Brief 5 : DNS, TLS et plus si affinité

Contexte du projet

Votre patron, par grandeur d'âme, vous a offert un magnifique nom de domaine ".space". Pour travailler. Évidemment.
Vous allez enfin pouvoir être débarassé du calvaire du challenge HTTP de Let's Encrypt ! Fini l'installation d'un serveur web uniquement pour générer un certificat !

Dans ce brief, vous allez générer un certificat TLS avec **Certbot** pour un sous-domaine de votre nom de domaine en utilisant le challenge DNS.

Votre patron, bien avisé, a acheté votre nom de domaine chez un fournisseur moderne qui fournit un plugin **Certbot** pour réaliser le challenge AU-TO-MA-TI-QUE-MENT. C'est beau.

Modalités pédagogiques

Chapitre 1 - Créer un sous-domaine

1. En utilisant l'interface web du registrar, créer un enregistrement A "vote" dans **votre** zone DNS pointant vers l'adresse IP de votre application de vote "Chien-Chat".
2. Vérifier la configuration avec votre navigateur web (il faudra peut-être quelques minutes pour que la configuration se propage jusqu'aux DNS utilisés dans la salle de formation).

Chapitre 2 - Créer un certificat

1. Installer le plugin **Certbot** du registrar :
   2. Pour Gandi : pip3 install certbot-plugin-gandi
   3. Pour OVH : pip3 install certbot-dns-ovh
2. Utiliser **Certbot** pour créer un certificat en utilisant le challenge DNS :
   4. https://github.com/obynio/certbot-plugin-gandi (https://github.com/obynio/certbot-plugin-gandi)
   5. https://certbot-dns-ovh.readthedocs.io/en/stable/ (https://certbot-dns-ovh.readthedocs.io/en/stable/)

Chapitre 3 - Charger le certificat dans Azure Application Gateway

1. Convertir le certificat du format PEM au format PKCS12 (aussi appelé PFX) :
   2. Concaténer le certificat TLS et la clé privée dans un seul fichier
   3. Utiliser openssl pour convertir et protéger le fichier PKCS12 avec un mot de passe : `openssl pkcs12 -export -out <fichier-destination.p12> -in <fichier-source.pem>`
2. Importer le certificat au format PKCS12 dans une Application Gateway existante : https://learn.microsoft.com/en-us/cli/azure/network/application-gateway/ssl-cert?view=azure-cli-latest#az-network-application-gateway-ssl-cert-update (https://learn.microsoft.com/en-us/cli/azure/network/application-gateway/ssl-cert?view=azure-cli-latest#az-network-application-gateway-ssl-cert-update)
3. Ou Créer une Application Gateway avec le certificat (–cert-file et --cert-password) (vous pouvez aussi modifier votre configuration Terraform / playbook Ansible)
4. Activer le HTTPS sur l'Application Gateway sur le port 443
5. Vérifier avec votre navigateur web l'accès à la page de vote en HTTPS

Chapitre 4 - Charger le certificat dans Azure KeyVault

1. Créer un KeyVault sur Azure en utilisant la CLI

2. Importer le fichier contenant le certificat et la clé dans le KeyVault en utilisant la CLI
3. Vérifier que le certificat est dans le KeyVault en utilisant la CLI : `az keyvault certificate show -v <vault-name> -n <certificate-name>`

Chapitre BONUS 5 - Limiter qui peut générer un certificat pour votre domaine

Ajouter un enregistrement CAA à votre domaine pour n'autoriser QUE Let's Encrypt à générer un certificat pour votre nom de domaine.
https://letsencrypt.org/fr/docs/caa/ (https://letsencrypt.org/fr/docs/caa/)

Chapitre BONUS 6 - Relier l'Application Gateway avec l'Azure KeyVault

Pour les warriors :
https://learn.microsoft.com/en-us/azure/application-gateway/key-vault-certs (https://learn.microsoft.com/en-us/azure/application-gateway/key-vault-certs)

Critères de performance

- Le certificat TLS est bien généré avec un challenge DNS
- La zone DNS est correctement configurée
- L'application de vote est maintenant accessible en HTTPS

Modalités d'évaluation

Restitution en groupe.
Relecture commentée de vos livrables par les formateurs.

Livrables

- Le script

Objectifs

À l’issue de ce brief, vous aurez :

- configuré une zone DNS
- utilisé certbot pour déployer un certificat TLS
- configuré le HTTPS sur Application Gateway