P.Lafourcade, G.Marcadet, M.Puys

TP2 Authentification

Exercise 1 (Sécurité des mots de passe (2 points))

1. Combien y a t'il de mots de passe différents ayant exactement 8 caractères, sachant que seuls les caractères autorisés sont alphanumériques, `-` et `_` ? Expliquez votre calcul et donnez le résultat sous la forme d'une puissance de 2.

2. Combien y a-t-il de mots de passe de 8 caractères, en supposant que les 128 caractères ASCII peuvent être utilisés ? Expliquez votre calcul et donnez le résultat sous forme de puissance de 2.

3. Supposons qu'il nécessite 1 jour pour effectuer une attaque par force brute sur un mot de passe de la question 1. Combien de jours avons-nous besoin pour effectuer une attaque par force brute sur un mot de passe de la question 2 ?

4. En supposant que le processus d'authentification n'utilise pas de sel pour calculer le hachage du mot de passe, quelle méthode peut être conseillée pour casser les mots de passe plus efficacement que l'attaque par force brute ?

Exercise 2 (Adobe (2 points))
Retrouver dans cette liste d'utilisateurs le plus de mots de passe possible en vous appuyant sur les indices. Confirmer que vous possédez bien le bon mot de passe en calculant par vous-même le hash (via `MD5`) et le comparer avec ceux présents dans le tableau.

Nom d'utilisateur	Indice	Hash
Inès	Viande populaire	e6a0ec03e2438e4b0d269f90a2d43e8b
Clara	Jeu de type Battle Royal	972b040498eb76771f064dd2c761edc9
Adam	Détective	c3e3b26d5591c7fcc326a7d399d734bb
Enzo	Sentiment positif	a4731d662cca04cc4e5e0de00714a50c
Hugo	Phénomène météorologique	2d304896fb01357a2dc75ae066138ca9
Emma	Destination de vacances	d366780b44fb6a9382339429a614826b
Léa	Film de science-fiction	69fa47f2852aa102fe7a075e3f403a47
Gabriel	Animal domestique	55dcdf017b51fc96f7b5f9d63013b95d
Lucas	Espace extérieur aménagé	3e22fc28f510cef90163578ad8006630
Sarah	Plateforme de streaming en direct	6a057ee64a34177e388abe7304fd3d78
Jules	Instrument de musique à cordes frottées	57a49eff49945af8a9c320b91180cd57
Chloé	Explorateur de l'espace	3725aaff1c293d536aed26604d874f45
Léo	J'aime les énigmes	c3e3b26d5591c7fcc326a7d399d734bb
Manon	221b	c3e3b26d5591c7fcc326a7d399d734bb
Camille	Appareil électronique	853ca16bda4f3d303e70e48db81c17c6

Remarque

Les exercices suivants traiterons des sujets différents et utiliserons des outils variés. Nous vous recommandons de faire un dossier par exercice pour mieux vous organiser.

Exercise 3 (Brute-force locale (3 points))
Récupérer sur ce site phpbb[1] la base de donées utilisateurs. Vous y trouvez aussi de nombreuses bases de données de mots de passe qui ont fuité ces dernières années :

[1] À une de ces deux adresses: `https://wiki.skullsecurity.org/index.php?title=Passwords` ou `https://sancy.iut.uca.fr/~lafourcade/phpbb.txt.bz2`

1. (1 point) `$1$AAAA$H9wXcd/WaaomJUgWKFspy.` est un mot de passe hashé et salé, retrouver l'agorithme de hachage utilisé, identifier le sel (`rtfm openssl`[2]). Avec `openssl` et la commande `passwd` vérifier que le mot de passe est `!!1331xxx`

2. (2 points) Quel sont les mots de passe associés à `$1$BABA$DOzBWHNx08SgVSX/YuYvC/` et à `$1$CACA$XLWo4OqFFCYICqYrZ0y5i/` à partir de `phpbb`.

   Vérifier sur ce site `https://unix4lyfe.org/crypt/` que vos calculs sont corrects.

Exercise 4 (Brute-force en ligne (3 points))
Récupérer l'archive[3] contenant tout le nécessaire pour cet exercice.

1. Vérifier les dépendances nécessaires avec la commande `pip3 install -r requirements.txt`.

2. Lancer le site `BruteMe` via la ligne de commande `flask run` dans le dossier `site` contenant le fichier `app.py`. Le terminal doit rester ouvert pendant tout l'exercice.

3. Monter une attaque permettant de retrouver le mot de passe par force-brute de votre utilisateur, en utilisant le fichier `callow.py` disponible dans le dossier `callow`.

Exercise 5 (EFAIL (3 points))
Télécharger l'archive[4] contenant l'ensemble des fichiers nécessaire pour cet exercice, et lire le fichier `README.md`. L'objectif est de compléter le fichier `efail_exercice.py` pour monter l'attaque EFAIL vue en cours.

Remarque

Le programme fourni nécessite de faire des requêtes sur le `localhost`. À l'IUT, il faut au préalable désactiver le proxy via la commande suivante:

```
unset http_proxy && unset https_proxy
```

Une fois l'exercice terminé, il faut remettre le proxy à l'aide de la commande suivante:

```
export http_proxy=http://193.49.118.36:8080
export https_proxy=http://193.49.118.36:8080
```

Exercise 6 (Cookie (1 point))
Télécharger *SecureLogin*[5] et passer administrateur, même sans connaître le mot de passe.

1. Une fois l'application téléchargée et extraite (`tar -zx secure-login.tar.gz`), il faut la lancer lancer avec `flask run`.

2. Aller sur votre navigateur favori sur `http://localhost:5000` et passer administrateur.

Exercise 7 (Json Web Token (2 points))
Un Json Web Token (JWT) est un token d'authentification qui permet de prouver son identité. Il se décompose en trois parties: La première contient les informations de génération du token, la seconde le contenu du token en lui-même (ID du token, durée de validité, etc), et la troisième qui constitue la signature du token, qui est en fait un MAC.

1. Aller sur le site `https://jwt.io` pour construire des tokens JWT. Vous utiliserez le mot de passe de votre choix, de quelques caractères (au plus 3).

[2] `openssl -1 -salt BBBB 'Toto'`
[3] `https://perso.limos.fr/~gamarcadet/enseignement/websec23/tp2/brute-me.tar.gz`
[4] À cette adresse `https://sancy.iut.uca.fr/~lafourcade/SECWEB/efail.tar`
[5] `https://perso.limos.fr/~gamarcadet/enseignement/websec23/tp2/secure-login.tar.gz`

2. Télécharger l'outil `jwtcracker`[6] pour brute-forcer le token et utiliser-le sur votre token.

3. Retrouver le mot de passe utiliser pour forger le token dans le fichier `token.txt`.

Exercise 8 (PGP (4 points))
PGP (*Pretty Good Privacy*) est un logiciel qui permet à ses utilisateurs de transmettre des messages chiffrés ou signés. Il suit le standard OpenPGP. L'implémentation GNU du standard OpenPGP s'appelle GnuPG ou GPG (*GNU Privacy Guard*).

Objectif : Prendre en main l'outil `gpg` en ligne de commande.

1. Avec `gpg --gen-key`, créez une clef OpenPGP permettant la signature avec DSA et le chiffrement avec ElGamal[7]. Explorez les différents choix de paramètres disponibles. La durée de validité de la clef doit être de 10 jours. La phrase de passe ne doit pas être vide. [8]

2. Avec `gpg --list-key`, affichez la liste des clefs de votre trousseau.

3. (2 point) À partir du menu d'édition (`gpg --edit-key`):
   (a) Affichez l'empreinte (*fingerprint*) de votre clef.
   (b) Changer la date d'expiration.
   (c) Ajoutez au nom d'utilisateur (`uid`) une image à votre clef, par exemple une photo d'identité numérique.
   (d) Ajoutez une sous-clef de chiffrement RSA 4096 expirant dans 4 jours.
   (e) Modifiez les préférences de votre clef pour mettre AES192, SHA256 comme fonction de chiffrement symétrique et de de hachage.
   (f) Quittez le menu d'édition en enregistrant les changements.

4. Supprimer la première clef et créer une clef associée avec votre adresse email de l'UCA en mettant les bonnes informations.

5. (1 point) Avec `gpg --gen-revoke`, créez un certificat de révocation de la clef que vous venez de créer. Pourquoi créer un tel certificat?

6. (1 point) Avec `gpg --export`, exportez votre clef au format binaire et au format texte (`--armor`).

7. (1 point) Créez un message chiffré pour `gael.marcadet@uca.fr`[9] ce message sera signé et contiendra votre clef publique. Pour cela installez un plugin GPG sur votre client de messagerie préféré (par exemple `Enigmail` pour Mozilla, `gpgtools` sous Mac, `Gpg4win` ou `kleopatra` sous Windows, `APG` sous Android, ou configurer `mutt` etc.). Avant de m'envoyer votre message faites des tests avec vos camarades.

8. (1 point) Téléchargez un message texte (quelques centaines de Ko, par exemple un livre dans le domaine public sur le site `http://www.gutenberg.org/`). Chiffrez-le pour votre binome et observer la taille du message chiffré par rapport au message clair. Vérifier que le fichier se déchiffre bien. Faire de même avec un fichier mp3. Expliquer les différences.

   ```
   gpg --encrypt --armor <filename>
   gpg --output <new filename> --decrypt <filename>
   gpg --sign <filename>
   gpg --verify <filename>
   ```

[6] `https://perso.limos.fr/~gamarcadet/enseignement/websec23/tp2/jwtcracker.tar.gz`
[7] `man gpg` ou `gpg --help` seront vos amis.
[8] Si erreur de `timeout` dû à l'entropie, les commandes suivantes peuvent vous aider `rm -r .gnupg` dans votre HOME, puis `gpg --full-gen-key`
[9] Elle se trouve sur la page web, `perso.limos.fr/~gamarcadet/gpg/pk.gpg`

9. (1 point) Chiffrez maintenant le même message pour vous et pour votre binôme une seule fois. Observez la taille du nouveau message chiffré. Vérifier que le fichier se déchiffre bien par les deux destinataires.

   Chiffrez maintenant les deux mêmes messages en utilisant un algorithme de chiffrement symétrique AES256. Que constatez-vous ?