCPDUMP dans un fichier au format .pcap puis lire ce fichier avec le logiciel [WireShark](#) (à télécharger et installer si nécessaire).

- À l'aide de (presque) la même commande TCPDUMP, recapturez le flux réseau entre votre poste hôte et votre VM Debian, lors de l'envoi d'informations, via la mire de connexion.

Ajoutez `-w capture.pcap` en fin de commande pour rediriger la sortie de la commande dans ce fichier.

- Utilisez la commande pour voir le contenu de votre fichier .pcap :

```
tcpdump -r capture.pcap
```

Après ces nouvelles actions effectuées, vous devriez avoir ce type de résultat :

```
root@tp-ps-svg:~# tcpdump -i enp0s3 -n host 192.168.1.50 and host 192.168.1.63 and port 80 -w capture.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C10 packets captured
0 packets dropped by kernel
root@tp-ps-svg:~# tcpdump -r capture.pcap
reading from file capture.pcap, link-type EN10MB (Ethernet), snapshot length 262144
11:29:53.510436 IP lenovohc.home.60079 > tp-ps-svg-1.home.http: Flags [S], seq 1830377497, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sack0
11:29:53.510465 IP tp-ps-svg-1.home.http > lenovohc.home.60079: Flags [S.], seq 4250514688, ack 1830377498, win 64240, options [mss 1460,nop,nop,sack0
11:29:53.511556 IP lenovohc.home.60079 > tp-ps-svg-1.home.http: Flags [.], ack 1, win 513, length 0
11:29:53.511558 IP lenovohc.home.60079 > tp-ps-svg-1.home.http: Flags [P.], seq 1:669, ack 1, win 513, length 668: HTTP: POST /login.php HTTP/1.1
11:29:53.511558 IP lenovohc.home.60080 > tp-ps-svg-1.home.http: Flags [S.], seq 2852403733, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sack0
11:29:53.511596 IP tp-ps-svg-1.home.http > lenovohc.home.60079: Flags [.], ack 669, win 501, length 0
11:29:53.511833 IP tp-ps-svg-1.home.http > lenovohc.home.60080: Flags [S.], seq 1747096327, ack 2852403734, win 64240, options [mss 1460,nop,nop,sack0
11:29:53.512675 IP lenovohc.home.60080 > tp-ps-svg-1.home.http: Flags [.], ack 1, win 513, length 0
11:29:53.513695 IP tp-ps-svg-1.home.http > lenovohc.home.60079: Flags [P.], seq 1:535, ack 669, win 501, length 534: HTTP: HTTP/1.1 200 OK
11:29:53.555010 IP lenovohc.home.60079 > tp-ps-svg-1.home.http: Flags [.], ack 535, win 511, length 0
root@tp-ps-svg:~#
```

- Téléchargez via FTP/SFTP sur le bureau de la machine hôte le fichier « capture.pcap », puis, ouvrez-le avec Wireshark.

Triez par protocole et concentrez-vous sur le protocole HTTP.

J'ai personnellement 2 lignes HTTP :

Source	Destination	Protocol	Length	Info
192.168.1.50	192.168.1.63	HTTP	722	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
192.168.1.63	192.168.1.50	HTTP	588	HTTP/1.1 200 OK (text/html)
192.168.1.50	192.168.1.63	TCP	66	60079 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
192.168.1.63	192.168.1.50	TCP	66	80 → 60079 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
192.168.1.50	192.168.1.63	TCP	60	60079 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
192.168.1.50	192.168.1.63	TCP	66	60080 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
192.168.1.63	192.168.1.50	TCP	54	80 → 60079 [ACK] Seq=1 Ack=669 Win=64128 Len=0
192.168.1.63	192.168.1.50	TCP	66	80 → 60080 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
192.168.1.50	192.168.1.63	TCP	60	60080 → 80 [ACK] Seq=1 Ack=1 Win=131328 Len=0
192.168.1.50	192.168.1.63	TCP	60	60079 → 80 [ACK] Seq=669 Ack=535 Win=130816 Len=0

- Trouvez et sélectionnez la ligne qui correspond à l'envoi POST des informations sur la page « login.php » (puisque c'est comme ça que fonctionne notre formulaire).

Dépliez la ligne concernant les paramètres encodés dans le formulaire : « **HTML Form URL Encoded** » :

SURPRISE ! 🤫

```

    ▼ HTML Form URL Encoded: application/x-www-form-urlencoded
      > Form item: "email" = "toto@tutu.lu"
      > Form item: "password" = "MrBr4s&Gen!a1"
  
```

Imaginez donc les découvertes que vous pourriez faire, si vous scannez et faites une capture de trames sur un réseau public (aéroports, restaurants, hôtels, campings...) et que d'autres utilisateurs utilisent des flux non chiffrés...

	<p>Disclaimer : Art. 323-1 du CP : « <i>Le fait d'accéder ou de se maintenir, frauduleusement, dans tout ou partie d'un système de traitement automatisé de données est puni de trois ans d'emprisonnement et de 100 000 € d'amende.</i> »</p> <p>Le contenu de ce Travail Pratique est strictement éducatif afin de vous faire réfléchir sur les menaces existantes pour mieux vous en protéger.</p> <p>Votre intervenant n'est pas responsable des actions que vous mènerez en dehors du cours. Respectez les lois, obtenez des autorisations et utilisez vos compétences de manière éthique et légale.</p>
--	--

Limitez donc au maximum votre utilisation de ces réseaux publics, au profit de réseaux personnels, tels que le partage de connexion depuis son mobile par exemple.

	<p>Jalonnement : Avant de passer à la partie suivante,appelez votre intervenant pour un contrôle intermédiaire de votre travail réalisé jusqu'à présent. Assurez-vous que tout soit fonctionnel avant son arrivée.</p>
--	---

2. Mise en place du certificat TLS

- Éteignez votre VM Debian et faites un clone intégral, via le mode expert, de votre machine virtuelle en générant de nouvelles adresses MAC. Rajoutez « -TLS » derrière le nom pour être propre, choisissez proprement le dossier de stockage du nouveau disque.

Ajustez les propriétés Hardware de vos 2 machines virtuelles pour qu'elles puissent fonctionner en simultanée (CPU et RAM).

Renommez votre machine virtuelle conformément à la documentation « Installation Debian ».

Sous VirtualBox, utilisez les groupes si nécessaires.

1) Génération des certificats

Sous Apache2, il existe une méthode simple pour mettre en place les certificats, étant donné qu'ils sont déjà préconfigurés. Cependant, nous allons générer nos propres certificats. Pour mener à bien cette opération, nous allons utiliser le paquet « openssl ».

- Démarrez votre nouvelle VM Debian -TLS et renommez-la conformément à la procédure.
- Mettez à jour la liste des paquets :

```
apt update -y
```

- Installez le paquet openssl :

```
apt install openssl -y
```



Dans les commandes que vous passez, vous devez voir l'utilisation régulière du terme « `-y` ». Ce terme permet d'accepter automatiquement les questions posées de sorte que les mises à jour ou l'installation se déroulent sans action utilisateur. C'est pratique pour les scripts. En revanche, ce n'est normalement pas une bonne pratique lors d'installations manuelles.



Vous le savez, un certificat authentique doit être émis et validé par une autorité de certification (CA) telle que Verisign, Thawte ou CertiSign. Si vous n'avez pas votre propre CA, vous devrez ajouter celle-ci à vos navigateurs pour éviter les avertissements. Dans notre cas, nous allons créer un certificat auto-signé, ce qui signifie qu'il n'a pas été vérifié par une autorité tierce. Bien que gratuit et suffisant pour un usage personnel, ce type de certificat ne garantit pas le même niveau de sécurité, car n'importe qui peut en créer un, ce qui ne permet pas de vérifier l'identité de l'émetteur.

- Créez un nouveau répertoire « `custom` » dans `/etc/ssl/` :

```
mkdir /etc/ssl/custom
```

- Générez votre propre certificat avec la commande suivante (amusez-vous à l'adapter au besoin). Renseignez ensuite les champs demandés un par un :

```
openssl req -x509 -nodes -days 365 -newkey rsa:4096 -sha256 -out  
/etc/ssl/custom/gest_util.crt -keyout /etc/ssl/custom/gest_util.key
```

Explications de la commande

- AB **openssl** : Outil en ligne de commande pour la gestion, la création de nouveaux certificats, clés et autres fichiers OpenSSL ;
 - AB **req -x509** : Demande de signature de certificat au format CSR. Normes d'infrastructure de clé publique à laquelle SSL et TLS adhèrent ;
 - AB **-nodes** : Ne pas utiliser de passphrase. Nous avons besoin qu'Apache2 soit capable de lire le fichier ;
 - AB **-days 365** : Période de validité du certificat, ici 365 jours, soit 1 an ;
 - AB **-newkey rsa:4096** : Génération d'un nouveau certificat et d'une nouvelle clé privée en même temps RSA de 4 096 bits ;
 - AB **-sha256** : Utilisation de SHA-256 plutôt que MD5.
 - AB **-out** : chemin de sortie du certificat public ;
 - AB **-keyout** : chemin de sortie de la clé privée.

À l'issue, vous deviez avoir un résultat comme celui-ci :

```
root@tp-ps-svg-TLS:/etc/ssl# openssl req -x509 -nodes -days 365 -newkey rsa:4096
...+...+.....+.....+..+.....+.....+.....+..+..+..+..+.....+..+....+
+..+.....+..+.....+.....+..+.....+.....+.....+.....+..+.....+..+
+++++++=*+....+.....+..+..+.....+.....+..+.....+..+.....+.....+
....+.....+.....+..+..+....+..+..+.....+..+..+..+.....+..+..+
++++*...+..+.....+.....+.....+.....+.....+.....+.....+.....+.....+
..+..+..+..+.....+.....+..+.....+..+..+..+.....+.....+.....+
.....+.....+.....+.....+.....+.....+.....+.....+.....+.....+
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:HERAULT
Locality Name (eg, city) []:MONTPELLIER
Organization Name (eg, company) [Internet Widgits Pty Ltd]:OPEN IT
Organizational Unit Name (eg, section) []:CYBERSECURITE
Common Name (e.g. server FQDN or YOUR name) []:BRES
Email Address []:contact@avmerick-bres.fr
```

➤ Positionnez seulement les droits de lecture sur le fichier .crt :

```
chmod 440 /etc/ssl/custom/gest util.crt
```

2) Configuration d'Apache2

- Créez le fichier « `gest_util-TLS.conf` » dans le dossier `/etc/apache2/sites-available/` en partant de ce modèle. Adaptez-le en fonction de vos besoins !

```
# Déclaration d'un VirtualHost pour un site web sur le port 443 (HTTPS)
<VirtualHost *:443>
    # Adresse e-mail de l'administrateur du serveur
    ServerAdmin webmaster@example.com

    # Nom et/ou adresse IP du serveur virtuel
    ServerName 192.168.1.67
    ServerAlias www.example.com exemple.com

    # Emplacement des fichiers du site web
    DocumentRoot /var/www/gest_util

    # Configuration SSL
    SSLEngine on
    SSLCertificateFile /chemin/vers/certificat.crt
    SSLCertificateKeyFile /chemin/vers/clé_privée.key
    # SSLCertificateChainFile /chemin/vers/chaîne_intermédiaire.crt

    # Autoriser l'accès aux fichiers dans le répertoire DocumentRoot
    <Directory /var/www/gest_util>
        Require all granted
        AllowOverride FileInfo
    </Directory>

    # Journalisation
    ErrorLog ${APACHE_LOG_DIR}/gest_util_TLS_error.log
    CustomLog ${APACHE_LOG_DIR}/gest_util_TLS_access.log combined
</VirtualHost>
```

	L'entrée « <code>SSLCertificateChainFile</code> » permet de renseigner un certificat .crt provenant d'une autorité de certification tierce, cela ne nous concerne pas ici, vous pouvez commenter la ligne pour qu'elle ne soit pas prise en compte.
--	---

	Ctrl + O permet d'enregistrer votre travail sous Nano. Ctrl + X permet de quitter Nano. La commande <code>apache2ctl configtest</code> permet de vérifier la syntaxe correcte de votre fichier de configuration.
--	--

- Démarrez le module SSL pour Apache :

```
a2enmod ssl
```

- Créez maintenant un lien symbolique avec la commande **ln** depuis le dossier « sites-available » vers « sites-enabled » pour demander à Apache d'activer votre site, c'est-à-dire de le mettre en ligne :

```
ln -s /etc/apache2/sites-available/gest_util-TLS.conf
/etc/apache2/sites-enabled/
```

Voici le résultat attendu :

```
root@tp-ps-svg-TLS:/etc/apache2/sites-available# ln -s /etc/apache2/sites-available/gest_util-TLS.conf /etc/apache2/sites-enabled/
root@tp-ps-svg-TLS:/etc/apache2/sites-available# ln -s /etc/apache2/sites-available/gest_util.conf /etc/apache2/sites-enabled/
root@tp-ps-svg-TLS:/etc/apache2/sites-available# ls -l ..sites-enabled/
total 0
lrwxrwxrwx 1 root root 35 9 févr. 11:54 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root 43 6 mai 10:47 gest_util.conf -> /etc/apache2/sites-available/gest_util.conf
lrwxrwxrwx 1 root root 47 6 mai 10:47 gest_util-TLS.conf -> /etc/apache2/sites-available/gest_util-TLS.conf
```

- Redémarrez Apache2 pour prendre en compte nos modifications :

```
systemctl restart apache2
```

- Accédez à votre site en utilisant le préfixe HTTPS : https://<IP_Debian>



Lors de votre première visite en HTTPS, vous remarquerez un message vous indiquant que votre connexion n'est pas privée. Vous pouvez malgré tout choisir d'accéder au site internet. C'est parfaitement normal, vous le savez : votre client web tente d'identifier la CA à l'origine du certificat. Cependant, comme il s'agit d'un certificat auto-signé par vos soins, votre client ne parvient pas à authentifier le certificat et vous délivre ce message d'erreur. Le flux est cependant bien chiffré.

Vérifiez les informations de votre certificat :

Lecteur du certificat : BRES

Général Détails

Émis pour

Nom commun (CN)	BRES
Organisation (O)	OPEN IT
Unité d'organisation (OU)	CYBERSECURITE

Émis par

Nom commun (CN)	BRES
Organisation (O)	OPEN IT
Unité d'organisation (OU)	CYBERSECURITE

Durée de validité

Émis le	vendredi 9 février 2024 à 12:54:15
Expire le	samedi 8 février 2025 à 12:54:15

Empreintes SHA-256

Certificat	1200ce1961e5f3c83472a1d1089816bfd18a742e2fc00337505db64038 d382d
Clé publique	52320a4d89e52c1a92fcc75e0e09f0da411fa5a096222614b87e9326f496 c3cc



Jalonnement : Avant de passer à la partie suivante, appelez votre intervenant pour un contrôle intermédiaire de votre travail réalisé jusqu'à présent. Assurez-vous que tout soit fonctionnel avant son arrivée.

3) Redirection automatique vers un flux HTTPS

Dans l'état actuel des choses, vous remarquerez que votre site est aussi et toujours disponible en HTTP. Après tant d'efforts, nous allons vouloir rediriger tout le trafic sur notre site en HTTPS !

- Supprimez le lien symbolique de votre fichier de configuration non-HTTPS :

```
rm /etc/apache2/sites-enabled/gest_util.conf
```

- Dans votre fichier de configuration « *gest_util-TLS.conf* », ajoutez le VirtualHost suivant au-dessus de la configuration *:443 :

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias www.example.com

    # Redirection permanente vers HTTPS
    Redirect 301 / https://www.example.com/
</VirtualHost>
```



La directive **301** est une instruction de redirection HTTP utilisée pour indiquer aux navigateurs web et aux moteurs de recherche qu'une ressource a été définitivement déplacée vers une nouvelle URL. Le code de statut HTTP 301 "Moved Permanently" est renvoyé avec cette redirection.



Ctrl + O permet d'enregistrer votre travail sous Nano. Ctrl + X permet de quitter Nano. La commande **apache2ctl configtest** permet de vérifier la syntaxe correcte de votre fichier de configuration.

- Reload le service Apache2 :

```
systemctl reload apache2
```

- **Essayez de refaire une connexion HTTP simplement.** Normalement ça devrait être impossible.

4) Amélioration de la sécurité

- **Avec les failles connues et décrites dans ce cours, nous allons apporter une dernière modification à notre configuration d'Apache2. Rendez-vous dans le fichier de configuration :**

```
nano /etc/apache2/apache2.conf
```

Puis ajoutez ces lignes en fin de fichier et redémarrez Apache2 :

```
<IfModule ssl_module>
    SSLProtocol -ALL +TLSv1 +TLSv1.1 +TLSv1.2
    SSLHonorCipherOrder On
    SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-
        SHA256:HIGH:!MD5:!aNULL:!EDH:!RC4
    SSLCompression off
</IfModule>
```

- AB **SSLProtocol** : Activation uniquement du TLS ;
- AB **SSLHonorCipherOrder** : Le serveur Web doit honorer l'ordre de préférence des algorithmes de chiffrement fournis par le client. Cela renforce la sécurité en prévenant les attaques de type BEAST ;
- AB **SSLCipherSuite** : Définit la liste des algorithmes de chiffrement autorisés avec un ordre de préférence pour leur utilisation ;
- AB **SSLCompression** : Désactive la compression TLS considérée comme vulnérable.

5) Mise en place du certificat dans le client Web

Modification du fichier HOSTS

Il est possible de supprimer le message d'erreur « Votre connexion n'est pas privée » en disant à notre client Web, que le certificat TLS auto-signé par vos soins est digne de confiance. Pour ce faire, nous utiliserons le magasin de certificat Windows.

Bien-sûr, tout ce que nous allons faire ensuite, fonctionne uniquement localement sur votre ordinateur.

Pour rendre la situation plus crédible, nous allons ajouter un nom DNS à notre site Web. Il existe une manière simple et rapide de faire des modifications DNS sans installer votre propre serveur DNS sur un LAN, et sans vous connecter à l'interface d'administration de votre routeur : le fichier HOSTS !

- **Sous Windows, rendez-vous dans "C:\Windows\System32\drivers\etc" et ouvrez le fichier « hosts » avec le droit Administrateur. Ajoutez en fin de fichier la ligne :**

```
<IP_Debian> comptes.com
```

Le fichier HOSTS est toujours le premier fichier lu par votre PC lors de la résolution DNS si une entrée y est renseignée, elle est prioritaire sur toutes les autres.

- **Modifiez votre fichier de configuration « gest_util-TLS.conf » pour mettre « comptes.com » en ServerAlias dans le *:80 et *:443. Reload le service Apache2 ensuite.**



Ctrl + O permet d'enregistrer votre travail sous Nano. Ctrl + X permet de quitter Nano. La commande **apache2ctl configtest** permet de vérifier la syntaxe correcte de votre fichier de configuration.

Cela signifie que votre site est désormais accessible via l'IP, mais aussi avec le nom FQDN « comptes.com ».

- **Testez et troublshootez si besoin.**

Régénération des certificats

- **Retournez sur votre serveur Debian, et rendez-vous dans votre dossier certificats customs, normalement /etc/ssl/custom :**

```
cd /etc/ssl/custom
```

- **Supprimez les 2 anciens certificats :**

```
rm -f *
```

Lors de la première génération des certificats, rappelez-vous, nous avons passé tous les paramètres dans la ligne de commande. Si nous voulons faire passer notre certificat pour un certificat de confiance, nous allons passer par un fichier de configuration intermédiaire, car il nous faut

renseigner plus d'informations dans notre certificat, notamment les « *autres nom de l'objet* » (« *alt_names* »), nécessaires au bon fonctionnement de notre certificat vérifié.

- Créez et complétez le fichier « confcomptes.cnf » en ajoutant les lignes suivantes :

```
nano confcomptes.cnf
```

```
[req]
distinguished_name = dn
default_bits = 4096
prompt = no
default_md = sha256
req_extensions = v3_req
x509_extensions = v3_ca

[dn]
emailAddress = <votre_adresse_courriel>
CN = comptes.com
O = OPEN IT
OU = CYBERSECURITE
L = MONTPELLIER
ST = HERAULT
C = FR

[v3_req]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, keyEncipherment
subjectAltName = @alt_names

[v3_ca]
subjectAltName = @alt_names

[alt_names]
DNS.1 = comptes.com
IP.1 = <IP_Debian>
```

Essayez de comprendre ce fichier de configuration. Vous devrez l'expliquer avec votre intervenant. Renseignez *<les attributs obligatoires>*. Vous pouvez modifier les attributs [dn] uniquement si vous souhaitez personnaliser votre certificat.

- Envoyez la commande de génération des certificats depuis un fichier de configuration :

```
openssl req -x509 -nodes -days 365 -out comptes.crt -keyout comptes.key -config
confcomptes.cnf
```

Vous remarquerez par conséquent que la commande est donc beaucoup plus courte et que vous n'avez plus d'interaction utilisateur lors de la génération : « *prompt = no* ».

Comme nous travaillons dans le dossier */etc/ssl/custom* ils sont déjà positionnés au bon endroit.

- Rappelez-vous de la [configuration Apache2](#) que vous avez effectuée plus haut et modifiez les chemins vers le certificat et la clé par ceux que nous venons de créer.



Ctrl + O permet d'enregistrer votre travail sous Nano. Ctrl + X permet de quitter Nano. La commande **apache2ctl configtest** permet de vérifier la syntaxe correcte de votre fichier de configuration.

- Positionnez seulement les droits de lecture sur le fichier .crt :

```
chmod 440 /etc/ssl/custom/comptes.crt
```

- Effectuez un reload d'Apache2. Rendez-vous sur votre site Web, et vérifiez que ce nouveau certificat soit pris en compte :

```
systemctl reload apache2
```

Qualifier ce certificat comme certificat de confiance (Google Chrome)

À ce stade, vous avez toujours le message d'alerte. C'est parfaitement normal.

Exportez le certificat depuis Google Chrome :

1. Affichez le certificat ;
2. Allez dans « Détails » ;
3. Cliquez sur « Exporter » en bas à droite ;
4. Enregistrez le fichier avec le format par défaut (.pem ; .crt).

Mettre ce certificat dans le magasin de certificat de confiance Windows :

1. Allez dans les paramètres de Google Chrome ;
2. « Confidentialité et sécurité » ;
3. « Sécurité » ;
4. « Gérer les certificats » ;
5. Onglet « Autorité de certification racines de confiance » ;
6. « Importer » ;
7. Sélectionnez votre certificat exporté lors de l'étape précédente (affichez tous les fichiers) ;

8. Redémarrez Google Chrome.

- Rendez-vous sur votre site Web <https://comptes.com>. Vous n'avez plus de message d'erreur. Si vous retournez voir le certificat, celui-ci est désormais considéré comme valide.

Vous venez de mettre votre certificat auto-signé dans les certificats racines de confiance de votre ordinateur. Par conséquent, votre ordinateur le considère désormais comme un certificat légitime.

	Google Chrome et Microsoft Edge utilisent tous deux le magasin de certificat Windows. Mozilla Firefox, quant à lui, possède son propre magasin de certificats racines de confiance. Les étapes sont alors différentes.
---	--

6) Lecture des flux chiffrés

- En reprenant les étapes effectuées lors de la [lecture des flux en clair](#), prouvez, avec TCPDUMP et Wireshark, que les informations sont maintenant chiffrées lors du transit. (Application Data).

	Jalonnement : Avant de passer à la partie suivante,appelez votre intervenant pour un contrôle intermédiaire de votre travail réalisé jusqu'à présent. Assurez-vous que tout soit fonctionnel avant son arrivée.
---	--

- Prenez quelques minutes pour nettoyer les fichiers installés pour ce TP. Commentez ou supprimez la ligne dans le fichier « hosts » concernant l'entrée DNS « comptes.com ». Puis révoquez également le certificat que vous avez ajouté dans votre magasin de certificats Windows. EN REVANCHE, NE SUPPRIMEZ PAS LES VMs !

3. Clé SSH

- Démarrez votre première VM, celle qui n'a pas le chiffrement TLS.
➤ Identifiez bien les adresses IP de vos 2 machines.

Voici les miennes (pour les exemples) :

- AB VM Web : **192.168.1.67**
- AB VM Web-TLS : **192.168.1.68**

La VM cliente (depuis laquelle on se connecte) sera la VM Web-TLS, la VM serveur (sur laquelle on se connecte) sera la VM Web. Donc, source : 192.168.1.68 et destination : 192.168.1.67.

1) Connexion SSH Standard

Lors de cette partie, vous avez oublié un document sur l'ancien serveur Web non TLS. Cet ancien serveur est dans le cloud et votre seul moyen pour y accéder et de vous y connecter en SSH depuis la VM Web-TLS.

- Établir une connexion SSH standard avec la VM Web :

```
ssh <utilisateur>@<IP_Debian_Web>
```

Par sécurité, lors de votre première connexion SSH, la Fingerprint du serveur distant vous est présenté.

- Rendez-vous sur la VM serveur distante (via la fenêtre de VirtualBox) et vérifiez la Fingerprint :

```
ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub
```

Si la Fingerprint du serveur distant correspond bien à la Fingerprint affiché lors de votre première connexion SSH, alors vous vous connectez bien à la bonne machine. Vous pouvez accepter la connexion.

- Refaites une connexion SSH, renseigner le mot de passe du serveur distant et vous êtes connectés.
- Fermez la connexion ssh avec **exit**.

Si vous établissez une nouvelle connexion, vous verrez que vous avez toujours besoin du mot de passe.

Nous allons utiliser une autre solution : la connexion par clé SSH.

2) Connexion par clé SSH

Génération de la clé

- Générez une clé SSH de 4096 bits depuis votre client, à l'emplacement par défaut :

```
ssh-keygen -b 4096
```



Il vous est ensuite proposé de renseigner une **passphrase** pour sécuriser la clé. Je veux vous montrer une connexion sans mot de passe. Mettre une passphrase à notre clé reviendrait alors strictement au même que d'utiliser un mot de passe. **Dans ce cas, nous n'allons pas mettre de passphrase.** En production, il est recommandé d'en mettre une.

Laissez donc l'option « passphrase » vide.

Vos fichiers se sont créés ici :

```
ls -la /root/.ssh
```



Sous Linux, un dossier précédé d'un point est un dossier caché. La philosophie est la même que sous Windows. Pour afficher ces types de dossier, la commande `ls -l` ne suffit plus, il faut utiliser `ls -la`.



Partie cours

Nous nous trouvons toujours sur un client Linux, où vous remarquerez l'existence d'un fichier supplémentaire nommé « known_hosts ». Ce fichier joue un rôle crucial dans l'identification des serveurs lors des connexions SSH. Chaque fois que vous vous connectez à un nouveau serveur depuis votre compte utilisateur, une nouvelle entrée est ajoutée à ce fichier. Cela signifie que lors de la première connexion à un serveur, vous serez invité à valider sa clé, mais cette étape ne sera pas nécessaire pour les connexions ultérieures.

Dépose de la clé sur le serveur

Pour déposer notre clé sur le serveur distant, nous allons utiliser la commande « ssh-copy-id » avec, pour la dernière fois, le couple utilisateur et mot de passe.

- Envoyez la clé sur le serveur distant :

```
ssh-copy-id <utilisateur>@<IP_Debian_Web>
```

Normalement, vous devriez avoir un rendu comme celui-ci :

```
root@tp-ps-svg-TLS:~# ssh-copy-id root@192.168.1.67
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
root@192.168.1.67's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'root@192.168.1.67'"
and check to make sure that only the key(s) you wanted were added.
```

- Établissez une nouvelle connexion :

```
ssh <utilisateur>@<IP_Debian_Web>
```

Normalement, plus aucun mot de passe vous est demandé, car la clé générée et envoyée effectue son travail d'authentification.

- Sur le serveur distant, vérifiez le stockage de la clé dans le répertoire de l'hôte :

```
cat /root/.ssh/authorized_keys
```

Vous verrez en bout de ligne de la clé que cette clé n'est valable que pour un utilisateur précis, et depuis un client précis :

```
root@tp-ps-svg:~/._ssh# cat /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCYoXH7zkrmWdkLXhzFth348LdcumvbKWrz368Xcf0wWKUfg/qnLMwqa0pqPR5WdvtapfkrKH2AnlHn2bjU6ook0R0
wQaRYKxUm73mPz1KfdarYW+ipg02+v0rEdSc/lHr/UBDcfs0bSHwCzRmpgeySwc11LUfJ0Af4KUA340u9YMeNtBTbfKTxZl702qTuE4/qkRwhvHwhpwij0/C6MqmM
xvfWI+qUZd6hZ/j i0lddoBSpbKV16xImS3y4W3nU/9JUJPTxcE0l2F9VmlKsRl/I1ZmbF9FftiaC9U073MnQa4EZyIrxvvYg0Wjd4h0PXLuM5EKepgAfVBvuTy8bzI
HCo3Rjw7aeg9UcUB20bXhdRNCKpnVm3RvKb5Pvy7gh4v7xY+37BU0Jan9DDQJ7dgQMnIiwiwXhnCrtCJ4Ss+WylmxPEqV62gCsU3rw== root@tp-ps-svg-TLS
```

- Dans les paramètres SSH du serveur distant, modifiez la ligne « **PermitRootLogin yes** » par « **PermitRootLogin prohibit-password** »

```
nano /etc/ssh/sshd_config
```

- Redémarrez SSH :

```
systemctl restart ssh
```

- Déconnectez-vous du serveur distant :

```
exit
```

- Établissez une nouvelle connexion via la clé. Vérifiez le bon fonctionnement.
- Commentez la ligne qui correspond à votre clé dans le fichier :

- Mettre un « # » pour commenter une ligne ;
- **Ctrl+o** pour enregistrer, **Ctrl+x** pour quitter.

```
nano /root/.ssh/authorized_keys
```

- **Déconnectez-vous du serveur distant :**

```
exit
```

- **Établissez une nouvelle connexion via mot de passe.**

Question 1 : Que constatez-vous ? Justifiez votre réponse.



Jalonnement : Appelez votre intervenant pour un contrôle de votre travail réalisé jusqu'à présent. Assurez-vous que tout soit fonctionnel avant son arrivée.