TD : Serveur http et Ssl + Cloud (Iaas)

Objectifs :

- Création d'un site web de test on-premise et dans le cloud
- Voir la différence de sécurité avant et après l'installation de Ssl et de l'utilisation de certificats autosignés.
- Étude des trames avec Wireshark

Architecture :

- Utiliser une machine virtuelle Linux debian et un client Windows 10.
- Changer le nom de l'ordinateur
- Attention : Résolution de noms avec fichier hosts
- Penser à modifier le fichier hostname pour lui mettre un nom
- Installer les mises à jour : apt-get install –fix-missing
- Installation d'apache2 : paquetage s'appelle apache 2
- Installer php (dernière version)

Mise en œuvre

1. **Création d'un site web en local (LAN)**

- Modifier ou créer la page index.html dans /var/www/html
- Vous utiliserez winscp ou putty pour créer votre page sur votre machine Windows et la copier sur la machine linux. Cet accès se fera en ssh.

```
<p>
   Cette page ne contient que du HTML.<br />
   Veuillez taper votre nom :
</p>  <form action="cible.php" method="post">
<p>
   <input type="text" name="nom" />
   <input type="submit" value="Valider" />
</p>
</form>
```

- Vous utiliserez winscp pour créer la page cible.php et la copier dans /var/www/html sur la machine linux. Cet accès se fera en ssh.

```
<p>Bonjour !</p>
 <p>Vous vous appellez <?php echo $_POST['nom']; ?> !</p>
 <p><a href="index.html">cliquez ici</a> pour revenir à la page d'accueil.</p>
```

- Sur votre client Windows 10, tester l'accès à la page web via son url est : http://srvweb, montrer l'accès à votre site par les journaux.
- Faites une capture via Wireshark pour voir le protocole http et le nom rentré

2. **Faites que votre site passe par une authentification (cf cours)**

- **nano /etc/apache2/sites-enabled/000-default.conf**

```
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory "/var/www/html">
     AuthType Basic
     AuthName "Restricted Content"
     AuthUserFile /etc/apache2/.htpasswd
     Require valid-user
</Directory>
```

- **Relancer service apache2**

- **nano /etc/apache2/apache2.conf**

```
<Directory /var/www/>
  Options Indexes FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>
```

- **nano /var/www/html/.htaccess**

```
AuthType Basic
AuthName "Restricted Content"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
```

- **Systemctl restart apache2**

3. **Création de deux sites web virtuels en local (LAN), on utilisera le fichier HOSTS pour la résolution de noms**

   www.site1.fr

   www.site2.fr

   Vous personnaliserez la page comme vous le souhaitez.

   Faire les tests d'un client via son URL pour montrer l'accès aux 2 sites we

4. **<u>Création d'un certificat autosigné SSL (</u>**Secure Socket Layer)

   On peut voir que tout est en clair donc pour sécuriser nous allons chiffrer entre le client et le serveur

- Qu'est-ce qu'un certificat autosigné ?
- Quelles différences entre SSL et TLS ?
- Faite le test en https ? Est-ce que cela fonctionne ?
- Lecture pour comprendre le mécanisme

  http://www.sebsauvage.net/comprendre/ssl/

Par défaut Apache 2 contient deux sites préconfigurés : « default » et « default-ssl » qui pointent tous les deux vers le répertoire « /var/www », mais la première écoute sur le port 80 (HTTP) et le second sur le port 443 (HTTPS).

Dans la configuration d'origine, seul le site « default » est actif ce qui permet d'accéder à la page « It Works ! » d'Apache tout de suite après avoir effectué l'installation.

Vu que le site par défaut SSL est préconfiguré pour fonctionner. De ce fait, il suffit d'effectuer :

1. Activer le module SSL d'Apache 2. : a2enmod ssl (va créer le fichier default-ssl.conf)
2. Activer le site « default-ssl » d'Apache : a2ensite default-ssl.conf

Une fois que ces deux activations sont effectuées,

- service apache2 reload
- Relancer l'accès à votre site en https sur IE ou firefox. Que se passe t-il ?
- Afficher le certificat.

Vous remarquerez qu'il 'y a un pb

.

- Ce message s'affiche car notre certificat a été auto-signé par apache2. Pour qu'il soit vraiment valide, il faut que notre certificat soit signé par une autorité qui signe les certificats comme Google Trust Service. Attention à la résolution de noms

SSL Let's Encrypt : Installation et configuration sur Debian 11 | HABEUK

- Capturer une trame avec wireshark. Montrer que l'accès se fait bien en https et que le nom n'est plus lisible (voir son rôle TLS)

5. Mise en place d'un serveur web sur le CLOUD

1. Connectez vous sur les sites suivants et lire les chapitres

   Microsoft Learn | Microsoft Docs

   a) Concepts du cloud computing :

   https://docs.microsoft.com/fr-fr/learn/modules/principles-cloud-computing/

   b) Créer une machine virtuelle Linux dans azure :

   https://docs.microsoft.com/fr-fr/learn/modules/create-linux-virtual-machine-in-azure/

   c) Créer une machine virtuelle windows dans azure :

   https://docs.microsoft.com/fr-fr/learn/modules/create-windows-virtual-machine-in-azure/

2. Connectez vous sur le cloud azure
   https://azure.microsoft.com/fr-fr/free/students/
   https://portal.azure.com/#home

3. Créer un serveur linux sous azure

   

   Attention le coût est excessif si votre machine fonctionne tous le temps. Il faut vraiment penser à l'éteindre quand vous avez fini votre travail, le stockage coute aussi.

   a) Administrer votre machine Linux sur le cloud de votre client Windows en ssh
   b) Reproduire la question N°1
   c) Redeployer toute votre architecture (JSON)
   d) Etudier coût, groupe de ressource, ressources et network + résolution de noms
   e) Faire les tests d'accès au serveur WEB d'un client

4. Même chose sous google cloud

Student Coupon Retrieval Link Google cloud Plateform

Attention le coût est excessif si votre machine fonctionne tous le temps. Il faut vraiment penser à l'éteindre quand vous avez fini votre travail

Qu’est-ce qu’un certificat numérique ?

Un certificat numérique est un type d’identifiant électronique qui peut prouver l’authenticité d’un utilisateur, d’un terminal, d’un serveur ou d’un site web. Il utilise l’infrastructure PKI pour permettre l’échange sécurisé de communications et de données sur Internet.

Cette forme d’authentification est un type de cryptographie qui exige l’utilisation de clés publiques et privées pour valider les utilisateurs.

Les certificats de clé publique sont émis par un tiers de confiance, une autorité de certification, qui signe le certificat, puis vérifie l’identité du terminal ou de l’utilisateur demandant un accès. Pour garantir sa validité, la clé publique sera combinée à une clé privée correspondante dont seul le destinataire a connaissance. Les certificats numériques disposent d’une paire de clés spécifique à laquelle ils sont associés : une clé publique et une clé privée.

Un certificat numérique contient les informations d’identification suivantes :

- Nom de l’utilisateur
- Entreprise ou département de l’utilisateur
- Adresse IP ou numéro de série du terminal
- Copie de la clé publique du titulaire du certificat
- Durée de validité du certificat
- Domaine que le certificat est autorisé à représenter

Qu’est-ce qu’un certificat auto-signé ?

Un certificat TLS/SSL auto-signé n’est pas signé par une autorité de certification (CA) publiquement reconnue, mais par le développeur ou la société responsable du site web. Comme ils ne sont pas signés par une CA publiquement reconnue, ils sont généralement considérés comme peu sûrs pour les applications et sites web publics.

Le rôle d’une CA publique est de garantir la validité des informations incluses dans un certificat, notamment la propriété et/ou le contrôle du ou des noms de domaine associés au certificat dans le cas de TLS/SSL. Par conséquent, l’utilisation de

certificats auto-signés équivaut à l'utilisation d'informations d'identification qui n'ont pas été émises par une autorité valide.

L'expression certificats auto-signés » fait généralement référence aux certificats TLS/SSL qui ont été générés de manière autonome, sans aucun lien avec un certificat racine ou intermédiaire. Elle peut également s'appliquer à d'autres certificats X.509 de signature numérique tels que S/MIME, de signature de code et de signature de document.

La nature des certificats auto-signés implique que les informations figurant sur le certificat n'ont pas été vérifiées par une partie de confiance (une CA publique), et ces certificats déclencheront une alerte de sécurité : Les navigateurs Web et les systèmes d'exploitation détecteront et signaleront les certificats qui n'ont pas été signés par une CA publique de confiance, car ils représentent un risque pour la sécurité de leurs utilisateurs : Le certificat ne provient pas d'une partie de confiance, il pourrait donc être l'œuvre d'un attaquant déployant une attaque de type « man-in-the-middle ».

Ces affichages d'avertissement font fuir les utilisateurs, qui craignent que leurs données personnelles ou financières ne soient en danger en interagissant avec votre site.