Fiches techniques
Génération de certificats

Noé Dallet

19 novembre 2021

1 Problématique

Dans le cadre de ce projet, le client demande d'avoir une autorité capable de délivrer des licences, et un utilisateur qui exécute un programme qui communique des informations sensibles avec cette autorité.

Il faut être en mesure de certifier que chacun des deux partis est bien la personne qu'il prétends être.

Dans le cas contraire plusieurs attaques peuvent être faite à l'encontre du système :

— L'utilisateur demande une clef à l'autorité, par un moyen ou un autre, un pirate s'empare de la connexion de l'utilisateur et se fait passer pour lui puisque la connexion n'est pas protégée. Alors le serveur pense parler à l'utilisateur, et le pirate envois alors son certificat matériel pour générer la licence. La licence est alors dépendante du matériel du pirate, et pas de l'utilisateur qui en a fait la demande, le pirate a volé une licence gratuitement.

— L'utilisateur télécharge son programme mais en réalitée un pirate se fait passer pour l'autorité, qui est aussi en charge de fournir les programmes notamment et au minimum celui qui gère les licences. Alors l'autorité est en mesure de fournir à l'utilisateur un programme modifié, et donc sûrement malveillant.

Une deuxième problématique entre en jeu, l'utilisateur lui même peut essayer de dupliquer sa license, la license doit donc être dépendante du matériel.

2 Solution

Pour être en mesure de certifier l'authenticité des deux partis, on utiliseras des certificats de type "x.509" qui repose sur un chiffrement asynchrone. Dans ce système pour certifier l'authenticité d'un parti, une autorité de confiance doit donner son avis sur ce parti. Dans notre cas, il faut avoir un certificat SSL (certificat "x.509" classique, mais souvent utilisé pour se connecter sur un serveur web), délivré par une autorité de certification dite "racine" (que tout le monde connaît, et à qui tout le monde fait confiance, elle est obligatoirement authentique).

2.1 Certifier le fournisseur du service

Le mécanisme se découpe en six étapes :

1. Le fournisseur du service effectues une demande de certificat incluant une clef publique fraîchement généré, à l'autorité racine
2. L'autorité racine lui chiffre avec sa clef privée et lui envois le résultat
3. Le fournisseur du service envoi sa clef publique chiffrée par l'autorité racine à un utilisateur
4. L'utilisateur du service a déjà la clef publique de l'autorité racine, il l'utilise pour déchiffrer la clef publique du fournisseur du service.
5. Le fournisseur du service envoi ses messages chiffrés avec sa clef privée
6. L'utilisateur du service déchiffre les messages avec la clef de l'étape 4.

Si le message est correctement déchiffré l'émetteur est assuré d'être celui qu'il prétends.

2.2 Certifier l'utilisateur du service

Ici pour certifier l'authenticité de l'utilisateur le même principe est mis en place mais un roulement est fait : le rôle de autorité racine est joué par le fournisseur du service, le rôle du fournisseur du service par l'utilisateur qui cherche à être vérifié par l'utilisateur, et le rôle de l'utilisateur par le fournisseur de service également.
Quésako ?
C'est simple, à la première connexion sur le site avec le bon PC qui fait tourner les programmes sous licence, l'utilisateur effectues l'étape 1, 2, 3, 4 et 5. Tant que le message se décrypte par le fournisseur de service sans nouvelle clef, le PC de l'utilisateur n'as pas changé ou sa clef privée a fuité. Si l'utilisateur change de machine ses données d'authentification sont redemandés, si il est correct, soit il a fuité, soit c'est la bonne personne, alors on ajoute le PC à la liste de confiance du fournisseur et on retourne les étapes 1 à 5.

3 certifier le matériel de l'utilisateur

Pour certifier que l'utilisateur utilise bien la bonne machine pour exécuter les programmes, on doit utiliser une variable qui diffère d'une machine à une autre : les "adresses" (récupérable avec : "" wmic csproduct get UUID "") des différents périphériques du système que l'on autorise pas à être changés : (ici juste la carte mère mais on pourrait autoriser l'installation sur un périphérique usb qui se déplacerais de machines en machines à volonté (ici : "" wmic DISKDRIVE get SerialNumber "")) j'appelerais cela le WMIC dans le reste du document.

Cette information est sensible et ne doit pas être partagée, elle sera brouillée en effectuant un chiffrement symétrique avec elle même comme clef par exemple lors de la communication avec le serveur pour l'associer à une licence.

A chaque besoin de la licence il faudra extraire ces informations WMIC et déchiffrer la licence puis l'utiliser pour lancer le programme sans laisser de traces de la licence.

—¿ problème : comment le fournisseur du service s'assure qu'une même licence ne s'utilise qu'a un endroit de manière sûre ?

L'utilisateur pourra toujours récupérer ou générer les informations intermédiaires (licence déchiffrée par exemple) pour les utiliser manuellement sur un autre ordinateur comme licence valable hors connexion avec le fournisseur du service, ce procédé le rends plus compliqué mais pas impossible de plus rien n'empêche l'utilisateur de faire tourner son programme.