Contrôle d’accès au Loria

Contextualisation

Le LORIA contient des zones avec des niveaux de confidentialité différents. Tous les employés sont admis dans les zones de confidentialité basse. Et seules quelques personnes sont admises dans les zones classifiées “Secret”. Il est utile qu’une personne puisse tout de même effectuer une demande d’accès exceptionnelle et que cette demande soit gérée dans les plus brefs délais.

Architecture

Le réseau est organisé comme suit :

- Des badgeuses à l’entrée de chaque pièce, qui permettent d’octroyer l’accès à la pièce.
- Un serveur central qui vérifie que les permissions des employés utilisant la badgeuse sont conformes à ce qui est attendu.

Rôle du serveur

Le serveur vérifie que les autorisations sont les bonnes et renvoie un message à la badgeuse indicatif de l’action à effectuer.

Première version : naïve

Connaissances initiales

Le serveur connaît les identifiants des employés et des badgeuses et garde en mémoire leur niveau de confidentialité, il connaît de plus sa clef privé et sa clef publique, ainsi que les clefs publiques des badgeuses
La badgeuse connait l’identifiant de l’employé qui vient de badger. Mais aussi son propre identifiant, ses clefs publiques et privées, l’adresse du serveur et sa clef publique.

Protocole

Badgeuse -> Serveur : {IdBageuse, IdEmploye}_PKs

Serveur -> Badgeuse : { Msg }_PKb

Badgeuse -> Serveur: { Msg }_PKs

Limitation

Ce protocole est sensible au rejeu. On peut rejouer le premier message de demande d'accès au serveur, mais aussi envoyer directement le message { Msg }_PKb à la badgeuse.

Deuxième version : Nonces

Connaissances initiales

Les connaissances initiales sont les mêmes, des nonces sont ajoutés afin d'éviter que le deuxième message soit rejoué. Des secrets sur les nonces sont prévus pour empêcher une authentification entre serveur et intrus.

Protocole

Badgeuse -> Serveur : {IdBageuse, IdEmploye, Nb}_PKs

Serveur -> Badgeuse : { Msg , Nb.Ns}_PKb

Badgeuse -> Serveur: { Msg , Ns }_PKs

Limitation

Une attaque de type MitM est possible, l'intrus peut récupérer le Nonce. Et ainsi modifier le message renvoyé par le serveur et par exemple, forcer la porte à rester fermée.

Troisième version : Clef de session

Connaissances initiales

Les connaissances initiales restent identiques, cependant pour éviter le rejeu et l'attaque de type MitM on va mettre en place une clef de session.
On rajoute aussi un TypeAction pour la badgeuse, qui correspond à une demande d'ouverture traditionnelle ou exceptionnelle si d'habitude la personne n'a pas accès à cette zone. Pour le serveur, on rajoute une Action qu'il renvoie à la badgeuse qui autorise ou non l'accès à la zone si l'identifiant de l'employé est autorisé à pénétrer dans la zone.

Protocole

```
Badgeuse -> Serveur : {IdBageuse, IdEmploye}_PKs

Serveur -> Badgeuse : { ClefSession}_PKb

Badgeuse -> Serveur: { Msg }_ClefSession

Serveur -> Badgeuse: { ok .H(Msg)}_ClefSession
```

Discussion sur la gestion d'un accès exceptionnel

On peut imaginer deux façons de gérer une demande d'accès exceptionnelle, soit la logique interne du serveur gère la demande soit un message spécial est envoyé par la badgeuse.
La première version semble la plus légitime, une demande est faite en amont et le serveur lorsqu'il reçoit une demande d'accès vérifie que celle-ci est légitime. Mais la demande d'accès reste la même que habituellement.
La seconde version consisterait à ce que la badgeuse modifie son message lorsqu'une demande exceptionnelle est faite. Une action de l'utilisateur est alors requise pour faire cette demande et elle doit être validée sinon des abus pourrait exister.
Pour palier à cette problématique, nous allons considérer que le message envoyé est de type message (typeAction), le contenu est laissé libre, on peut donc imaginer les deux méthodes la première étant a privilégier. On préfère une logique centralisée.

Mise en place de sécurité

Afin de rendre le protocole plus sûr, des sécurités sont mises en place avec notamment des informations tenues secrètes comme la clé de session entre la badgeuse et le serveur, l'Id de l'employé et du serveur et l'action transmise.
On vérifie les acquittements entre la badgeuse et le serveur.

Limites

Il ya toujours des limites. Ici, nous nous sommes intéressés aux propriétés d'authentification(les IDs), de confidentialité(clé de session + secrecy_of), de non répudiation(propriété de authentication_on). Et a priori, le protocole que l'on a mis en place semble plutôt résistant (voir analyse.pdf (Analyse.pdf)).