

**Michael Anthony Borchardt**  
**Information Security: GPG Document Encryption**

March 12<sup>th</sup>, 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.

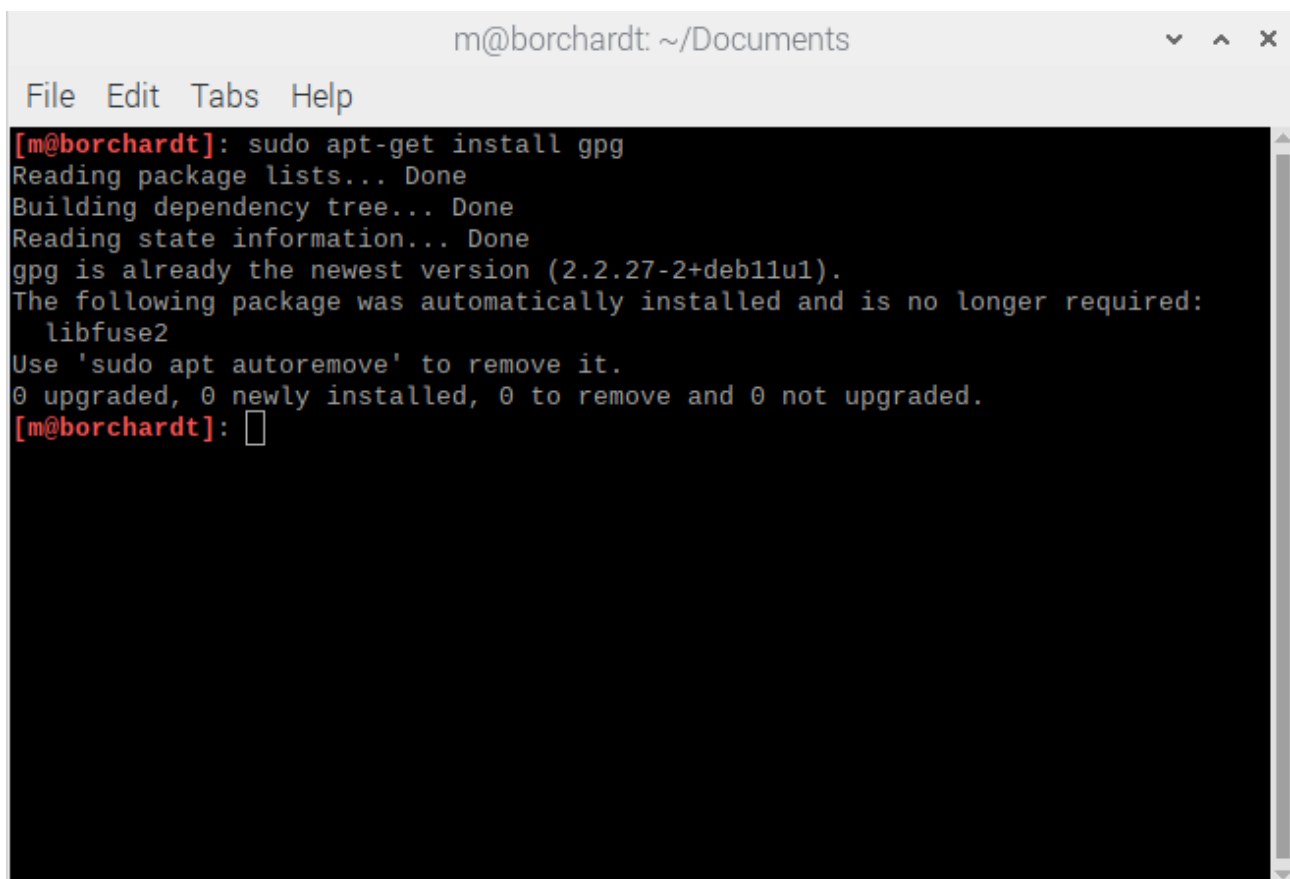
1. Introduction.....	3
2. Installing GPG.....	3
3. Password Protection.....	4

## 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:

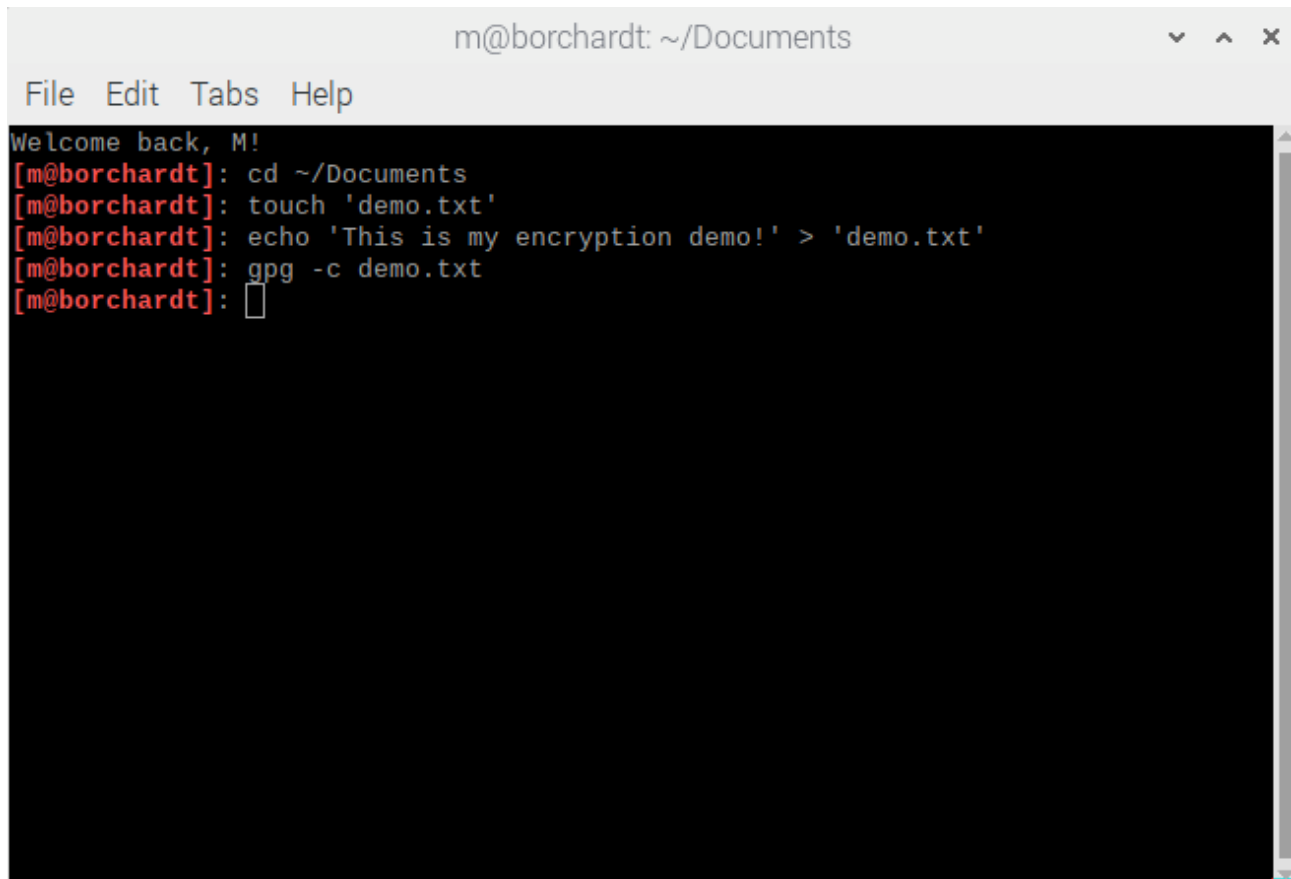
A terminal window titled 'm@borchardt: ~/Documents' with a menu bar (File, Edit, Tabs, Help). The terminal output shows the command 'sudo apt-get install gpg' being executed. The output indicates that gpg is already installed and is the newest version (2.2.27-2+deb11u1). It also lists 'libfuse2' as a package that was automatically installed and is no longer required, suggesting its removal with 'sudo apt autoremove'. The terminal ends with a prompt '[m@borchardt]: ' and a cursor.

```
m@borchardt: ~/Documents
File Edit Tabs Help
[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.

A terminal window titled 'm@borchardt: ~/Documents' with a menu bar 'File Edit Tabs Help'. The terminal output shows a series of commands and their results: 'Welcome back, M!', 'cd ~/Documents', 'touch \'demo.txt\'', 'echo \'This is my encryption demo!\' > \'demo.txt\'', and 'gpg -c demo.txt'. The prompt '[m@borchardt]:' is followed by a cursor after the last command.

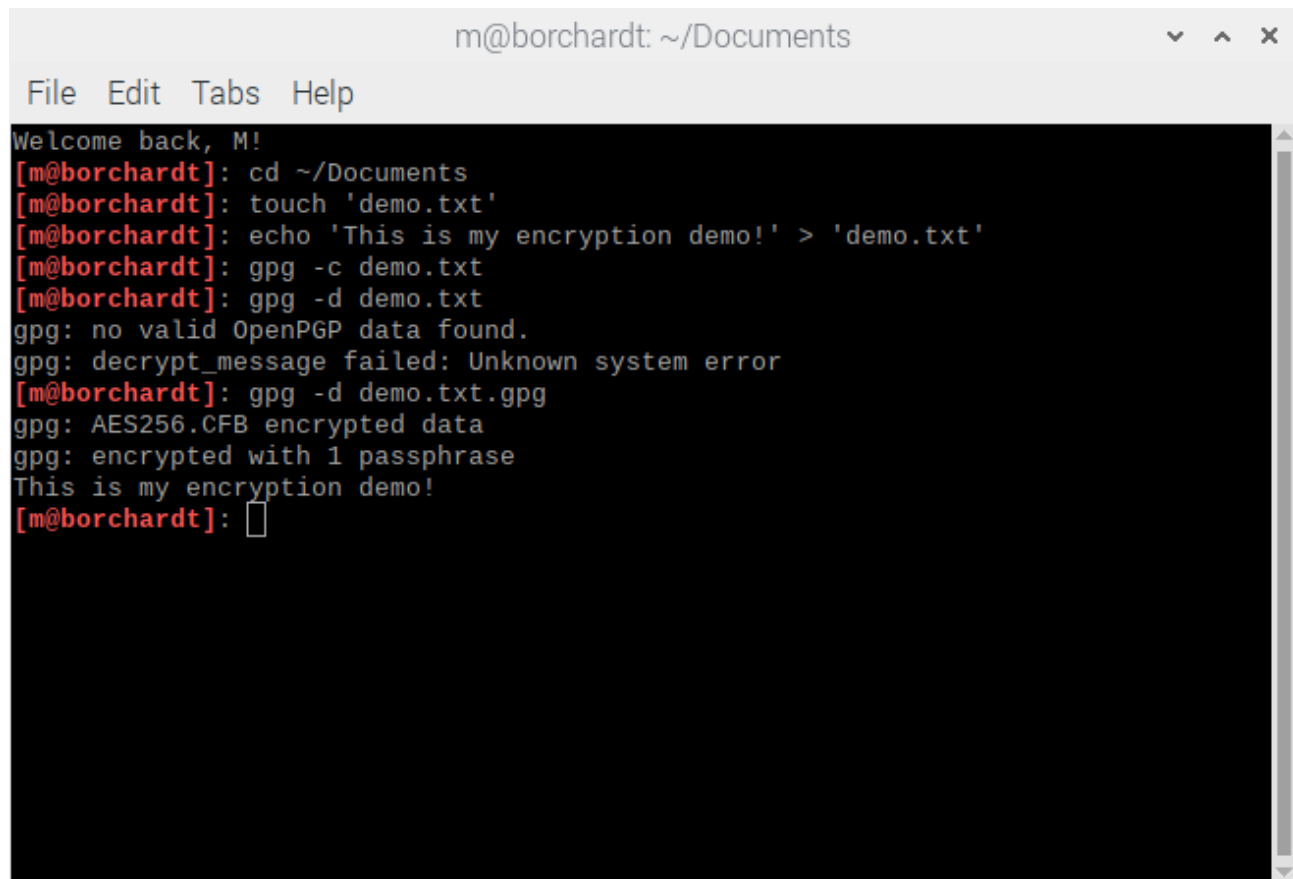
```
m@borchardt: ~/Documents
File Edit Tabs Help
Welcome back, M!
[m@borchardt]: cd ~/Documents
[m@borchardt]: touch 'demo.txt'
[m@borchardt]: echo 'This is my encryption demo!' > 'demo.txt'
[m@borchardt]: gpg -c demo.txt
[m@borchardt]:
```

**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:

A terminal window titled 'm@borchardt: ~/Documents' with a menu bar (File, Edit, Tabs, Help). The terminal shows a sequence of commands and their outputs. The user creates a file 'demo.txt' and encrypts it with GPG. When they attempt to decrypt the original 'demo.txt', an error is shown. When they decrypt the encrypted file 'demo.txt.gpg', the original text is successfully recovered.

```
m@borchardt: ~/Documents
File Edit Tabs Help
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.