Qu'est ce que l'authentification ?



L'identification est le procédé qui permet de vérifier l'identité d'un utilisateur. L'objectif de cette opération est de différencier les utilisateurs les uns des autres. Pour ce faire, l'utilisateur répond à la question : ***Qui suis-je ?*** en fournissant un identifiant.

L'authentification est le procédé qui permet de vérifier qu'un utilisateur identifié est bien celui qu'il déclare être. Pour le prouver, l'utilisateur répond à la question : ***Que sais-je ?*** en fournissant un mot de passe.

L'authentification à deux facteurs est un procédé qui complète l'authentification afin de vérifier de manière encore plus certaine que l'utilisateur authentifié n'est pas un individu malveillant. Pour le prouver, l'utilisateur répond à la question : ***Que possèdè-je ?*** en :

- fournissant un code aléatoire obtenu sur un générateur OTP ;
- fournissant un code reçu par SMS ;
- validant un notification sur une App mobile.

L'habilitation est le procédé qui, lorsqu'un utilisateur a été authentifié, permet de déterminer quels accès lui seront accordés. L'habilitation consiste à appliquer une politique de droits pré-établie répondant à la question : ***Qu'as-tu le droit de faire … ou pas ?***.

Notes :

- Aujourd'hui, l'authentification à deux facteurs s'appuie largement sur les smartphones (qui concrétisent naturellement “ce que je possède”). Elle est réellement plus exigeante et, pour s'en convaincre, il suffit de faire l'expérience de ne plus avoir accès à sont smartphone (perte, oubli, panne, etc.) au moment où le second facteur est nécessaire. La situation peut être chaotique, voire kafkaïenne ;
- Pour un organisation, la multiplication des besoins d'authentification a conduit à fabriquer des guichets centralisés d'authentification, appelés Single Sign-On (SSO).

Comment créer une authentification

Voici un exemple de code PHP pour se connecter à une base de données SQL avec un nom d'utilisateur et un mot de passe :

Voici un exemple de code PHP qui peut être utilisé pour vérifier le nom d'utilisateur et le mot de passe que l'utilisateur entre dans un formulaire, et qui est vérifié dans une base de données SQL :

Les bonnes pratiques de l'authentification web

Les **bonnes pratiques de l'authentification web** comprennent plusieurs éléments clés pour garantir la sécurité des utilisateurs et des systèmes. L'un de ces éléments est **le hachage des mots de passe**. Le hachage des mots de passe consiste à utiliser **un algorithme de hachage pour transformer un mot de passe en une chaîne de caractères aléatoires appelée “hash-code”**.

Le hash-code est stocké dans la base de données, au lieu du mot de passe lui-même. Lorsqu'un utilisateur entre son mot de passe pour s'authentifier, le système hache ce mot de passe et **le compare au hash-code stocké dans la base de données**. Si les deux hash-codes correspondent, l'utilisateur est authentifié.

Pour maximiser la sécurité, il est important d'utiliser** le hash-code le plus long possible**. Actuellement, **le hash-code le plus long est de 256 bits**, ce qui offre une très grande sécurité contre les attaques de déchiffrement. En outre, **il est important d'utiliser l'algorithme de hachage le plus lent possible. Actuellement, l'algorithme le plus lent est BCrypt**, qui utilise une fonction de hachage très complexe qui prend beaucoup de temps à exécuter. Cela empêche **les attaques de force brute**, qui consistent à essayer de deviner un mot de passe en exécutant rapidement des milliers d'essais.

En plus du hachage des mots de passe, il est important d'établir une politique des mots de passe pour gérer les mots de passe des utilisateurs. Cette politique doit inclure des règles sur la longueur et la complexité des mots de passe, ainsi que sur leur durée de vie. Elle doit également inclure un seuil de blocage en cas d'échecs répétés, ainsi que des règles sur la réinitialisation des mots de passe en cas d'oubli. Enfin, la politique doit définir le moyen et le mode de communication initiale des mots de passe aux utilisateurs.

En utilisant le hachage des mots de passe, en établissant une politique des mots de passe et en suivant ces bonnes pratiques, vous pouvez renforcer la sécurité de votre système d'authentification web. Cependant, il est également important de stocker les mots de passe de manière sécurisée, afin d'éviter qu'ils ne tombent entre de mauvaises mains en cas de fuite de données. Pour cela, il est recommandé d'utiliser un coffre-fort à mot de passe, qui est un outil de stockage et de gestion sécurisé des mots de passe

Les Fonctionnalités à considérer lors de l'authentification web

Gérer les nouveaux utilisateurs et les désactivations

Pour gérer efficacement l'authentification web, il est important de considérer plusieurs fonctionnalités clés. Tout d'abord, il est important de pouvoir **gérer les nouveaux utilisateurs et les désactivations**. Cela signifie qu'il faut être en mesure de **créer de nouveaux comptes** pour les utilisateurs autorisés, tout en ayant la **possibilité de désactiver les comptes des utilisateurs qui ne sont plus autorisés à accéder au système**.

Gérer les droits (autorisations et groupes)

Ensuite, il est important de pouvoir **gérer les droits des utilisateurs**. Cela implique de pouvoir **attribuer des autorisations et des groupes spécifiques à chaque utilisateur**, afin de contrôler leur **accès aux différentes parties du système**. Cela permet d'éviter que les utilisateurs n'accèdent à des informations ou des fonctionnalités auxquelles ils ne sont pas autorisés.

Gérer la double authentification

Il est également important de pouvoir gérer la **double authentification**, pour **renforcer la sécurité du système**. La **double authentification** implique de demander à l'utilisateur de fournir deux éléments d'identification pour accéder au système, comme **un mot de passe et un code à usage unique envoyé par SMS**. Cela empêche les utilisateurs non autorisés d'accéder au système même s'ils connaissent le mot de passe.

Gérer le changement de mot de passe et la réinitialisation des mots de passe

En outre, il est important de pouvoir gérer **le changement de mot de passe des utilisateurs**. Cela permet aux utilisateurs de **changer leur mot de passe régulièrement pour maintenir la sécurité du système**. Il est également important de pouvoir gérer la **réinitialisation des mots de passe en cas d'oubli**, afin que les utilisateurs puissent continuer à accéder au système sans interruption.

Gérer la communication et le stockage sécurisés

Enfin, il est important de gérer **la communication et le stockage sécurisés des informations d'authentification**. Cela implique de **crypter les informations d'authentification pour les protéger contre les attaques de pirates informatiques**. Il est également important de gérer **la confidentialité des informations d'authentification, en appliquant les droits d'accès aux utilisateurs en fonction de leurs autorisations et groupes**.

Quel rapport a le RGPD avec

l'authentification web ?

Le Règlement général sur la protection des données (RGPD) ne traite pas spécifiquement de l'authentification web, mais il s'applique généralement aux données personnelles collectées lors de l'authentification en ligne. Le RGPD établit des exigences en matière de protection de la vie privée des personnes, notamment en ce qui concerne la collecte, l'utilisation et la conservation de leurs données personnelles. Il exige notamment que les entreprises qui collectent des données personnelles obtiennent le consentement des personnes concernées et leur fournissent des informations claires et précises sur la façon dont leurs données seront utilisées. En outre, le RGPD exige que les entreprises prennent des mesures de sécurité appropriées pour protéger les données personnelles qu'elles collectent, y compris lors de l'authentification en ligne.

Le RGPD pour s'assurer du bon fonctionnement pourra sous un délai de 6 mois, demander le retracement des connexions dans un fichier LOG.

Voici un exemple de code PHP qui peut être utilisé pour écrire des entrées dans un fichier log :

Ce code ouvre le fichier log en mode écriture (a pour “append” ou ajout), ce qui signifie que les données seront ajoutées à la fin du fichier sans écraser son contenu existant. Le message à écrire dans le fichier log est ensuite écrit à l'aide de la fonction fwrite() et le fichier est fermé avec la fonction fclose().

Il est important de noter que le fichier log doit avoir les autorisations d'écriture correctes pour que ce code fonctionne correctement. Vous devrez peut-être modifier les autorisations du fichier pour permettre à PHP d'écrire dans le fichier.

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc3:authentification&rev=1680106994

Last update: **2023/03/29 18:23**