Michael Anthony Borchardt
Information Security: GPG Document Encryption

March 12th, 2024

Objective and Scope

The purpose of this Information Security document is to communicate how to password protect documents using the GPG tool.

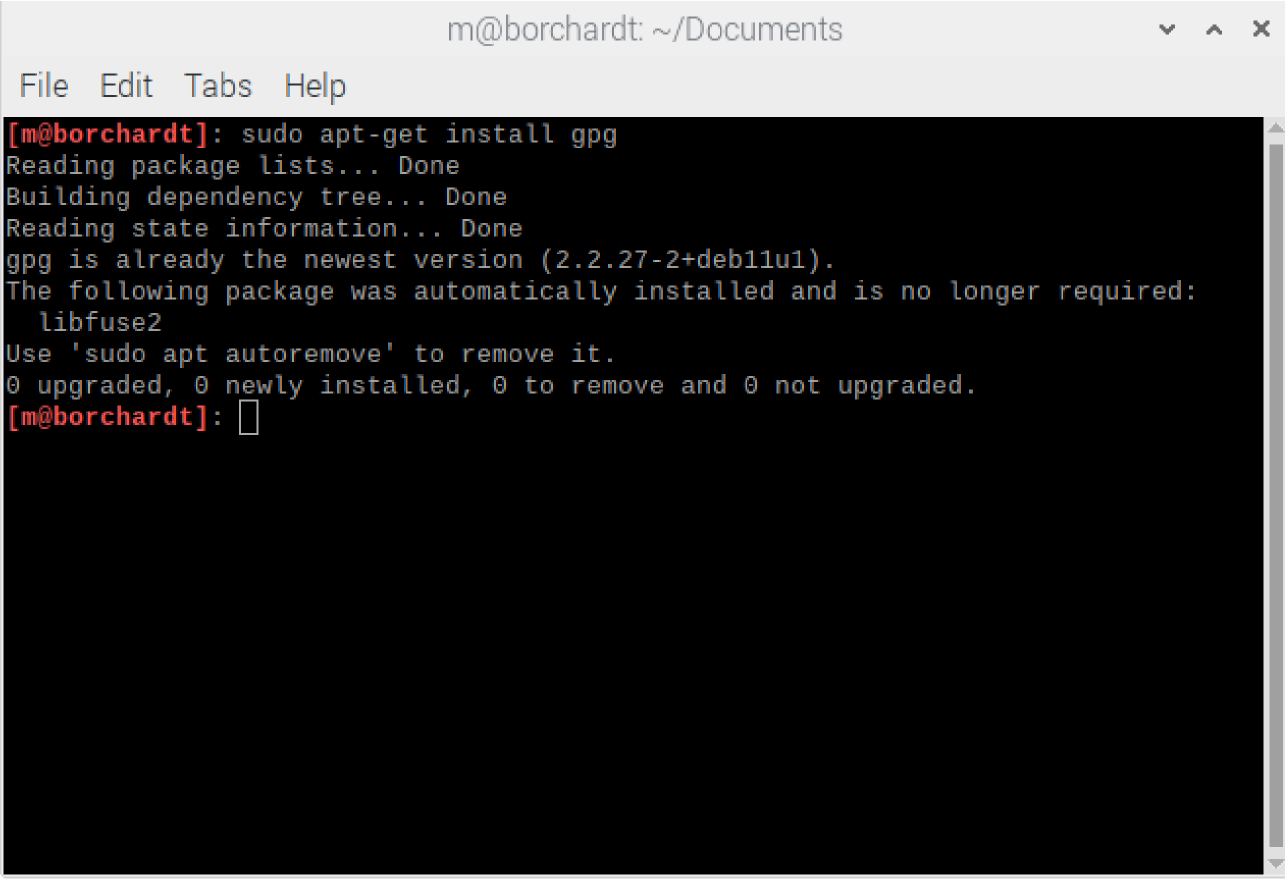
Foremost, this document is intended for use by an experienced Linux user. Also, this document is intended for use by personnel tasked with managing information systems.

1. Introduction

The materials presented within the contents of this document have been collected from several sources across the Internet. These organizations include, but are not limited to, the Open Web Application Security Project (OWASP), the National Institute of Standards and Technology (NIST), the SANS Institute, and other recognized sources of industry best practices.

2. Installing GPG

First, a user would have to install a version of GPG for the operating system they’re running. In my situation, I am running a version of Raspbian, which is a branch of Debian for the Raspberry Pi. Installing software onto this operating system is very similar to using a Debian package manager to install packages. In the following image, I demonstrate how to use a package manager to install GPG:

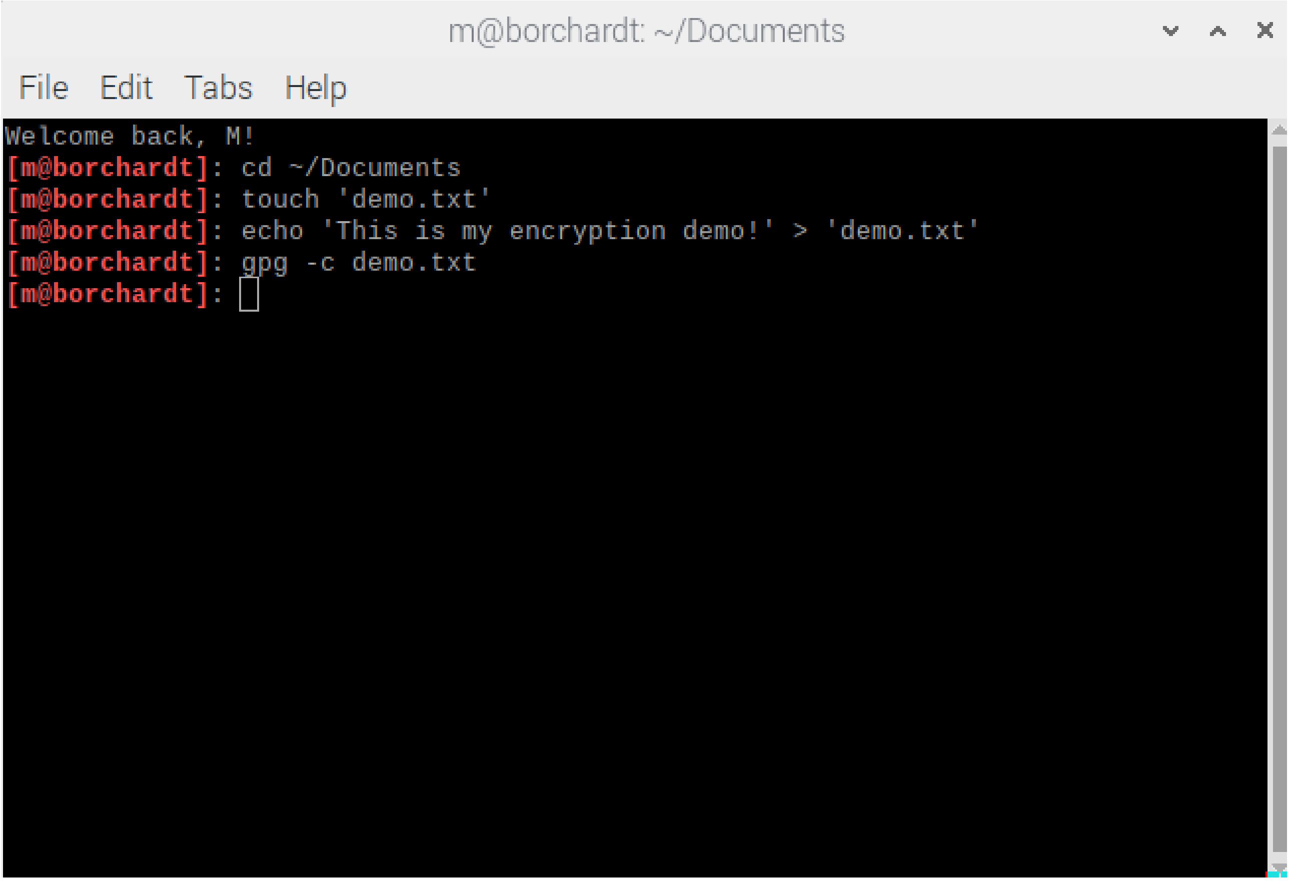


```
[m@borchardt]: sudo apt-get install gpg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gpg is already the newest version (2.2.27-2+deb11u1).
The following package was automatically installed and is no longer required:
  libfuse2
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[m@borchardt]:
```

Figure 1.0: As you can see in the image above, I already have GPG installed. The command used is still the same.

3. Password Protection

Next, we're going to use GPG to password protect a document. There are a few very simple methods of doing this, but I am going to show you my preferred method. In the following image, I demonstrate how to use GPG to password protect a document.



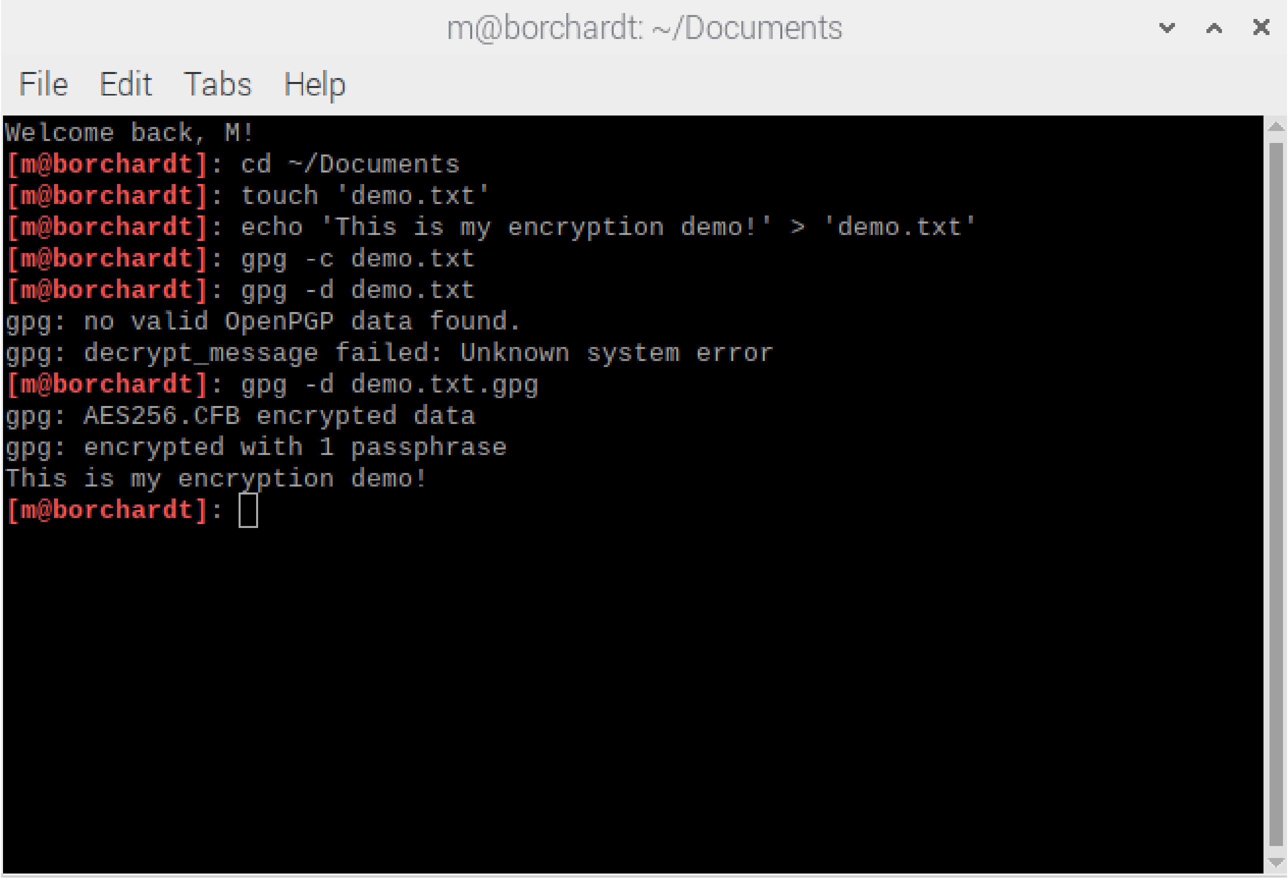
```
Welcome back, M!
[m@borchardt]: cd ~/Documents
[m@borchardt]: touch 'demo.txt'
[m@borchardt]: echo 'This is my encryption demo!' > 'demo.txt'
[m@borchardt]: gpg -c demo.txt
[m@borchardt]: 
```

Figure 1.1: Password protecting a document with GPG.

For the not technical user, what I did in Figure 1.1 is very simple. I used the GPG command with the -c option. Following this, I appended the path to a file I would like to password protect.

What happens next is interesting. GPG creates a copy of the file called "demo.txt.gpg," which is the version of the file encrypted with the password. In order to secure this encrypted version, I would delete the plain-text version of the file. There are multiple ways to do this.

In order to decrypt the file, use the GPG command with the -d parameter. Also, append the path to a file to decrypt in this command, such as in the following example:



```
Welcome back, M!
[m@borchardt]: cd ~/Documents
[m@borchardt]: touch 'demo.txt'
[m@borchardt]: echo 'This is my encryption demo!' > 'demo.txt'
[m@borchardt]: gpg -c demo.txt
[m@borchardt]: gpg -d demo.txt
gpg: no valid OpenPGP data found.
gpg: decrypt_message failed: Unknown system error
[m@borchardt]: gpg -d demo.txt.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
This is my encryption demo!
[m@borchardt]: 
```

Figure 1.2: Decrypting a password encrypted document with GPG. As you can see, you need to decrypt the GPG version of the document, which is the ciphered version. Decrypting a plain-text version of the document will result in an error being thrown.