Authentification

LABO 2

Alec Berney

SEC – 27.05.2022

Questions

1. What are the advantages of a challenge-response authentication compared to a weak authentication protocol?

Tout d'abord, le hash du mot de passe n'est pas transmis au client, ce qui permet de garder un peu plus secret le mot de passe de l'utilisateur et cela apporte donc une sécurité supplémentaire.

De plus, l'authentification est différente à chaque tentative car le challenge utilisé est généré aléatoirement pour chaque tentative. Il est donc impossible de réaliser une attaque où l'on pourrait utiliser le challenge plusieurs fois jusqu'à obtenir un résultat positif.

Finalement, cela permet de ne pas envoyer le mot de passe en clair au serveur lors de l'authentification. C'est une sécurité en plus, car dans un cas classique, si l'utilisateur envoie un mot de passe correcte et qu'un attaquant le récupère entre le client et le serveur, il obtient le mot de passe du client.

2. In your application, when do you require the user to input its Yubikey? Justify.

L'utilisation de la Yubikey apparaît dans les cas suivants :

1. Lors de l'enregistrement du compte,
2. Lors de l'authentification, comme second facteur (si activé),
3. Lors de la réinitialisation du mot de passe pour confirmer l'identité de la personne en plus de l'email fourni.

Lors du point 1, il est nécessaire d'utiliser la Yubikey pour générer la paire de clés (publique et privée). La clé privée est stockée sur la Yubikey, tandis que la clé publique est enregistrée dans la base de données côté serveur. Sans cette étape, il est impossible de faire d'utiliser la Yubikey comme second facteur d'authentification ou pour la réinitialisation du mot de passe.

Lors du point 2, la Yubikey est utilisée comme second facteur et il est possible de demander le PIN de la clé (selon Policy choisie) afin d'authentifier l'utilisateur. La clé signera un challenge envoyer par le serveur, qui sera uniquement déchiffrable avec la clé publique. La correspondance entre les deux challenges sera réalisée et pourra déterminer si la bonne Yubikey est utilisée ou non.

Lors du point 3, le même principe de challenge est utilisé qu'au point 2. La seule différence est que ce n'est pas un deuxième facteur mais une couche de sécurité en plus qu'il est important d'ajouter. Dans certaines applications il est possible de réinitialiser le mot de passe juste en fournissant une adresse email correcte. Ici, on préfère valider que ce soit la bonne personne demandant la réinitialisation du mot de passe avec une authentification via la Yubikey.

Finalement, il aurait également été possible de redemander une authentification via la Yubikey lorsque l'on change si l'on souhaite utiliser le 2ème facteur ou non, mais j'ai considéré que l'utilisateur était déjà connecté et que cela était suffisamment sûr. Cette partie est cependant facile à implémenter.

3. What can you do to handle Yubikey losses?

L'axe de réflexion est surtout basé sur comment empêcher quelqu'un d'utiliser ma Yubikey perdue. Pour ceci, la première chose à faire, est de changer tous les paramètres par défaut.

Il faut donc modifier la Management Key, le PUK et le PIN par défaut. Les paramètres par défaut étant accessible par tout le monde, si la clé ne les utilise plus, il sera donc impossible pour un attaquant d'utiliser la clé avec ces paramètres-là.

Il serait également intéressant de limiter le nombre d'essai concernant le code PIN de la clé, afin de bloquer le compte après x erreurs, par exemple 3.

Il faudrait aussi pouvoir contacter les gérants de l'application de manière sécurisée afin de supprimer momentanément la double authentification et de pouvoir, par la suite, récupérer une nouvelle clé et régénérer la paire de clés comme réaliser dans la partie enregistrement du programme.

4. An attacker recovered the challenge and the associated response (in the password authentication). How does this allow an attacker to perform a bruteforce attack? What did you implement to make this attack hard?

L'attaquant peut réaliser un bruteforce en utilisant un dictionnaire de mot de passes les plus utilisés et s'en servir en passant chaque mot de passe dans la fonction de dérivation de clé, puis générer un MAC et comparer avec le challenge récupérer. Il aura trouvé le mot de passe si les MAC sont égaux. À la place d'un dictionnaire, il pourrait essayer de générer toutes les possibilités de mot de passe existantes mais c'est actuellement impossible de le faire, tellement il y a de combinaisons.

J'ai mis en place plusieurs protections dans mon programme, les voici :

1. La fonction de dérivation, de hash utilise l'algorithme Argon 2. Cette fonction prend un certain temps pour générer le hash et cela ralentit fortement les tentatives de bruteforce.
2. L'utilisation d'un sel aléatoire de 128 bits / 16 bytes, pour dériver le mot de passe à l'aide d'Argon 2. Cela permet, entre autres, de neutralise les rainbow table.
3. Une politique forte a été mise en place pour les mots de passes (cela permet d'éviter au mieux les mots de passes faibles), ils doivent contenir au moins :
    - 1 lettre minuscule,
    - 1 lettre majuscule,
    - 1 chiffre,
    - 1 charactère spécial,
    - Avoir une longueur minimum entre 8 et 64 charactères. (64, pour éviter les DoS).
4. L'utilisation de SHA256 avec HMAC pour générer les MAC pour l'authentification. L'algorithme SHA256 permet de limiter au mieux les collisions.

5. For sending the email, what are the advantages of using an application password?

Les avantages d'utiliser une application sont les suivants :

- Les mots de passes et secrets ne sont pas stocker en clair dans le code.
- Un accès unique au service mail peut être configurer. C'est-à-dire qu'il est possible de restreindre l'application à certaines fonctionnalités du service mail. Par exemple, l'application pourra uniquement envoyer des emails et non lire la boîte mail de l'utilisateur.
- Cela évite l'implémentation d'un serveur mail local. Ce qui est un travail compliqué, surtout si l'on souhaite l'authentifier.
- Si l'on utilise Gmail, cela permet de ne pas être dépendant d'un réseau, par exemple, celui de la HEIG-VD.
- En cas de vol de secret ou autre situation, il est très simple de révoquer les codes pour les applications touchées.

6. In the Yubikey, what is the purpose of the management key, the PIN and the PUK?

Tout ce qui va être expliquer dans la réponse à cette question est tiré de la documentation officielle de la Yubikey, disponible à la page suivante :

https://developers.yubico.com/PIV/Introduction/Admin_access.html

Cette page définie également un tableau d'exemple indiquant quelles actions exigeant / demandant quels outils (MGM Key, PIN, PUK).

Action	Requires MGM	Requires PIN	Requires PUK	Notes
Generate key pair	x			
Change MGM	x			Requires a new MGM
Change retry counters	x	x		Yubico extension. Resets PIN and PUK to defaults
Import private key	x			
Import certificate	x			
Set CHUID	x			
Reset card				Requires both PIN and PUK to be blocked
Verify PIN		x		
Sign data		x		
Decrypt data		x		
Change PIN		x		Requires a new PIN
Change PUK			x	Requires a new PUK
Reset PIN (unblock)			x	Requires a new PIN

Tout d'abord, il est nécessaire de préciser que le protocole PIV propose 2 différents niveaux d'accès à la Yubikey. En résumé, il y a l'utilisateur classique qui détient et utilise la clé tous les jours et de l'autre côté, l'administrateur de cette dernière, protégé par la « management key » qui permet provisionner les informations d'identification.

Le but est donc de définir un système d'autorisation à l'aide d'une liste d'actions définies en fonction de son niveau d'authentification.

En résumé, voici un tableau indiquant l'utilité, à quoi servent chacun des « outils » énoncé dans la question, dans une situation assez classique :

Outil	Utilité
MGM Key	Clé ayant les droits « admin », permettant, par exemple de gérer les certificats, générer les clés (publiques et privées) et de changer le compteur du nombre de tentatives incorrectes. Connue par les administrateurs.
PIN	Code permettant d'authentifier l'utilisateur utilisant la Yubikey. Le code PIN peut être bloqué après 3 tentatives incorrectes d'affilées. Permet, par exemple, de réaliser des signatures et du chiffrement et déchiffrement sous code PIN valide. Utilisé par un utilisateur lambda.
PUK	Code permettant de réinitialiser le code PIN bloqué après 3 tentatives ratées consécutives. Connu par les administrateurs.