2 Hashing

2-1 Usage

Exécutez les commandes :

md5sum texteAscii
sha1sum texteAscii
sha256sum texteAscii
sha512sum texteAscii

Que remarquez-vous

Rechercher dans la manuel du programme wc (man wc) l'option permettant de compter le nombre d'octets pour compter ceux des hash produits par les programmes md5, sha1 ...

Indiquez cette commande :

wc - Afficher le nombre de lignes, de mots et d'octets d'un fichier

Indiquez les longueurs des hash produits :

md5 : 32 caractere
sha1 : 41 caractere
sha256 : 53
sha512 : 141

2-2 Mots de passe

Le hashing est utilisé pour sécuriser les mots de passes. Expliquez comment :

il va crée une emprunte que seul celui possèdent la clé de hachage pourra déchiffrer

Les mots de passe d'une distribution Linux sont hachés et stockés dans le fichier /etc/shadow et non le fichier /etc/passwd (le nom est un faux ami)

Exécuter la commande sudo cat /etc/shadow

Seul les comptes utilisateurs autorisés à s'identifier sur le système possèdent un mot de passe. Indiquez les noms d'utilisateurs des comptes avec mot de passe sur la VM Debian et leur id. Les hash sont au format idsalt$hash.

etu1 : yj9T$jWCkxfQNrI9p.UJ79BV1U0$fr1j4VLY7UxUygRSUTpY9iyxPwDJh5NPsFgEmdSh

etu2 :yj9T$jWCkxfQNrI9p.UJ79BV1U0$fr1j4VLY7UxUygRSUTpY9iyxPwDJh5NPsFgEmdSh

Les id permettent d'identifier la méthode de hashing. Le site https://hashcat.net/wiki/doku.php?id=example_hashes en références beaucoup. Dans la colonne des hash on peut observer cette structure idsalt$hash mais pas en permanence. Il faut comprendre que l'id ne dépend finalement que du programmeur, le salt s'additionne au mot de passe en clair et le hash est celui du salt et du mot de passe en clair. Le salt permet d'éviter que deux utilisateur utilisant le même mot de passe se voie attribuer le même hash. De plus il permet d'éviter tous piratage utilisant une Rainbow Table.

L'objectif étant de rendre le plus difficile possible l'action inverse du hashing à partir du hash, il est difficile d'admettre que l'id et le salt soient indiqués. Cependant dans une distribution Linux, le compte root est très protégé.

La fonction crypt() de php retourne des hash calculés avec différents algorithmes.

Indiquez les algorithmes indiqués sur le page https://www.w3schools.com/php/func_string_crypt.asp

Les différents algorithmes :
crypt blow fish

2-3 Vérifier un checksum

Calculer le hash d'un fichier permet de vérifier que ce fichier n'a pas été modifier depuis le dernier hash.

Créer 3 fichiers avec les commandes suivantes :
echo "texte du premier fichier" > fichier1
echo "texte du deuxième fichier" > fichier2
echo "texte du troisième fichier" > fichier3

Vérifier la présence des fichiers avec la commande ls -l

Vérifier le contenu des fichiers avec la commande cat fichier1

Exécuter la commande sha256sum fichier1.
Quelle est la structure de la réponse :

elle recalcule la somme des hash attribuer au par avant

Exécuter la commande suivante qui va enregistrer les différents hash des fichiers dans le fichier hashList : sha256sum fichier1 fichier2 fichier3 > hashList

Exécuter la commande cat hashList pour vérifier que les hash de chaque fichier sont présents suivis du nom de chaque fichier

Exécuter la commande echo "un peu de texte en plus" >> fichier1 pour ajouter "un peu de texte en plus" à la fin de fichier1

Exécuter la commande cat fichier1 pour vérifier la modification.

Exécuter la commande sha256sum -c hashList qui va vérifier l'égalité des hash actuels et ceux enregistrés dans hashList. Indiquez la réponse de la commande. Que concluez-vous ?
Quelle fonction de cybersécurité (Disponibilité, Intégrité, Confidentialité) peut être effectuée avec cette méthode.

fichier1: Échec
fichier2: Réussi
fichier3: Réussi
sha256sum: Attention: 1 somme de contrôle ne correspond pas

ont en conclue que le fichier1 marque echec car une parti du fichier n'est pas chifrrer