TP CYBERSECURITÉ – HASH & MOTS DE PASSE (SIMULATION)

Identification d'un hash MD5 + démarche d'audit encadrée

Niveau : débutant / intermédiaire

Environnement : labo uniquement (VM Kali autorisée, données fictives)

==========

CONTEXTE

==========

Lors d'un incident simulé, une fuite expose des identifiants avec un mot de passe stocké sous forme de hash. Votre mission est d'identifier le type de hash, d'expliquer les risques, puis de vérifier (en labo) si le mot de passe est faible via un outil d'audit.

IMPORTANT :

- Ce TP est réalisé UNIQUEMENT sur un environnement autorisé (labo).
- Aucune tentative sur des systèmes réels.
- Vous avez le droit d'utiliser ChatGPT pour vous aider à comprendre, vous guider dans la démarche et vérifier vos hypothèses.

==========

DONNÉES FOURNIES (EXTRAIT LEAK SIMULÉ)

==========

Site : http://cible.labo

Login : toto

Hash : f71dbe52628a3f83a77ab494817525c6

IP : 192.168.56.101

==========

PARTIE 1 – RECONNAÎTRE UN HASH (MD5)

==========

Objectif :

Identifier le type de hash et expliquer comment on le reconnaît.

Travail demandé :

1) Observer le hash :

f71dbe52628a3f83a77ab494817525c6

2) Relever ses caractéristiques :

- longueur (nombre de caractères)
- type de caractères (hexadécimal ?)
- présence ou non de “salt” visible (ex: 1, $2y$, etc.)

3) Formuler une hypothèse :

- Quel est le type de hash le plus probable ? (MD5, SHA1, SHA256, etc.)
- Justifier votre réponse.

Questions :

- Pourquoi un hash MD5 est considéré comme faible aujourd’hui ?
- Quelle différence entre “hash” et “chiffrement” ?

Livrable :

Réponses rédigées (8 à 12 lignes).

==========

PARTIE 2 – RISQUES & BONNES PRATIQUES DE STOCKAGE

==========

Objectif :

Comprendre les conséquences d’une fuite de hash et proposer des mesures correctives.

Travail demandé :

1) Expliquer pourquoi même un hash (sans mot de passe en clair) est dangereux.

2) Citer au moins 3 mesures correctives :

- côté utilisateur (mots de passe)
- côté application (stockage)
- côté organisation (procédures)

Questions :

- Quel algorithme est recommandé pour stocker des mots de passe et pourquoi ?

(pistes : bcrypt, scrypt, Argon2, facteur de coût, salt)

Livrable :

Liste structurée + courte justification.

==========

PARTIE 3 – AUDIT EN LABO : VÉRIFIER SI LE MOT DE PASSE EST FAIBLE

==========

Objectif :

Dans le cadre du TP, vérifier si le mot de passe correspondant au hash est un mot de passe très courant (faible).

Cadre :

- Données fictives.
- Autorisation explicite.
- But : démonstration pédagogique sur la faiblesse de MD5 + mots de passe courants.

Méthode (niveau “procédure”, sans recette d’attaque) :

1) Préparer un fichier de travail sur Kali contenant le hash fourni (1 hash par ligne).

2) Utiliser un outil d'audit de mots de passe (ex : John the Ripper) dans un mode prévu pour les formations/CTF afin de tester des mots de passe très courants.

3) Noter si un mot de passe est retrouvé, puis analyser ce que ça implique.

Consignes :

- Vous pouvez vous aider de ChatGPT pour comprendre les termes et l'outil.
- Vous devez respecter les règles du labo (cible fictive uniquement).
- Si aucun mot de passe n'est trouvé rapidement, expliquez pourquoi (ex : liste de test, mot de passe moins courant, algorithme plus résistant, etc.)

Livrable :

- Indiquer "faible / non démontré faible"
- Justifier en 3–5 lignes

==========

PARTIE 4 – COMPTE-RENDU (SYNTHÈSE)

==========

Objectif :

Rédiger une mini synthèse "incident" orientée sécurité.

À rendre :

1) Type de hash identifié et justification
2) Risques (2–3 points)
3) Mesures correctives prioritaires (3 points)
4) Conclusion : "Pourquoi MD5 ne doit pas être utilisé pour des mots de passe ?"

==========

FIN DU TP

==========