Alec Berney

SEC – 26.05.2022

Authentification

Labo 2

# 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. Il est donc impossible de réaliser une attaque où l’on pourrait utiliser le challenge plusieurs fois jusqu’à obtenir un résultat positif.

De plus, l’authentification est différente à chaque tentative car le challenge utilisé est généré aléatoirement pour chaque tentative.

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,
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 5.

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?**

Un attaquant a récupéré le défi et la réponse associée (dans l'authentification du mot de passe). Comment cela permet-il à un attaquant de réaliser une attaque par force brute ? Qu'avez-vous mis en place pour rendre cette attaque difficile ?

Hash avec argon 2 + sel 128 bits / 16 bytes -> pas de rainbow table

Une politique forte a été mise en place pour les mots de passes, ils doivent contenir au moins :

* 1 lettre minuscule
* 1 lettre majuscule
* 1 chiffre
* 1 charactère spécial

Et doivent avoir une longueur de minimum 8 charactères.

hmac

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

* Accès uniquement au service mail
* Évite l’implémentation d’un serveur mail local

Serveur mail local -> nulle

Serveur mail local authentifié -> super dur à faire

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

Dans le Yubikey, à quoi servent la clé de gestion, le PIN et le 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 indiquant quelles actions exigent / demandent quelles fonctionnalité (MGM Key, PIN, PUK).

MGM Key : Clé permettant de gérer les certificats et clés. Définie par un administrateur.

PIN :

PUK :