# Projet sécurité
# Implémentation d'un réseau clients-serveurs à l'aide du protocole Needham-Schroder

Le but du projet est d'implémenter une version allégée du protocole Kerberos dans un réseau local, sur lequel existent plusieurs "serveurs de services" (SS) et plusieurs clients (un SS peut être lui-même client). Un serveur de certificats (SC) unique permet à tous les SS et clients de se voir authentifiés leurs clés publiques. À tout moment, des nouveaux clients ou SS peuvent rejoindre le réseau. Les clients peuvent demander certains services aux SS, et les demandes de services se font par l'intermédiaire du protocôle Needham-Schroeder, avec ou sans la phase initiale de demande de certificats de la part du SC, en fonction du fait que le client possède déjà ou pas un certificat du SS qu'il veut contacter.

Les nonces qui apparaissent dans le protocole Needham-Schroeder seront utilisées par les "interlocuteurs" (client et SS) pour lancer une session (et une seule), dans laquelle le client demande un certain service au SS qui lui fournit ce service. La session devrait permettre à chaque interlocuteur de s'assurer de l'authenticité de l'autre, et de la confidentialité des échanges.

Les services que les SS peuvent fournir ne sont pas spécifiés, vous pouvez choisir vous-mêmes deux types de services (exemples : lecture/écriture de données dans un fichier local du SS, service d'impression, service d'achats avec liste d'achats stockée sur serveur). Si un client demande à un SS un service indisponible, votre session doit se terminer par un message d'erreur envoyé par le SS au client. Chaque fourniture de service nécessite un acquittement de la part du client, car les services pourraient être facturés. Cette partie de facturation n'est pas spécifiée, vous pouvez imaginer aussi la manière dans laquelle cela peut se passer, sans oublier les aspects de sécurisation ! Un SS pour un type de service peut être aussi client pour un autre type de service. Une interface de gestion permet de configurer le réseau au lancement, en fixant le nombre maximal de SS et de clients existants, et le type de service que chaque serveur fournit.

Dans votre réseau vous devez gérer aussi l'apparition de nouveaux clients. Chaque nouveau client ou SS est supposé posséder la clé publique du serveur de certificats (sous forme d'un certificat auto-signé) et une paire de clé avec un certificat délivré par le SC pour la clé publique. Chaque client aura un keystore local dans lequel il sauvegardera les clés des serveurs une fois les certificats respectifs vérifiés, en plus de sa propre clé privée.

La création des certificats devrait se faire hors-ligne, par le SC. Cela veut dire que vous ne devez pas prévoir des communications réseau pour les demandes de création de certificats,

mais juste la création de fichiers correspondant à un requête de certificat (CSR) coté demandeur et la création du certificat coté SC.

Chaque client qui voudra demander un service devra exécuter les opérations suivantes :

1. Vérifier s'il a stocké un certificat pour le SS dans son keystore, dans quel cas il devrait l'utiliser dans le protocole.
2. Sinon, demander au SC le certificat du SS, le vérifier et le stocker dans le keystore.
3. Accomplir la suite d'échanges de messages du protocôle Needham-Schroder pour créer la 2e nonce, ensuite utiliser cette 2e nonce pour demander le service qu'il souhaite auprès du serveur.

Vous devez implémenter aussi des "clients (ou SS) erronés" qui ne fonctionnent pas de façon correcte – par exemple, en n'envoyant pas les nonces dans le bon ordre dans le protocole Needham-Schroeder, ou en cryptant certaies données avec une mauvaise clé de session, etc. Ces clients erronés devraient être détectés par les serveurs, qui devraient afficher des messages d'erreur.

Attention ! il ne s'agit pas d'implémenter des "attaquants" (de type Dolev-Yao), mais juste des clients ayant un fonctionnement erroné !

La spécification ci-dessus permet d'avoir un projet assez bon. Pour un projet très bon, vous devez choisir parmi les options suivantes :

1. Implémenter ue liste de révocation de certificats (CRL), que le SC doit maintenir. La prise en compte de la CRL peut se faire de deux manières :
   (a) Soit le serveur publie périodiquement une nouvelle CRL qui devra être récupérée par chaque client ou serveur en début de chaque session pour vérifier que le certificat de son interlocuteur n'est pas révoqué.
   (b) Soit en implémentant le protocôle OCSP, donc un SC qui répond en ligne à toute demande de vérification de révocation pour un certificat donné, avec la même consigne pour chaque client/serveur en début de chaque session de vérifier que le certificat qu'il souhaite utiliser n'est pas révoqué.
2. Implémenter l'attaque contre Needham-Schroder telle que vue en CM, en créant un SS malhonnête qui implémente cette attaque. Les communications réseau de chaque client et serveur (sauf le malhonnête) devraient être rédirigées dans des fichiers et le SS malhonnête se chargerait de jouer le rôle du réseau, donc en recopiant le fichier qui représentent le message envoyé par un agent Alice dans le fichier que son interlocuteur Bob attend de lire pour avancer dans le protocôle.

Votre projet doit comporter des jars contenant les différents agents participant et/ou l'interface de gestion, ainsi que l'infrastructure nécessaire au démarrage du réseau (distribution des clés). D'autre part, vous devez écrire un rapport contenant la description assez détaille de vos protocôles, les choix que vous avez fait pour les parties non-spécifiées du projet, les opérations à effectuer au démarrage et la configuration nécessaire. Votre implémentation devrait fonctionner sur n'importe quel type de réseau, et pas seulement sur une seule machine. Une partie essentielle du rapport doit détailler l'implémentation des protocoles de sécurité :

comment les diverses vérifications nécessaires à la validation de chaque étape se déroulent dans votre implémentation, et quels sont les variantes d'échec qui peuvent apparaître à chaque étape. Vos variantes des clients erronés doivent être aussi détaillées.

Vous êtes aussi censés préparer une soutenance de votre projet de 10-15 minutes, lors de laquelle vous allez présenter une suite d'opérations (préparées à l'avance) pour exemplifier le fonctionnement de votre implémentation.