MI11 Master en Architecture des Systèmes Informatiques

UE : Principes de sécurité informatique

M18 Master en Sciences de l'Ingénieur Industriel Informatique

AA : Principes de sécurité informatique

Projet 2 : Serveur HTTPS et 3DSecure

Version 1 du 14/11/2024

2024-2025

Samuel Hiard

Préambule : Les étudiants ingénieurs (M18) ne doivent pas réaliser ce projet car ils ont un énoncé séparé pour leur UE intégrée.

1. Introduction

Ce document a pour vocation de décrire le projet de programmation que vous devrez réaliser pendant le second quart de ce quadrimestre. Il est peut être vu comme un « cahier des charges » décrivant le contexte, l'objectif, les fonctionnalités, les modalités, ce qui est attendu, et éventuellement ce qui est autorisé ou interdit. Il peut être soumis à changement.

2. Contexte

Vous développez un site de vente en ligne pour un entreprise gérant un complexe de vacances. Les futurs clients doivent pouvoir réserver leur séjour en ligne, et confirmer leur réservation via le paiement d'un acompte.

3. Fonctionnalités souhaitées

Vous développerez un serveur HTTPS et accepterez le paiement via 3D-Secure

3.1 SSL et HTTPS (Phase 1)

Le serveur écoutera sur le port 8043 et répondra aux requêtes HTTP(S) envoyées par un navigateur. Vous ne devez pas écrire le code du navigateur.

Vous générerez un certificat RSA avec une clé de 2048 bits. Ce certificat sera auto-signé. Je vous conseille d'utiliser l'outil *keytool* pour la génération.

Vos clients sont enregistrés sur le serveur (soit via un simple fichier, soit dans une base de données). Le login est conservé en clair, et le mot de passe est haché et salé. Avant de pouvoir valider l'achat, les clients vont donc s'authentifier à l'aide d'un formulaire, qui aura pour effet l'envoi d'une requête POST à votre serveur. Si les identifiants sont corrects, l'utilisateur aura alors accès à une page de paiement qui, pour simplifier le projet, consistera simplement en un bouton « paiement » qui, dans cette phase, n'a aucun effet.

Vous développerez également 2 serveurs (ACS et ACQ) qui sont capables de communiquer via SSL en mode end-to-end. L'ACQ doit pouvoir envoyer un message (le contenu n'est pas important à ce stade) et l'ACS lui renvoie une réponse (idem pour le contenu), le tout de manière chiffrée. Le serveur HTTPS devra également pouvoir communiquer de manière sécurisée avec l'ACQ (Le serveur HTTPS initie la connexion à l'ACQ). A vous de gérer le fait que les clés/certificats soient dans les bons keystores/truststores.

3.2 3D-Secure (Phase 2)

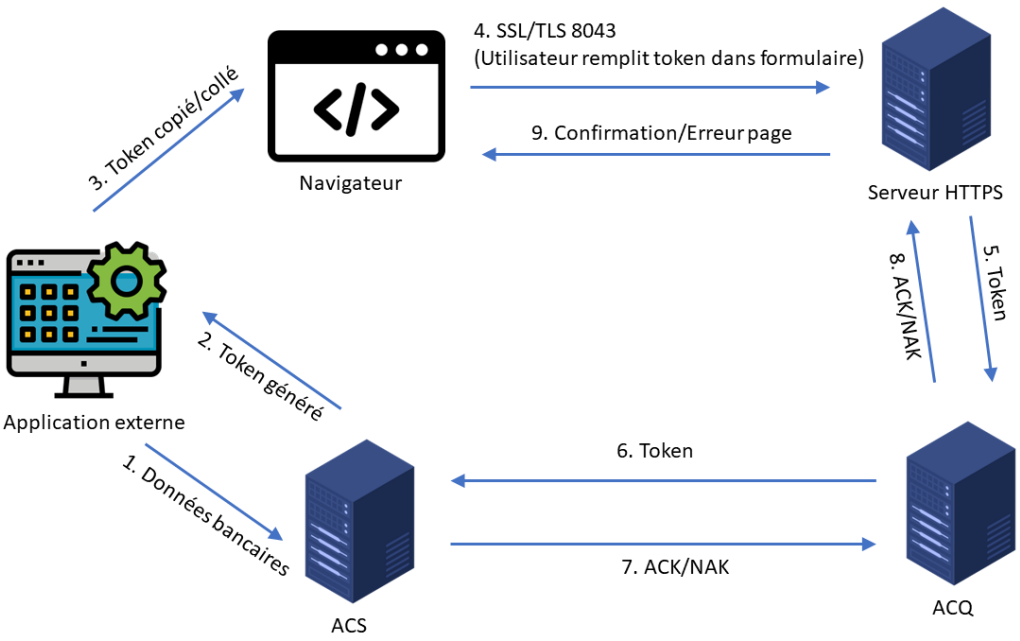
Vous implémenterez une version simplifiée du protocole 3D-Secure.

Le scénario sera le suivant :

1. Via une application tierce (un autre programme), le client enverra la date et le numéro de carte de crédit (qui peut être hard-codé dans l'application) à l'ACS qui écoute sur le port PORT_AUTH (valeur à définir). Ce message sera signé par la clé privée du client. La communication est sécurisée par SSL

2. L'ACS vérifie la signature et le numéro de carte de crédit et génère un code d'authentification (potentiellement aléatoire) et le stocke temporairement. Ce code est signé par l'ACS et renvoyé au client.
3. Le client vérifie la signature et affiche le code généré à l'écran.
4. Sur le site HTTPS, le bouton « paiement » est précédé d'un textfield dans lequel l'utilisateur encode le code reçu par l'application tierce.
5. Lors de l'appui sur le bouton paiement, le formulaire génère une requête POST.
6. Lorsque cette requête est reçue, le serveur contacte l'ACQ pour lui transférer le code qui, en retour, renvoie ce code à ACS (sur un port PORT_MONEY, à définir). Les deux communications sont sécurisées via SSL.
7. Si le code reçu par l'ACS correspond au code généré, alors l'ACS renvoie un ACK (sinon, il renvoie un NACK) à l'ACQ, qui transmet cette information au serveur HTTPS.
8. En fonction de la réponse, la page générée est : « paiement validé » ou « paiement refusé ».

Le schéma de l'architecture est donc le suivant :



4. Consignes supplémentaires

- Ce projet est à réaliser par groupe de 3 étudiants (même malus éventuel que pour le projet 1)
- Il sera présenté le 16 décembre. Le code source sera envoyé au plus tard ce même jour, par mail à samuel.hiard@hepl.be dans une archive (.rar, .zip ou .7z) au format *XXX-YYY*—ZZZ-MasiSecuProj2.*EXT* où *XXX*, *YYY et ZZZ* sont les noms de famille des trois étudiants ayant participé au projet, et *EXT* l'extension (rar, zip ou 7z) de l'archive.
- Le langage de prédilection est le langage Java, mais vous pouvez utiliser un autre langage si vous le souhaitez. Dans ce cas, l'aide apportée par le professeur pendant les laboratoires sera faible, voire nulle.
- Amusez-vous bien !