Introduction à la Sécurité

TP2

Hachage et Signature – Outil GPG :

**BINOMES**

1. **PRENOM NOM :**
2. **PRENOM NOM :**

**Compte de rendu de : Prénom Nom**

IMPORTANT :

* Choisir qui sera Binôme 1 et qui sera Binôme 2
* Le TP est un travail personnel, chaque binôme rendra un rapport personnel de son travail

**Objectifs du TP**

⇒Se familiariser avec les fonctionnalités et la bibliothèque GPG

⇒Savoir utiliser les différents mécanismes de hachage et de signature

**Notez bien** : lorsque les informations qui vous sont fournies sont insuffisantes, faites une recherche dans l’aide qui vous est proposée par Linux avec la commande

« man *gpg* ».

1. I – L’outils GPG

* GNU Privacy Guard (GPG) est un logiciel permettant de chiffrer et/ou signer des documents (fichiers, emails, commits git...), fonctionnant principalement à partir d'un système de paires de clés publique et privée.

**Pour générer de nouvelles paires de clé voir le TP 1**

1. Exemple d’utilisation de gpg pour signer des documents:
   * En utilisant vos clés asymétriques générés lors du TP précédent, vous pouvez signer des messages. Pour cela, il faut utiliser  : *gpg --clearsign <Fichier-à-signer>*  
     Ceci créera un nouveau fichier d’extension .asc qui contiendra le fichier original avec une signature vous authentifiant. Durant la signature, il vous sera demandé la passephrase de votre clé privée
   * Pour générer un fichier de signature séparer du document à signer, ajouter l’option --detach-sign: *gpg* --detach-sign *<Fichier-à-signer>*  
       
     Cela va créer un fichier texte d’extension .sig. devra être envoyé avec le fichier original.

**Pourquoi ?**

* + Pour vérifier un fichier signé, on utilise :  
    *gpg --verify fichier.asc*

*Ou*

gpg --verify fichier.sig  
Il faut bien entendu pour cela disposer de la **clé publique de la personne** qui a signé le document.

* + vous pouvez authentifier une clé publique pour dire « je fais confiance à cette clé publique... » On parle de contre-signature. La commande est simplement : *gpg --sign-key uid*

La commande gpg --fingerprint permet de connaitre le uid de la clé à contresigne.

***Tâche 1 : Signature et vérification***

1. *Créer un dossier* ***signature***
2. *Copier le fichier* ***Prenom\_Nom.txt***
3. *Signer le fichier* ***Prenom\_Nom.txt*** *en utilisant les deux méthodes de signatures.*
4. *Modifier le fichier* ***Prenom\_Nom.txt***
5. *Vérifiez la signature du message qui a été modifié après signature en utilisant le fichier d’extension .asc et le fichier d’extension .sig. Que se passe-t-il ?*
6. *Que pouvez vous en conclure*

***Tâche 2 : Signature et vérification avec votre binome***

1. *Créer un dossier* ***signature2***
2. *Copier le fichier* ***Prenom\_Nom.txt***
3. *Signer le fichier* ***Prenom\_Nom.txt*** *vers un fichier signé d’extension .asc*
4. *Exporter votre clé publique en le nommant en* ***nomdefamille.pub***
5. *Envoyer le fichier signé et votre clé publique à votre Binôme*
6. *Placer le fichier signé et la clé publique reçu de votre Binôme dans le dossier* ***signature2***
7. *Importer la clé publique de votre Binôme*
8. *Vérifier la validité de la signature du fichier de votre binôme. Que remarquez-vous*

***Tâche 3: Contresignature de la clé publique de votre binome***

1. *Contre signer la clé publique de votre binôme que vous venez d’importer à l’étape Tache2*
2. *Vérifier la validité de la signature du fichier de votre binôme. Que remarquez-vous*
3. Exemple d’utilisation de gpg vérifier l’empreinte et la signature d’un fichier télécharger:

***Tâche 4: Vérification d’un iso de debian***

1. *Créer un dossier Debian*
2. *Copier dans le dossier les fichier SHA256SUMS.sign SHA256SUMS.txt*
3. *Vérifier la signature avec la commande****: gpg --verify SHA512SUMS.sign SHA512SUMS.txt***
4. *Quel est le uid de la clé utilisé pour la vérification ? Disposez-vous de la clé publique nécessaire à la vérification de la signature*
5. *Télécharger la clé publique à partir des serveurs de clé de Debian****: gpg --keyserver keyring.debian.org –recv uid*** *. Où uid est l’identité de la clé à telécharger (voir étape* ***d****)*
6. *Revérifier la signature avec la commande de l’étape c)*
7. *Que pouvez dire des informations affichées ?*
8. *Certifier la clé publique téléchargée en le signant. Utiliser pour ce faire la commande : gpg --sign-key uid . Où uid est l’id de la clé publique de debian télécharger à l’étape e)*
9. *Revérifer la signature du fichier avec la commande* ***: gpg --verify SHA512SUMS.sign SHA512SUMS.txt***
10. *Que pouvez-vous dire dire des informations affichées  ?*
11. *Vérifier l’empreinte du fichier avec la commande : sha512sum -c SHA512SUMS 2>/dev/null | grep debian-live-9.4.0-amd64-gnome.iso*
12. *Que pouvez dire des informations affichées ?*