

Encoding v Hashing v Encryption

Exercise 3 - Encryption

The aim of this assignment is to show how GnuPG can be used to encrypt and decrypt files on a Linux System. To achieve the desired aim, the following is needed:

1. A Linux System
2. GnuPG Installation

Task 1 is to install the GnuPG using the following command: `sudo apt install gnupg`. This is illustrated below:

```
tee@demo2: ~  
tee@demo2:~$ sudo apt install gnupg  
[sudo] password for tee:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
gnupg is already the newest version (2.2.27-3ubuntu2.1).  
The following packages were automatically installed and are no longer required:  
  linux-headers-5.15.0-107 linux-headers-5.15.0-107-generic linux-image-5.15.0-107-generic  
  linux-modules-5.15.0-107-generic linux-modules-extra-5.15.0-107-generic  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
tee@demo2:~$
```

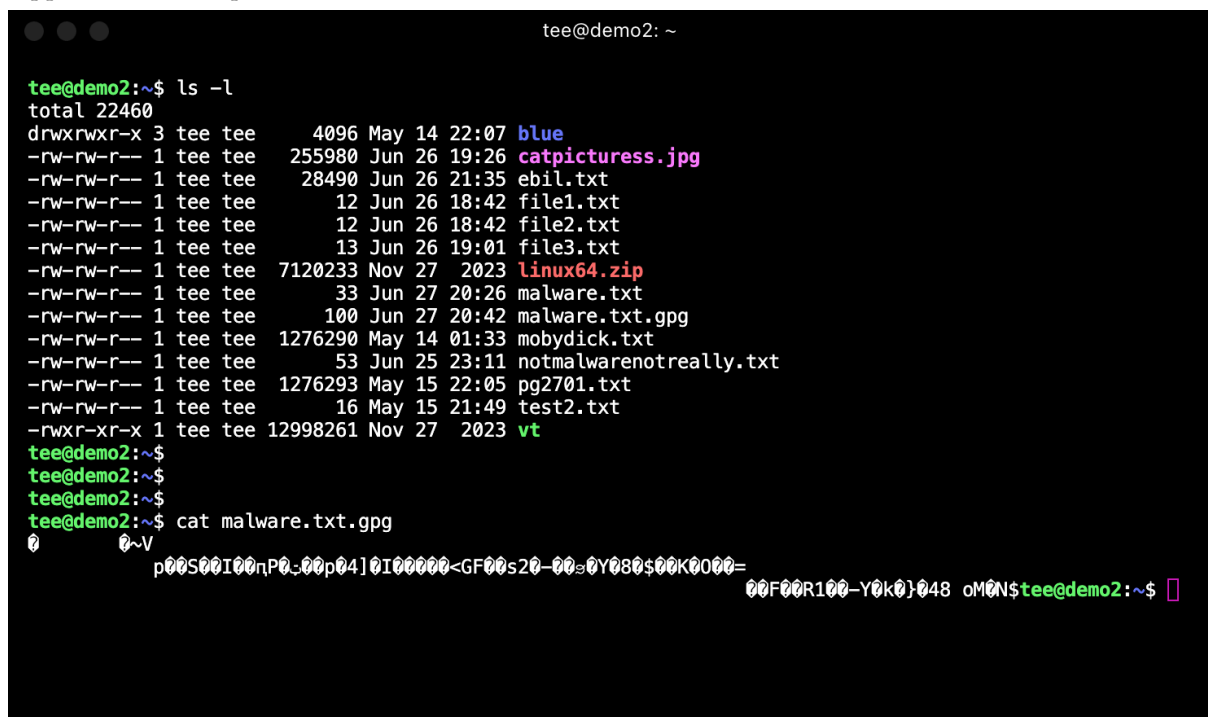
Now that the encryption tool is installed, task 2 is to encrypt a file. In this instance, the file to be encrypted is named 'malware.txt'. To encrypt this file, the following command will be used: `gpg --symmetric malware.txt`. This is illustrated below:

```
tee@demo2: ~  
tee@demo2:~$ gpg --symmetric malware.txt
```

After running this command, the user will be prompted to enter a passphrase as illustrated below:

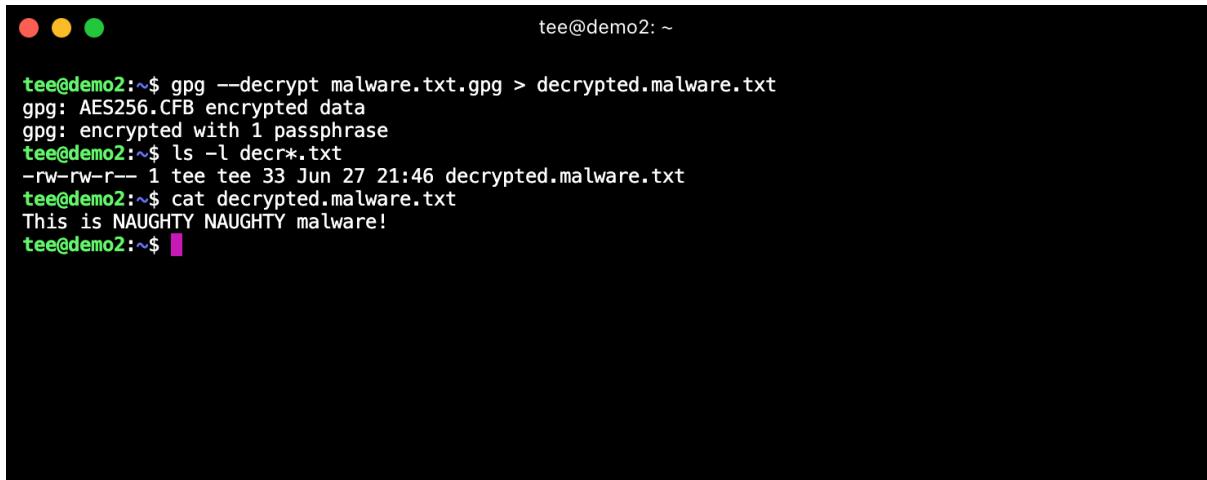


Upon entering of this passphrase, a new file with the extension .gpg will be created. In this case; malware.txt.gpg. The following illustration will show the newly created file as mentioned above and what happens when it is opened:



The output of the malware.txt.gpg file is not in human-readable format. It is now encrypted.

When decrypting locally, there is no requirement to input the pass phrase created earlier as the file is being decrypted locally. There is a local file that stores symmetric keys, hence the absence of the need to enter the pre-created passphrase. However, if the encrypted file is shared to another user, the passphrase will be required to decrypt it. To decrypt the file locally, use the following command: `gpg --decrypt malware.txt.gpg > decrypted.malware.txt`. This command decrypts the `malware.txt` file, and writes it to a new file; `decrypted.malware.txt`. This is illustrated below:

A terminal window with a black background and green text. The window title is 'tee@demo2: ~'. The terminal shows the following commands and output:

```
tee@demo2:~$ gpg --decrypt malware.txt.gpg > decrypted.malware.txt
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
tee@demo2:~$ ls -l decr*.txt
-rw-rw-r-- 1 tee tee 33 Jun 27 21:46 decrypted.malware.txt
tee@demo2:~$ cat decrypted.malware.txt
This is NAUGHTY NAUGHTY malware!
tee@demo2:~$
```

As shown above, viewing the file reveals that its contents have been decrypted. The content of a file has been successfully encrypted and decrypted.