Maxime
LE BRAS
SIO1
Bloc 1 - Info/Réseaux

Le 30/01/2024

# TP2 - Les empreintes cryptographiques MD5/SHA1

# Sommaire

# I - Empreinte MD5

1) Manipulations basiques

Questions 1 à 5:

```
administrateur@Debian-12-Bookworm: ~
administrateur@Debian-12-Bookworm:~$ echo bonjour > fichier1
administrateur@Debian-12-Bookworm:~$ md5sum fichier1
94baaad4d1347ec6e15ae35c88ee8bc8  fichier1
administrateur@Debian-12-Bookworm:~$ echo bonjour > fichier2
administrateur@Debian-12-Bookworm:~$ md5sum fichier2
94baaad4d1347ec6e15ae35c88ee8bc8  fichier2
administrateur@Debian-12-Bookworm:~$
```

On remarque que les empreintes cryptographiques MD5 des deux fichiers “fichier1” et “fichier2” sont identiques, car ils ont le même contenu “echo bonjour”.

Question 6:

```
administrateur@Debian-12-Bookworm:~$ echo bonjour1 > fichier3
administrateur@Debian-12-Bookworm:~$ md5sum fichier3
f5acb92e2ac2c8403f8503c552a1d659  fichier3
administrateur@Debian-12-Bookworm:~$
```

Le résultat de la commande “md5sum” pour le fichier est cohérent à mes attentes, car lorsqu’on ajoute un chiffre, un caractère ou qu’on modifie le contenu de la chaîne de caractères “bonjour” alors l’empreinte cryptographique change totalement.

2) Quelques subtilités

```
administrateur@Debian-12-Bookworm:~$ echo bonjour | md5sum
94baaad4d1347ec6e15ae35c88ee8bc8  -
```

Questions 7 à 8:

La commande “echo bonjour | md5sum” calcule le résumé de l’affichage de “bonjour”.
On remarque que sur les deux screens que les deux empreintes cryptographiques sont différentes, cela est dû à la commande “echo bonjour” affiche la chaîne de caractères “bonjour” avec un retour à la ligne alors que le calcul md5 basique de celle-ci, il y a pas de retour à la ligne.

Questions 9 à 10:

```
administrateur@Debian-12-Bookworm:~$ echo -n bonjour | md5sum
f02368945726d5fc2a14eb576f7276c0  -
```

On peut conclure qu’on se retrouve avec la même empreinte cryptographique, car l’option -n de la commande permet de ne pas afficher la nouvelle ligne de fin.

## II - Empreinte SHA1

Questions 11 à 15:

```
administrateur@Debian-12-Bookworm:~$ echo bonjour > fichier4
administrateur@Debian-12-Bookworm:~$ sha1sum fichier4
e7bc546316d2d0ec13a2d3117b13468f5e939f95  fichier4
administrateur@Debian-12-Bookworm:~$ echo bonjour > fichier5
administrateur@Debian-12-Bookworm:~$ sha1sum fichier5
e7bc546316d2d0ec13a2d3117b13468f5e939f95  fichier5
```

Les résumés cryptographiques de fichier 4 et de fichier 5 sont les mêmes. On peut donc dire que l’empreinte SHA1 a le même fonctionnement de hachage que le MD5 sauf que c’est plus sécurisé.

Question 16:

```
administrateur@Debian-12-Bookworm:~$ echo bonjour1 > fichier6
administrateur@Debian-12-Bookworm:~$ sha1sum fichier6
c83904636c6d95cd84e2e298e1d7298e966aed98  fichier6
```

Le résultat de la commande “sha1sum” est cohérent à mes attentes (même justification que pour la question 6).

## III - Comparaison des résumés de “sum”, “md5sum”, “sha1” et “sh512sum”

Question 17:

```
administrateur@Debian-12-Bookworm:~$ sum fichier3
55386     1 fichier3
administrateur@Debian-12-Bookworm:~$ md5sum fichier3
f5acb92e2ac2c8403f8503c552a1d659  fichier3
administrateur@Debian-12-Bookworm:~$ sha1sum fichier3
c83904636c6d95cd84e2e298e1d7298e966aed98  fichier3
administrateur@Debian-12-Bookworm:~$ sha512sum fichier3
b137e593a9bc3632f2d963dd1105e5ccf072b119aaab2b7e7e04e6cd2e806031d8f82
0d207bb7420916a106f1b427508b5d2be605bc723706ea367e9d8c7e780  fichier3
```

On peut voir selon la façon qu’on hache, on voit que l’empreinte cryptographique est différente, plus ou moins longue et variée au niveau des caractères (chiffres, lettres) ce qui va définir la sécurité de celle-ci.

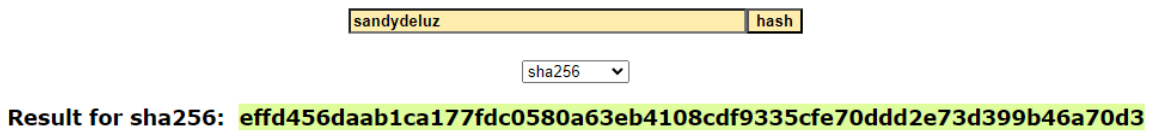
## IV - Vérifier l’intégrité d’un logiciel téléchargé

Question 18-19:

```
administrateur@Debian-12-Bookworm:~$ sha1sum gnufdisk-2.0.0a1.tar.gz.sha1
27c9d87ab989ebf4d481d4b7fe2624d521856dbc  gnufdisk-2.0.0a1.tar.gz.sha1
administrateur@Debian-12-Bookworm:~$ sha1sum gnufdisk-2.0.0a1.tar.gz.sha1.sig
b72983fc08bb2363ae8c92c4ed8bd06219935476  gnufdisk-2.0.0a1.tar.gz.sha1.sig
```

On voit que les deux empreintes cryptographiques sont différentes, donc le logiciel téléchargé n’est pas conforme par rapport au logiciel original.

Question 20:




Ce résumé cryptographique correspond au mot de passe “sandydeluz”.
On utilise ici l’algorithme de hachage sha256.

# V - Conclusion

Pour conclure, ce TP m'a permis de voir l'utilité et le fonctionnement de la cryptographie avec ses différents algorithmes de hachage (md5, sha1, etc…). Elle peut servir à vérifier l'intégrité d'un fichier pour voir s'il est corrompu ou pas.