perplexity

Rôle de `sha256sum`

`sha256sum` calcule et vérifie l'empreinte (hash) SHA-256 d'un fichier. Cette empreinte fait 256 bits et s'écrit en 64 caractères hexadécimaux. Elle fonctionne comme une signature numérique du contenu: si un seul bit du fichier change, l'empreinte change complètement.

Principaux usages

1. Vérifier l'intégrité d'un téléchargement
Vous comparez l'empreinte de votre fichier à celle publiée par l'éditeur. C'est l'usage le plus courant, par exemple pour une ISO de distribution Linux.

```
sha256sum debian.iso
```

2. Contrôler automatiquement avec un fichier de sommes

```
sha256sum -c SHA256SUMS
```

Cette commande lit les empreintes listées dans le fichier et affiche `OK` ou `ÉCHEC` pour chaque fichier. Pour ne contrôler que les fichiers présents, ajoutez `--ignore-missing`.

3. Créer vos propres sommes de contrôle

```
sha256sum *.pdf > SHA256SUMS
```

Vous pouvez ensuite détecter plus tard une corruption ou une modification.

4. Détecter une corruption ou une altération
La commande sert à repérer les erreurs de copie, les erreurs de disque et les modifications non autorisées. Elle sert aussi à comparer deux fichiers sans les ouvrir: deux empreintes identiques signifient un contenu identique.

5. Repérer des doublons

```
find . -type f -exec sha256sum {} + | sort | uniq -w64 -d
```

6. Hacher du texte ou un flux

```
echo -n "motdepasse" | sha256sum
```

Limites

- Une empreinte seule ne prouve pas l'authenticité. Si un attaquant peut modifier le fichier et l'empreinte publiée, la vérification ne protège pas. Pour l'authenticité, il faut signer la liste d'empreintes, par exemple avec GPG, ou la récupérer par un canal de confiance (HTTPS).
- Ce n'est pas adapté pour stocker des mots de passe. Utilisez plutôt `argon2`, `bcrypt` ou `scrypt`.

Il existe des commandes équivalentes pour d'autres algorithmes: `md5sum`, `sha1sum`, `sha512sum` (SHA-256 et SHA-512 sont recommandés, MD5 et SHA-1 sont considérés comme obsolètes).