Un fichier de cartes bancaires

Gilles Grimaud

Janvier 2021

Vous avez sans doute remarqué qu'il existe aujourd'hui deux manières pour un site web de vous demander un paiement. La première solution et de vous demander un n° de carte bancaire + les chiffres de CSV qui figure au dos. La seconde passe par l'utilisation de 3D Secure. 3D Secure devrait s'imposer car il permet de mettre en place un paiement plus sécurisé, où le site marchand demande à la banque d'authentifier le porteur de la carte bancaire. Cependant cette solution de paiement implique que :

1. selon sa banque, chaque utilisateur à une interface d'authentification du paiement différente ;
2. et que l'opération d'authentification soit demandée pour chaque paiement.

Aussi de nombreux acteurs économiques du web la boude. En effet, à cause de (1) le client un peu perdu demande de l'aide au site marchand à propos de quelques chose (l'authentification) qui est du ressort de la banque. Et à cause de (2) le paiement "au click" n'est pas possible.

Ainsi de nombreux sites marchands historiques continuent à utiliser le Système carte bancaire + CSV.

Cette solution, moins protégée encourage le vol de fichiers marchands de carte bancaire. Il peut s'agir d'un vol « informatique » autant que d'un vol « physique », avec éventuellement la compromission de certains administrateurs des systèmes.

Aussi les GIEs Carte Bancaire ont définit des normes et audits précisant les conditions sous lesquelles un serveur en ligne peut légitimement détenir des identifiants de cartes bancaire. Nous ne détaillons pas dans ce sujet les 1200 pages de documentation qui définissent ces normes, mais nous retenons pour l'exercice proposé les points suivants :

1. Le serveur qui stocke et gère les paires nom / n°carte bancaire doit être identifié, il fournit les services suivants :
    i. Mise en service,
    ii. Ajouter une paire,
    iii. Supprimer une paire,
    iv. Chercher les n° de cartes associées à un nom,

2. La sécurité du fichier ne doit pas reposer :
    i. ni sur la sécurité du disque qui contient le serveur (ce disque peut être volé informatiquement ou physiquement),
    ii. ni sur la sécurité des programmes qui fournissent les services (il sont sur le disque),
    iii. ni sur la bonne volonté de l'administrateur du serveur (on peut faire pression sur lui) ;
3. Aussi, les paires nom / n°carte bancaires stockées dans un fichier doivent être chiffrées ;
4. Les clefs de chiffrements ne doivent jamais être (en clair) sur le disque ou le fichier est stocké, mais elles peuvent être stockées, lors de la mise en service du serveur, sur un RAM disk par exemple ;
5. La mise en service du serveur (et les moyens qui permettent de déchiffrer le fichiers) ne doivent être possible que sous l'autorité de deux responsables (un responsable technique et un responsable juridique), qui doivent être physiquement présents et authentifiés par deux facteurs lors du démarrage.

Pour comprendre ces points, il faut comprendre que le modèle de l'attaquant anticipe que ce dernier peut avoir un accès physique ou logique au disque et aux programmes du serveur. Selon le modèle d'attaque, une attaque a échouée, si a l'issue de cette dernière plus personne n'est en mesure d'utiliser le fichier. Cela signifie que dans l'absolue, le dénie de service n'est pas redouté.

Nous vous demandons de réaliser un *proof of concept* du serveur en question. Ce *PoC* démontrera l'usage qui est proposé de la cryptographie, et ce en quoi ils permettent d'atteindre les impératifs de sécurité définit. Le *PoC* pourra consister en autant de script *bash* ou *python* que de services.

Partie 1 : Responsabilité partagée

Les contraintes de sécurité stipulent qu'il n'est possible de démarer le service que sous l'authorité de 2 responsables. Le fichier chiffré doit rester indéchiffrable par quiquonque (même le concepteur du logiciel) en l'absence de l'authentification à 2 facteurs de ces deux responsables. Puisque l'on suppose que l'attaquant dispose du programme, l'obtention de la clef de dechiffrement ne doit être possible qu'en présence de l'authentification des deux responsables, par deux facteurs.

On parle de *security by design*.

Question 1 :

Proposez une solution pour que le déchiffrement ne soit rendu possible que par l'authentification à deux facteurs des deux responsables. Pour des raisons pratiques, les deux facteurs seront

(1) ce que je sais (un *password*) et ;

(2) ce que j'ai (une clef usb).

Pour des raisons pratiques, le *PoC* ne devra pas reposer sur de la biometrie.

Question 2 :

Proposez une implémentation en *bash*, *zsh* ou en *python* des services 1.i., 1.ii., 1.iii. et 1.iv.. De plus, comme la mise en service suppose que les clefs usb des responsables et éventuellement le disque dur aient été initialisés d'une certaine manière, proposez un service 1.v. initialisation qui initialise les deux clefs usb et le disk.

Partie 2 : Délégation de droit

Comme les responsables ne sont pas toujours disponibles et que le bon fonctionnement de ce serveur conditionnement le chiffre d'affaire de la société. Les normes prévoient qu'il soient possible de définir un "représentant légal" pour chacun des deux responsables (dont un représentant technique du responsable technique et un représentant juridique du responsable juridique).

Question 3 :

Proposez une évolution de votre solution afin que les representants puissent se substituer aux responsables en cas de besoin lors de la (re)mise en service du serveur. Le système ne doit pas reposer sur la confusion entre un responsable et son potentiel représentant. Au contraire, il doit être capable de discréminer l'un de l'autre sans pour autant que cela change quoi que ce soit au fonctionnement des services.

Question 4 :

Modifiez en conséquence votre service de démarrage et les autres services si besoin.

Partie 3 : Révocation de droit

Comme il est possible qu'un responsable soit démit de ses fonctions ou qu'il soit répudié, il faut que la gestion du mécanisme d'authentification puisse évoluer en conséquence.

Question 5 :

Proposez un nouveau service : 1.vi. répudiation qui doit permettre au serveur de retirer la responsabilité d'un responsable ou de son représentant. Cette répudiation ne doit pouvoir être menée qu'une fois l'ensemble des ayants droits (a l'exception de la personne répudiée) authentifié, et elle doit être impossible *par design* sans eux.

Question 6 :

Implémentez un service 1.vi. répudation conformément à votre proposition faire en réponse de la question 5.