

Sommaire

- 1) *Introduction*
- 2) *Créer un certificat SSL, sur Apache pour Ubuntu server 14.04*
- 3) *Conclusion*
- 4) *Sécuriser son serveur web*
- 5) *Tester la sûreté de son serveur*

1) Introduction :

SSL/TLS est un protocole ayant pour but de créer un canal de communication authentifié, protégé en confidentialité et en intégrité. L'objectif initial de SSL/TLS était la sécurisation du protocole HTTP, mais son champ d'application s'est élargi depuis : protection d'autres services comme SMTP ou LDAP, création de réseaux privés virtuels (VPN), sécurisation de réseaux sans-fil (EAP-TLS).

Après un rapide historique des différentes versions de SSL/TLS et une description du protocole, cet article présente un panorama des attaques connues, qu'elles portent sur le protocole en lui-même, les algorithmes cryptographiques, ou les certificats mis en œuvre.

Cette énumération des vulnérabilités de SSL/TLS permettra de déduire des recommandations pour une utilisation sécurisée de SSL/TLS. Chaque recommandation sera validée par une expérimentation visant à déterminer la faisabilité de leur application.

SSL (Secure Sockets Layer) et TLS (Transport Layer Security) sont deux variantes d'un même protocole.

Leur objectif est de fournir un certain nombre de services pour sécuriser un canal de communication :

- Authentification unilatérale ou mutuelle :
- Confidentialité des données échangées de bout en bout :
- intégrité des données de bout en bout.

Cette couche de sécurité peut être appliquée à tout type de canal de communication entre deux parties garantissant la transmission des données de façon ordonnée. En pratique, SSL/TLS est surtout utilisé sur la couche transport TCP, afin de proposer des versions sécurisées de protocoles existants (par exemple : HTTPS = HTTP + SSL)

Signatures de certificats

On distingue différents types de certificats selon le niveau de signature :

Les certificats auto-signés sont des certificats à usage interne. Signés par un serveur local, ce type de certificat permet de garantir la confidentialité des échanges au sein d'une organisation, par exemple pour le besoin d'un intranet. Il est ainsi possible d'effectuer une authentification des utilisateurs grâce à des certificats auto-signés.

Les certificats signés par un organisme de certification sont nécessaires lorsqu'il s'agit d'assurer la sécurité des échanges avec des utilisateurs anonymes, par exemple dans le cas d'un site web sécurisé accessible au grand public. Le certificateur tiers permet d'assurer à l'utilisateur que le certificat appartient bien à l'organisation à laquelle il est déclaré appartenir.

Types d'usages

Les certificats servent principalement dans trois types de contextes :

Le certificat client, stocké sur le poste de travail de l'utilisateur ou embarqué dans un conteneur tel qu'une carte à puce, permet d'identifier un utilisateur et de lui associer des droits. Dans la plupart des scénarios il est transmis au serveur lors d'une connexion, qui affecte des droits en fonction de l'accréditation de l'utilisateur. Il s'agit d'une véritable carte d'identité numérique utilisant une paire de clé asymétrique d'une longueur de 512 à 1024 bits

Le certificat serveur installé sur un serveur web permet d'assurer le lien entre le service et le propriétaire du service. Dans le cas d'un site web, il permet de garantir que l'URL et en particulier le domaine de la page web appartiennent bien à telle ou telle entreprise. Par ailleurs il permet de sécuriser les transactions avec les utilisateurs grâce au protocole SSL.

Le certificat VPN est un type de certificat installé dans les équipement réseaux, permettant de chiffrer les flux de communication de bout en bout entre deux points (par exemple deux sites d'une entreprise). Dans ce type de scénario, les utilisateurs possèdent un certificat client, les serveurs mettent en oeuvre un certificat serveur et les équipements de communication utilisent un certificat particulier (généralement un certificat IPSec).

Contexte :

Cette situation professionnelle est simulée dans le contexte d'une société m'ayant confier comme tâche de créer un serveur WEB complet, sécuriser et redondant. La société souhaite ainsi protéger son serveur WEB de façon efficaces afin d'être en mesure de se prémunir contre des attaques internes ou externes

2) Créer un certificat SSL sur Apache pour Ubuntu 14.04

Pour commencer, il faut installer apache avec la commande suivante :

```
sudo apt-get update
```

```
sudo apt-get install apache2
```

Activez le module SSL

```
sudo a2enmod.ssl
```

Ensuite après l'activation du protocole SSL , il faut redémarrer le serveur WEB pour que le changement soit reconnu :

```
sudo service apache2 restart
```

Le serveur est maintenant capable de gérer SSL.

Créer un certificat SSL auto-signé

Il faudra commencer par créer un sous-répertoire dans la hiérarchie de configuration d'Apache pour placer les fichiers de certificats que nous allons faire :

```
sudo mkdir /etc/apache2/ssl
```

Ensuite il y a un emplacement pour placer notre clé et le certificat, nous pouvons créer ainsi les deux en une seule étape grâce à la commande :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048  
-keyout/etc/apache2/ssl/apache -out/etc/apache2/ssl/apache.crt
```

Dictionnaire :

-OpenSSL : Ceci est l'outil de ligne de commande de base fournie par OpenSSL pour créer et gérer des certificats, des clés, des demandes de signatures, etc.

-req : Ceci indique le certificat X.509 (CSR) de gestion. X.509 est une norme de l'infrastructure à clé publique qui adhère à SSL pour sa clé et le certificat management. Puisque nous souhaitons créer un nouveau certificat X.509, voilà ce que nous voulons.

-x509 : Cette option spécifie que nous voulons faire un fichier de certificat auto-signé au lieu de générer une demande de certificat.

-nodes : Cette option indique à OpenSSL que nous ne voulons pas sécuriser notre fichier clé avec un mot de passe. Avoir un fichier de clé protégée par mot de passe serait d'obtenir automatiquement le démarrage d'Apache et nous devrions entrer le mot de passe à chaque fois que le service redémarre.

-days 365 : Ceci indique que le certificat que nous créons sera valide pour un an.

-rsa -newkey:2048 : Cette option permet de créer la demande de certificat et une nouvelle clé privée dans le même temps. Cela est nécessaire car nous ne créons une clé privée à l'avance. Le [RSA : 2048](#) raconte OpenSSL pour générer une clé RSA qui est de 2048 bits.

-keyout : Le nom de paramètres du fichier de sortie qui a été créé pour le fichier de clé privée est en cours de création.

-out : Le nom de cette option du fichier de sortie pour le certificat, a été généré. Lorsque vous appuyez sur la touche « Entrée », il vous sera demandé un certain nombre de questions.

L'élément le plus important qui est demandé est la ligne qui dit « Common Name (nom de domaine complet du serveur ou votre nom) ». Il faut entrer le nom de domaine que vous souhaitez

Associer avec le certificat, ou l'adresse IP publique du serveur si vous ne disposez pas d'un nom de domaine.

```
Paramétrage de openssl (1.0.1f-1ubuntu2.19) ...
root@ubuntu:/home/ubuntu# cd /etc/ssl
root@ubuntu:/etc/ssl# cd /etc/ssl
root@ubuntu:/etc/ssl# sudo openssl genrsa -out server.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
root@ubuntu:/etc/ssl# sudo openssl req -new -key server.key -out server.csr
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]:bts
string is too long, it needs to be less than 2 bytes long
Country Name (2 letter code) [AU]:ft
State or Province Name (full name) [Some-State]:bts
Locality Name (eg, city) []:Réunion
Organization Name (eg, company) [Internet Widgits Pty Ltd]:btssio
Organizational Unit Name (eg, section) []:bts
Common Name (e.g. server FQDN or YOUR name) []:sio
Email Address []:fabrice.tamil@gmail.com_
```

La clé et le certificat seront créés et placés dans le répertoire / etc/apache2 /ssl.

-Configurer Apache pour utiliser SSL

Configurer le fichier default-ssl.conf fichier qui contient une configuration SSL par défaut.

Ouvrez le fichier avec les privilèges root :

```
GNU nano 2.2.6 Fichier : ...apache2/sites-available/default-ssl.conf Modifié
<IfModule mod_ssl.c>
    <VirtualHost 192.168.0.66:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For exam$
        # following line enables the CGI configuration for this host on$
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

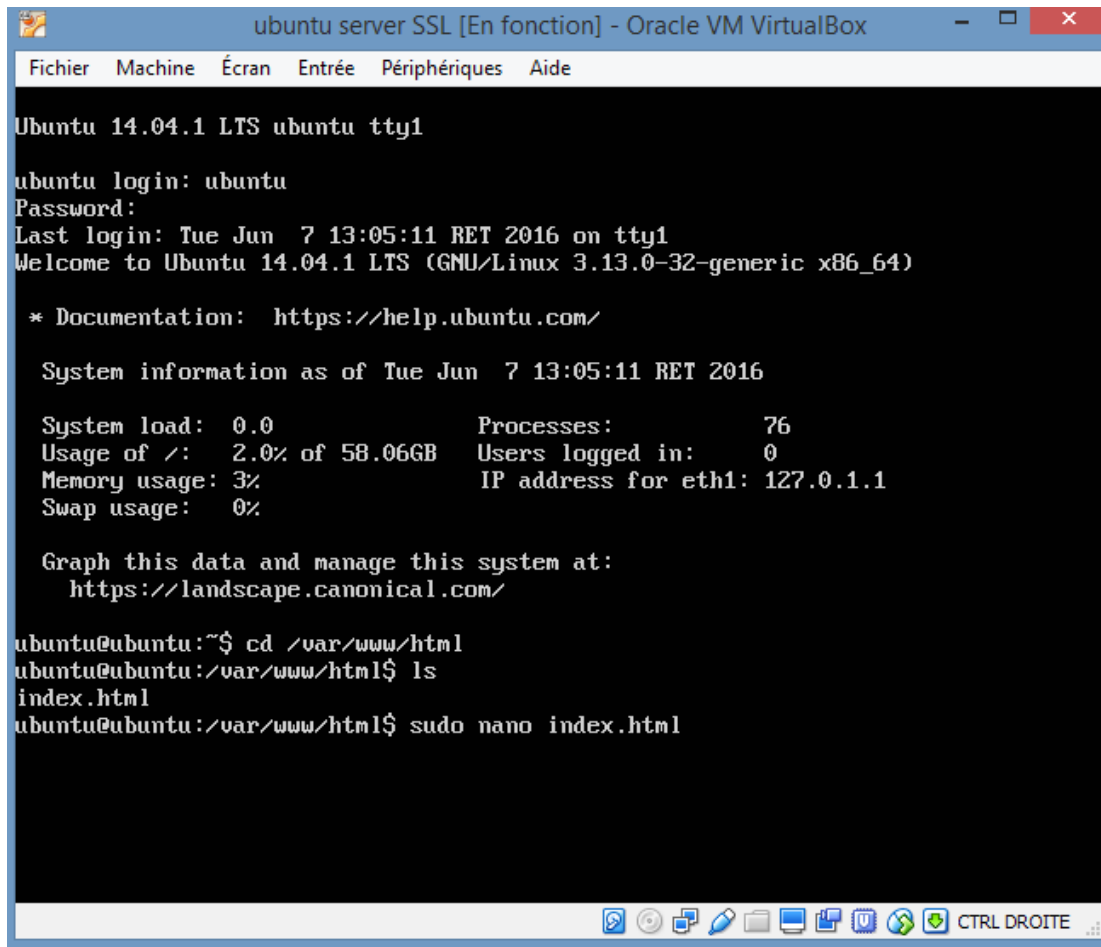
        # SSL Engine Switch:
        # Enable/Disable SSL for this virtual host.
        SSLEngine on

^G Aide      ^O Écrire   ^R Lire fich.^Y Page préc.^K Couper     ^C Pos. cur.
^X Quitter   ^J Justifier^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

`sudo nano /etc/apache2/sites-available/default-ssl.conf`

Le fichier doit être édité de cette manière :

Modifier le nom du serveur :



```
ubuntu server SSL [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Entrée  Périphériques  Aide

Ubuntu 14.04.1 LTS ubuntu tty1

ubuntu login: ubuntu
Password:
Last login: Tue Jun  7 13:05:11 RET 2016 on tty1
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Tue Jun  7 13:05:11 RET 2016

System load:  0.0                Processes:            76
Usage of /:   2.0% of 58.06GB     Users logged in:     0
Memory usage: 3%                IP address for eth1: 127.0.1.1
Swap usage:   0%

Graph this data and manage this system at:
https://landscape.canonical.com/

ubuntu@ubuntu:~$ cd /var/www/html
ubuntu@ubuntu:/var/www/html$ ls
index.html
ubuntu@ubuntu:/var/www/html$ sudo nano index.html
```

ubuntu server SSL [En fonction] - Oracle VM VirtualBox

FichierMachineÉcranEntréePériphériquesAide

GNU nano 2.2.6Fichier : index.html

```
<div class="section_header section_header_red">
  <div id="about"></div>
  Serveur1
</div>
<div class="content_section_text">
  <p>
    This is the default welcome page used to test the correct
    operation of the Apache2 server after installation on Ubuntu sy$
    It is based on the equivalent page on Debian, from which the Ub$
    packaging is derived.
    If you can read this page, it means that the Apache HTTP server$
    this site is working properly. You should <b>replace this file<$
    <tt>/var/www/html/index.html</tt>) before continuing to operate$
  </p>

  <p>
    If you are a normal user of this web site and don't know what t$
    about, this probably means that the site is currently unavailab$
    maintenance.
    If the problem persists, please contact the site's administrato$
  </p>
</div>
<div class="section_header">
```

^G Aide

^X Quitter

^O Écrire

^J Justifier

^R Lire fich.

^W Chercher

^Y Page préc.

^V Page suiv.

^K Couper

^U Coller

^C Pos. cur.

^T Orthograp.

CTRL DROITE

Activez l'hôte virtuel SSL

Maintenant que nous avons configuré notre hôte virtuel SSL.

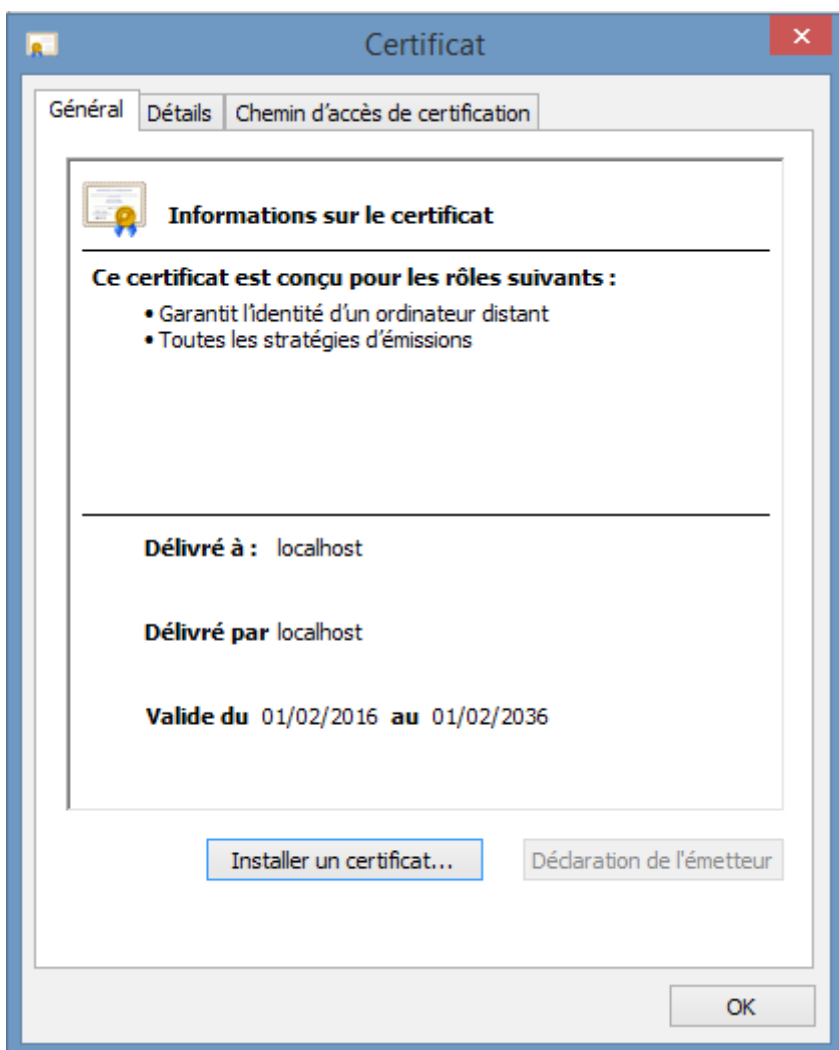
L'activé en tapant :

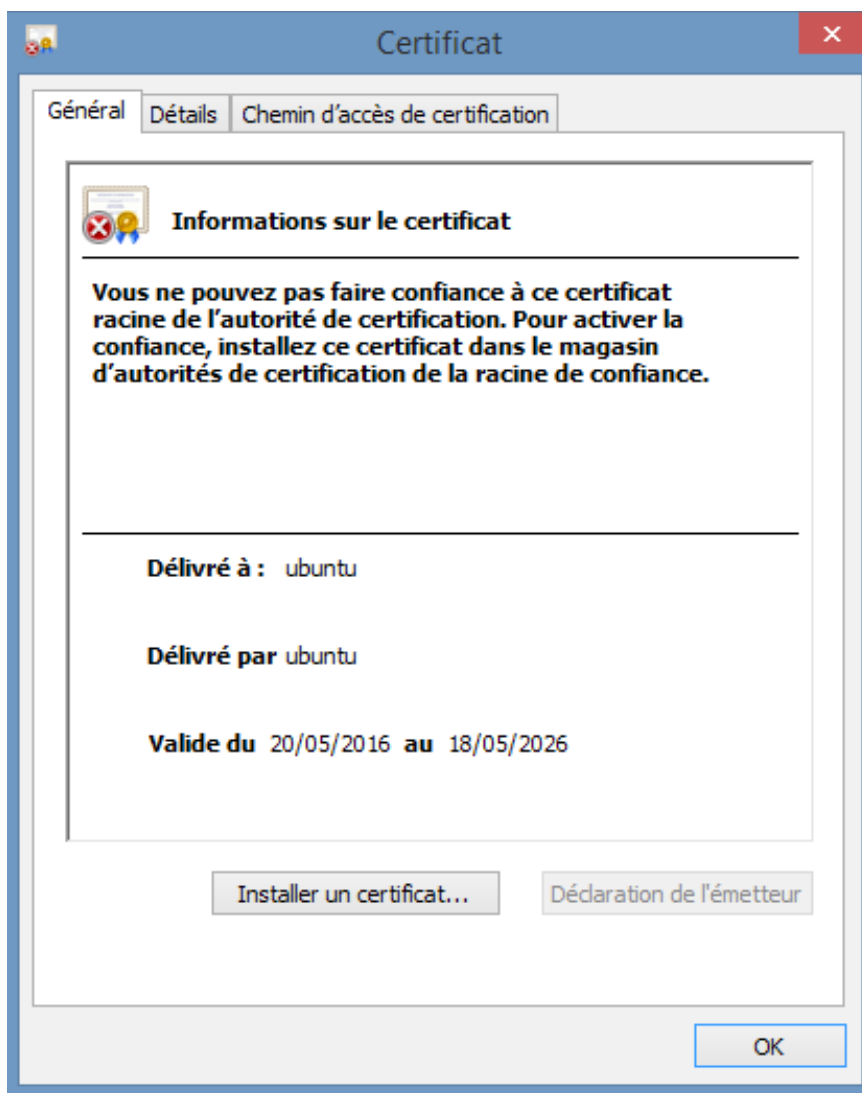
```
sudo a2ensite default-ssl.conf
```

Nous avons ensuite besoin de redémarrer Apache pour charger notre nouveau fichier d'hôte virtuel :

```
sudo service apache restart
```

Intégration des certificat :







Assistant Importation du certificat



Bienvenue dans l'Assistant Importation du certificat

Cet Assistant vous aide à copier des certificats, des listes de certificats de confiance et des listes de révocation des certificats d'un disque vers un magasin de certificats.

Un certificat, émis par une autorité de certification, confirme votre identité et contient des informations permettant de protéger des données ou d'établir des connexions réseau sécurisées. Le magasin de certificats est la zone système où les certificats sont conservés.

Emplacement de stockage


☒ Utilisateur actuel



☐ Ordinateur local

Pour continuer, cliquez sur Suivant.

Suivant

Annuler



 Assistant Importation du certificat

Magasin de certificats

Les magasins de certificats sont des zones système où les certificats sont conservés.

Windows peut sélectionner automatiquement un magasin de certificats, ou vous pouvez spécifier un emplacement pour le certificat.

☒ Sélectionner automatiquement le magasin de certificats en fonction du type de certificat

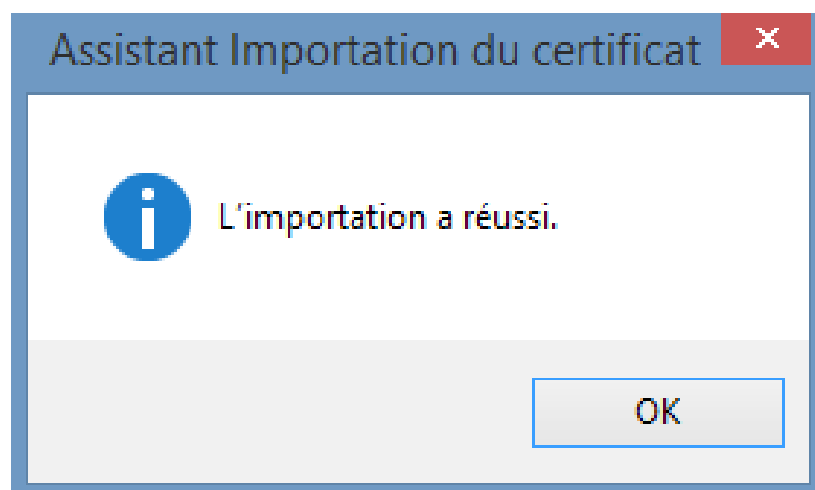
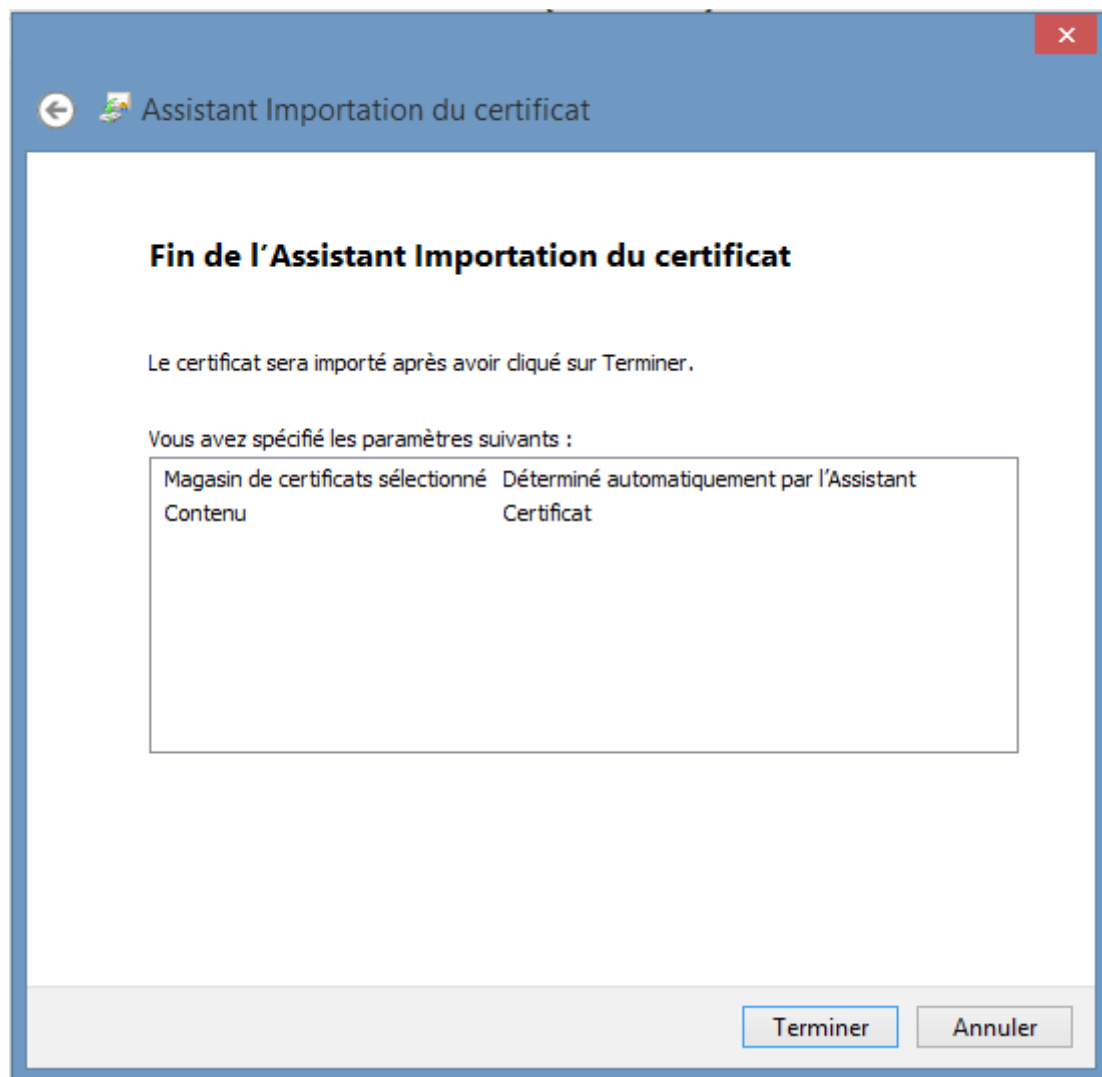
☐ Placer tous les certificats dans le magasin suivant

Magasin de certificats :

Parcourir...

Suivant

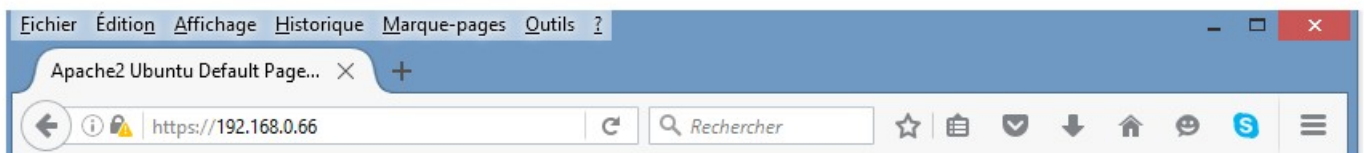
Annuler




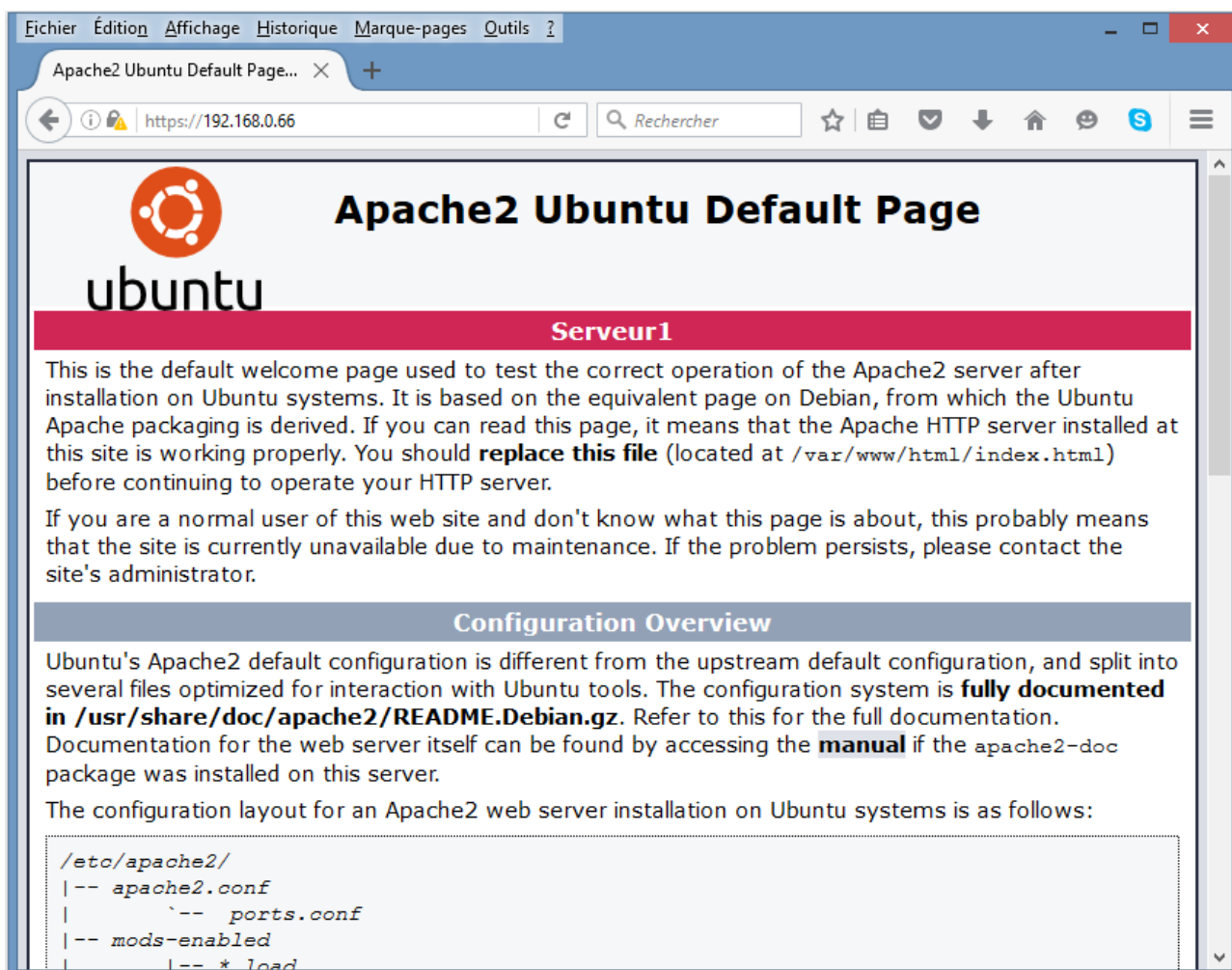
Testez la configuration

Tester la configuration en tapant le nom de domaine ou adresse IP publique,l'adresse du serveur après avoir spécifié les [https://protocole](https://) ,comme ceci :

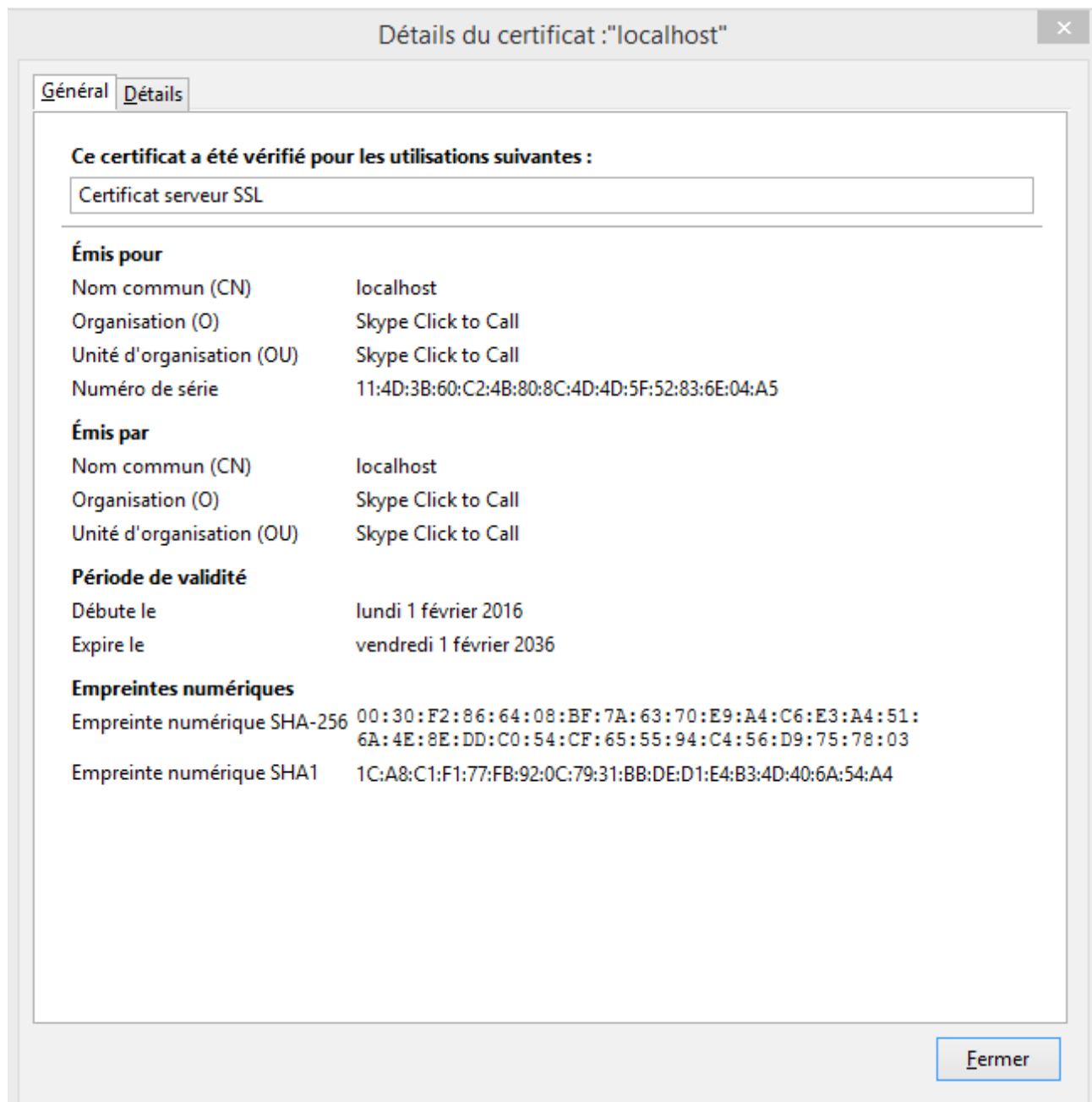
https://server_domain_name_or_IP



-  : un cadenas gris avec un triangle orange indique que Firefox ne bloque pas le contenu passif non sécurisé. Les pirates peuvent être capables de manipuler des parties de la page, par exemple en affichant du contenu falsifié ou inapproprié, mais ils ne devraient pas pouvoir voler vos données personnelles à partir du site.



Détail de certificats :



3) Conclusion

On doit voir activé SSL sur notre site web. Cela aidera à sécuriser la communication entre les visiteurs et notre site, mais chaque utilisateur ne peut pas vérifier la validité du certificat.

Structure d'un certificat

Les certificats sont des petits fichiers divisés en deux parties :

- La partie contenant les informations
 - La partie contenant la signature de l'autorité de certification
- La structure des certificats est normalisée par le standard X.509 de l'UIT (plus exactement X.509v3), qui définit les informations contenues dans le certificat :
- La version de X.509 à laquelle le certificat correspond ;
 - Le numéro de série du certificat ;
 - L'algorithme de chiffrement utilisé pour signer le certificat ;
 - Le nom (DN, pour Distinguished Name) de l'autorité de certification émettrice ;
 - La date de début de validité du certificat ;
 - La date de fin de validité du certificat ;
 - L'objet de l'utilisation de la clé publique ;
 - La clé publique du propriétaire du certificat ;
 - La signature de l'émetteur du certificat (thumbprint).
- L'ensemble de ces informations (informations + clé publique du demandeur) est signé par l'autorité de certification, cela signifie qu'une fonction de hachage crée une empreinte de ces informations, puis ce condensé est chiffré à l'aide de la clé privée de l'autorité de certification; la clé publique ayant été préalablement largement diffusée afin de permettre aux utilisateurs de vérifier la signature avec la clé publique de l'autorité de certification.

Si il s'agit d'un site public avec la nécessité d'un certificat SSL, il est conseillé d'acheter un certificat SSL auprès d'une autorité de certification de confiance.

Les Autorités de Certification (AC) émettent des certificats numériques. Les certificats numériques sont des petits fichiers de données vérifiables qui contiennent des informations d'identification permettant aux sites web, aux personnes et aux appareils de prouver l'authenticité de leur identité en ligne (authenticité garantie par la vérification de leur identité par une AC). Les AC jouent un rôle essentiel au niveau du fonctionnement d'Internet et de la transparence des transactions effectuées en ligne. Elles émettent des millions de certificats numériques chaque année et ces certificats peuvent être utilisés pour protéger des informations, chiffrer des milliards de transactions et sécuriser les communications.

4) Sécurisation du serveur Web

Sécurisation et initiation aux thématiques de la sécurité informatique.

Renforcement de sécurité d'un serveur qui peut être vulnérable face aux attaques les plus simplistes.

Filtrage du trafic vers le firewall

Le firewall (pare-feu) est l'élément indispensable pour sécuriser un serveur. Il filtre le trafic en n'autorisant que les échanges permis par l'administrateur. Sans firewall correctement configuré, tous les trafics sont plus ou moins permis, donc accessible et vulnérable, quelqu'un ayant de mauvaise intention peut interagir et attaquer les éléments qu'il souhaite. Cela peut être détectable grâce à un scan de ports.

Or, le noyau Linux offre déjà un pare-feu à l'utilisateur, qu'il est possible de configurer via le logiciel [iptables](#) (normalement contenu dans [/sbin/iptables](#)).

Si il n'est pas installé :

[apt-get install iptables](#)

Déclaration des règles

Création du script :

[nano /etc/init.d/firewall](#)

On écrit : # !/bin/sh

Mon pare-feu :

```
GNU nano 2.2.6      Fichier : /etc/init.d/firewall      Modifié

#!/bin/sh

#Réinitialise les règles
sudo iptables -t filter -F
sudo iptables -t filter -X

#Bloque tout le trafic
sudo iptables -t filter -P INPUT DROP
sudo iptables -t filter -P FORWARD DROP
sudo iptables -t filter -P OUTPUT DROP

#Autorise les connexions déjà établies et localhost
sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -t filter -A INPUT -i lo -j ACCEPT
sudo iptables -t filter -A OUTPUT -o lo -j ACCEPT

# HTTP
sudo iptables -t filter -A OUTPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -t filter -A INPUT -p tcp --dport 80 -j ACCEPT

# HTTPS
sudo iptables -t filter -A OUTPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -t filter -A INPUT -p tcp --dport 443 -j ACCEPT

^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper      ^C Pos. cur.
^X Quitter   ^J Justifier^W Chercher  ^U Page suiv.^U Coller     ^T Orthograp.
```

```
#SSH
```

```
sudo iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -t filter -A INPUT -p tcp --dport 443 -j ACCEPT
```

```
# FTP
```

```
sudo iptables -t filter -A OUTPUT -p tcp --dport 20:21 -j ACCEPT
sudo iptables -t filter -A INPUT -p tcp --dport 20:21 -j ACCEPT
```

```
#DNS
```

```
sudo iptables -t filter -A OUTPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p udp --dport 53 -j ACCEPT
sudo iptables -t filter -A INPUT -p tcp --dport 53 -j ACCEPT
sudo iptables -t filter -A INPUT -p udp --dport 53 -j ACCEPT
```

```
# Mail SMTP
```

```
sudo iptables -t filter -A INPUT -p tcp --dport 25 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 25 -j ACCEPT
```

```
# Mail POP3
```

```
sudo iptables -t filter -A INPUT -p tcp --dport 110 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 110 -j ACCEPT
```

```
#Mail IMAP
```

```
sudo iptables -t filter -A INPUT -p tcp --dport 143 -j ACCEPT
sudo iptables -t filter -A OUTPUT -p tcp --dport 143 -j ACCEPT
```

```
^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper      ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher   ^U Page suiv.^U Coller     ^T Orthograp.
```

```
# anti flood
```

```
sudo iptables -A FORWARD -p tcp --syn -m limit --limit 1/second -j ACCEPT
sudo iptables -A FORWARD -p udp -m limit --limit 1/second -j ACCEPT
sudo iptables -A FORWARD -p icmp --icmp type echo-request -m limit --limit 1/se$
```

Démarrage du pare-feu

Enfin, faudra lancer le pare-feu

```
chmod +x /etc/init.d/firewall  
/etc/init.d/firewall
```

Il est important de charger ce script au démarrage de la machine afin qu'un simple reboot ne vous laisse pas sans protection :

```
update-rc.d firewall defaults
```

Portsentry (Scan de ports)

Cet utilitaire permet de bloquer en temps réel la plupart des scans de port connus (même étant très discrets et échappant aux règles de filtrage du firewall basiques) . Scanner les ports signifie de tester tous les ports d'une machine afin de déterminer ceux qui sont ouverts et qui risque d'être attaquer (les portes d'entrées)

Portsentry : Interrompt l'attaquant dans ses manipulation

```
apt-get install portsentry
```

Configuration :

```
nano /usr/local/psionic/portsentry/portsentry.conf
```

Ou :

```
nano/etc/portsentry/portsentry.conf
```

Commentez les lignes `KILL_HOSTS_DENY`

Dé commentez la ligne `KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"`.

Ainsi, Portsentry ajoutera une règle dans le firewall (iptables) pour rejeter les paquets en cas de scans.

On démarre le logiciel (il faut le lancer deux fois, pour TCP et UDP) :

```
portsentry -audp  
portsentry -atcp
```

Rkhunter (rootkit et backdoors)

Dernier volet de cette section intrusion, les backdoors. Si un attaquant (hacker) arrive à prendre possession de la machine, c'est dans l'intention de laisser une backdoor (porte dérobée) qui lui donnera accès à reprendre le contrôle plus tard, ainsi qu'un rootkit pour la dissimuler : l'attaquant maintient ainsi un accès frauduleux à la machine.

Rkhunter est un utilitaire qui est chargé de détecter d'éventuels rootkits sur le serveur. Il est relativement léger (s'exécute une fois par jour par défaut).

Pour l'installer :

```
apt-get install rkhunter
```

Il est conseillé de modifier un peu la configuration :

```
nano /etc/default/rkhunter
```

-REPORT_EMAIL : indiquez un mail pour recevoir des alertes de Rkhunter ;
-CRON_DAILY_RUN : mettez « yes » pour une vérification quotidienne de votre machine vers un cron.

Rkhunter pas fiable à 100 %

5) Test de la sûreté du serveur

Outils d'analyses de défaillance :

Scanner de port

nmap est le meilleur outil de scan de ports : il détermine l'ouverture de connexions sur un grand nombre de ports de la machine afin de détecter si ils sont ouverts ou fermés.

```
apt-get install nmap
```

Comme notre serveur est important il est mieux de faire un scan complet et approfondi afin de détecter toute anomalie suspecte

```
nmap -v ip_ou_nom_de_la_machine
```

On aura donc la liste des ports ouverts.

On pourra ainsi tester un port en particulier avec l'argument -p port. Il n'est pas recommandé d'utiliser nmap sur quiconque autre que son propre serveur.