Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Sécurité des usages TIC

TP n°1

L'ANSSI, « Agence nationale de la sécurité des systèmes d'information »

1 – Allez sur le site de l'ANSSI, `http://www.ssi.gouv.fr/` et consultez le document suivant :

- « Guide d'hygiène informatique » :
  - `http://www.ssi.gouv.fr/administration/guide/guide-dhygiene-informatique/`

Mot de passe & Attaque « *bruteforce* »

2 – Allez sur le site `https://passwordmeter.com/` et testez des *mots de passe*.

a. Est-ce que le mot de passe que vous utilisez habituellement est aussi robuste que vous le pensez ?

b. Essayez le mot de passe « Sécurité ». Quelle est sa robustesse ?
Est-il facilement trouvable ? Comment ?

c. Que pensez vous des règles d'évaluation proposées par ce site ?

d. Allez lire la page concernant la mesure d'entropie d'un mot de passe :
`https://generatepasswords.org/how-to-calculate-entropy/`
Que pensez vous de la méthode « *Diceware* » ?
Version française à `http://weber.fi.eu.org/software/diceware/francais.pdf`

3 – Allez évaluer votre mot de passe contre les attaques « brute-force » sur:
`https://www.security.org/how-secure-is-my-password/`
Essayez le mot de passe donné par la méthode « *Diceware* » `cajous bordes set juge verte`.

4 – Un système utilise des mots de passe sur 5 lettres uniquement.

a. Proposez une méthode *brute force* pour casser les mots de passe utilisés par ce système.

b. Programmez cette méthode en Python et essayez là sur des mots de passe choisis entre vous.

Comment éviter l'affichage d'une saisie à l'écran

Pour réaliser la saisie de votre mot de passe, sans le révéler à ceux qui regardent votre écran :

```
subprocess.run("stty -echo", shell=True)
password=input("Mot de passe :")
subprocess.run("stty echo", shell=True)
```

5 – Allez sur le site `https://p-fb.net/` dans la rubrique correspondant au module et téléchargez le fichier dictionnaire du français « Corpus_des_mots_francais_propre-UTF8.txt ».
Vous disposez maintenant de la liste de tous les mots du français (ou une belle collection).
Vous avez entrevu sur quelles touches votre collègue a frappé pour entrer son mot de passe :

- il a appuyé sur 6 touches en tout, avant de valider son entrée ;
- vous avez pu voir qu'il a appuyé sur les touches suivantes : `s?r???` (les points d'interrogation indiquent que vous n'avez pas pu voir la touche qu'il a pressé).

Questions :

a. en tenant compte du nombre de caractères codés sur un octet, combien de possibilités de mot de passe existe-t-il ?

b. en tenant compte des caractères accessibles en utilisant le clavier ?

c. en utilisant le dictionnaire des mots français ?

6 – Vous avez récupéré une archive ZIP protégée par un mot de passe.
Essayez les mots du dictionnaire comme mot de passe pour retrouver le contenu de l'archive.

Créer un programme en Python pour automatiser la procédure.

```
$ unzip -P password archive.zip
```

Vous pouvez « fusionner » la sortie standard et celle d'erreur avec la commande suivante :

```
commande = subprocess.Popen("unzip -P %s -o archive.zip 2> /dev/st
dout"%motdepasse, shell=True,stdout=subprocess.PIPE)
(resultat, ignorer) = commande.communicate()
# la variable resultat contient la sortie de la commande
```

Vous récupérez l'archive sur le site `https://p-fb.net/`

Fonctions de hachage

7 – Comparaison des algorithmes de hachage :

a. Quelles sont les différences entre *sha1*, *MD5* et *sha-256* ?

b. Quel est celui qui est le plus employé ? Pour quel usage ?

c. Faites une recherche Google avec « `f71dbe52628a3f83a77ab494817525c6` », que trouvez vous ?

d. On détermine la valeur md5 suivante :

```
xterm
pef@samus:~$ echo -n 'https://p-fb.net/' | openssl dgst -md5
(stdin)= b0a433f46a0e73bb341c76b2920d53cf
```

Cherchez cette valeur sur Google ou sur le site `https://md5hashing.net/hash/md5`
Qu'est-ce que cela veut dire ?

8 – Qu'est qu'un MAC (Message Authentification Code) ? Quels sont les avantages ? Comment créer un algorithme de hachage à l'aide d'un chiffrement symétrique ?

9 – Quelles sont les conditions pour que l'on obtienne le même résultat pour le chiffrement en `-aes-ecb` et `-aes-cbc` ?
Vous consulterez au besoin la documentation de la sous-commande `enc` d'`openssl` sur le chiffrement à l'aide de la commande shell `man enc`.

10 – Quelles sont les différences entre les deux procédés suivants :

a. Vous effectuez un chiffrement *RSA* puis un haché *sha-256* ?

b. Vous effectuez un haché *sha-256* puis un chiffrement *RSA* ?

Intégration Empreinte

11 – Faire un programme en Python permettant de stocker l'empreinte d'un fichier dans un autre fichier ayant le même nom suffixé par le nom de la fonction de hachage (par exemple "fichier.SHA256").

12 – Faire un programme en Python permettant d'intégrer un fichier et ses fichiers empreintes (MD5 et SHA256) associées dans une seule archive.

13 – Faire un programme en Python permettant de faire l'inverse : calcul et vérification automatique des empreintes lors de l'extraction du fichier de l'archive.
Conseils : Vous pouvez travailler dans un sous-répertoire pour obtenir la liste des fichiers contenus dans l'archive et les traiter.

Compléments

▷ **Gestion des arguments en Python**

Pour la gestion des arguments sur la ligne de commande du *shell* :

```
xterm
$ python mon_programme toto 1120
```

Ces arguments s'obtiennent en Python par l'utilisation de :

```
import sys
(nom_programme_python, argument1, argument2) = sys.argv
nom_fichier = argument1
valeur = int(argument2) # convertir en entier l'argument donné sous forme de chaîne
```

▷ **Création d'archive zip**

Pour la création d'archive vous pouvez utiliser la commande `zip` :

⋆ pour compresser :

```
xterm
zip mon_archive fichier1_a_archiver fichier2_a_archiver
```

⋆ pour décompresser :

```
xterm
unzip mon_archive.zip
```