Encoding vs Hashing vs Encryption

Legend:

Input Command

Output of the previous command

EXERCISE 3 - Encryption

Encrypting and Decrypting Files with GPG

Objective

Learn how to encrypt and decrypt files using GnuPG (GPG) on a Linux system.

Prerequisites

- Access to a Linux system
- Install GnuPG installed.
- Basic knowledge of navigating the Linux command line interface.

Task 1

Installation: Ensure that GnuPG is installed on your system. You can install it using the package manager of your Linux distribution. For example, on Ubuntu or Debian, you would use:

sudo apt install gnupg

(optional if needed)

Also, ensure you still have the malware.txt file from Exercise 1. If you deleted it, recreate it.

echo This is evil naughty naughty malware > malware.txt

Task 2

Encrypting a File

To encrypt the file `malware.txt` symmetrically (using a passphrase), run:

gpg --symmetric malware.txt

GPG will prompt you to enter a passphrase.

Choose a strong, memorable passphrase.

Confirm the passphrase when prompted.

After encryption, a new file with the extension .gpg will be created (in this case, malware.txt.gpg). You can see this new file by listing the contents of the directory with ls.

Task 3

Attempt to view the encrypted file's contents using `cat malware.txt.gpg`. You'll see that the output is garbled, indicating the content is encrypted.

malware.txt.gpg

```
�     M d
  W M SE" 8   [ g Xz   &     k p7R q   U3 wGy   H    M%u   `
e   E   M   P ?     F)    p
```

Task 4

Keep in mind if you are decrypting locally, during the encryption process earlier you should have seen:

gpg: directory '/home/ajay/.gnupg' created

gpg: keybox '/home/ajay/.gnupg/pubring.kbx' created

this means there is a pubring, is a local file that is keeping your symmetric keys for you.

When you go to decrypt locally, you won't be prompted to provide the password.

If you provide it to another person, you will need to provide the symmetric private key.

To decrypt the `malware.txt.gpg` file, run:

gpg --decrypt malware.txt.gpg > decrypted.malware.txt

Verifying Decryption

View the contents of the decrypted file using `cat decrypted.malware.txt`. The output should match the original `malware.txt` file, in this case, displaying "This is evil naughty naughty malware".

You have successfully encrypted and decrypted a file using GnuPG. This process ensures the confidentiality of your files, allowing only those with the correct passphrase to access the encrypted content. Remember, the security of encrypted files depends on the strength of the passphrase and the security of the system used for encryption and decryption.

Cleanup (Optional)

- If you wish to clean up the files created during this lab, you can remove them using the `rm` command, e.g., `rm malware.txt.gpg decrypted.malware.txt`.
- Be cautious with `rm`, as it permanently deletes files.