**Name: Ayush Srivastava**

**Regno:19BCE1562**

**ISM LAB-11**

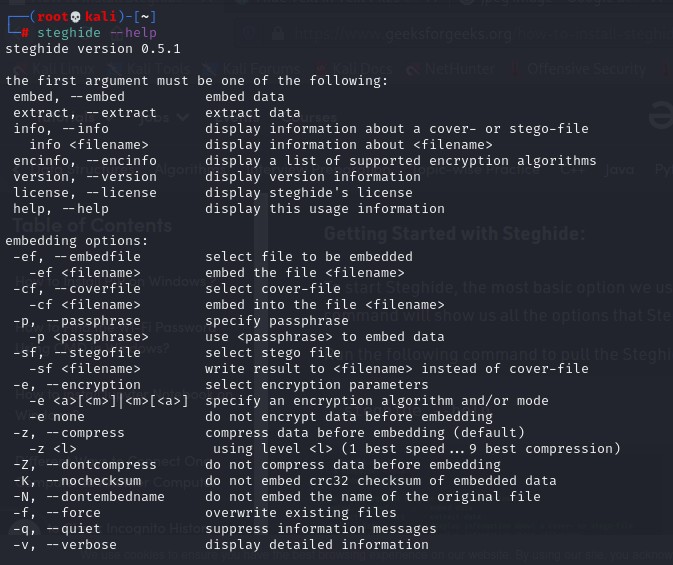
**STEGANOGRAPHY**

Steghide is a steganography tool that allows you to cover confidential records inside a

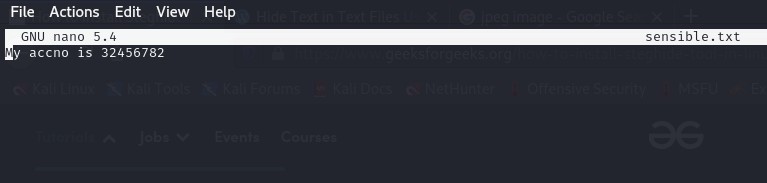
picture or sound record with a passphrase. Bolsters BMP and JPEG picture group, AU and WAV sound group. This device has its advantages and disadvantages. One upside is that it is much better at covering and can extend a lot without any type of document. It does this by using a propelled calculation to shroud it inside a picture (or sound) record without changing the form (or sound) of the document. This is additionally without using Steghide (or if there is not the same scientific method as Steghide) then it is difficult to remove the hidden documents from the picture.

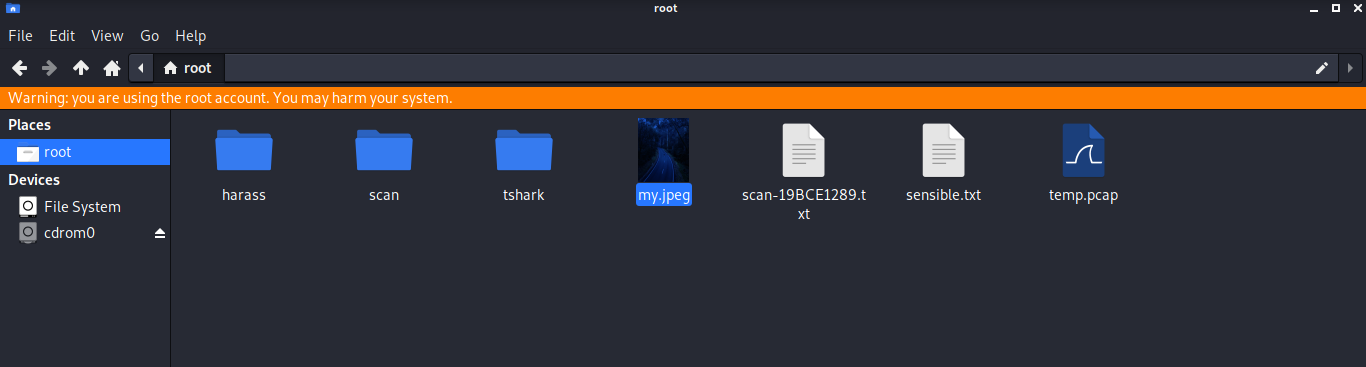
# Steghide Installation

**Verifying installation**

**Steghide help:** command will show us all the options that Steghide offers us.

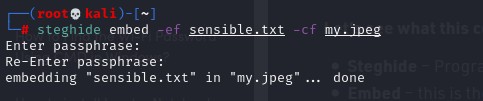
Create a txt file in root folder



Save a jpeg image in root folder

# Embedding data in the image:

We hide the data in the image using the Steghide so that only the person who accepts it can read it. Therefore, we created a text file named “sensible.txt”, in which we wrote our confidential data and images. JPEG is the file in which we are embedding our data.



Here, ef and cf are termed as embedded files and cover files, respectively. Let’s see what this command is doing:

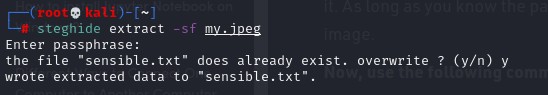
Steghide – Program Name Embed – this is the command

-cf – This flag is for the cover file (the file used to embed the data) filename – this is the name of the cover file

-ef – This flag is for the embed file (the file that will be embedded) Filename – This is the name of the embedded file

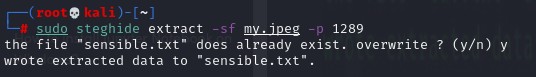
# Extraction of Data From Image Via Steghide:

Using Steghide adds an extra layer of security by allowing us to use a password for it. As long as you know the passphrase, it is quite easy to extract data from the image.



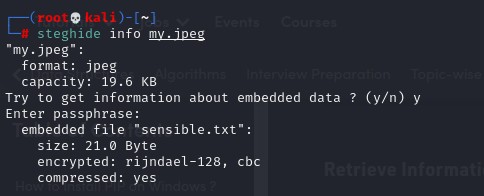
# Password Protect Files:

Now, we can also extract files using the following command. This command is different in that it specifies a password in the command itself, therefore, we do not need to specify it separately.



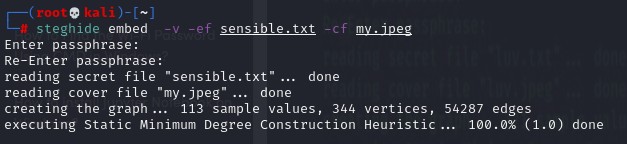
# Retrieve Information of Embedded File:

If we have an image in which the data is suspected to be hidden and if so, what algorithm is used to encrypt the data in the file?



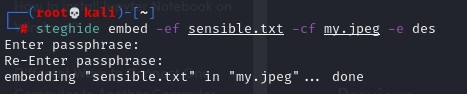
# Verbose Mode

To obtain every information of a file during extraction, we can use verbose mode. The verbose mode gives you detailed information.

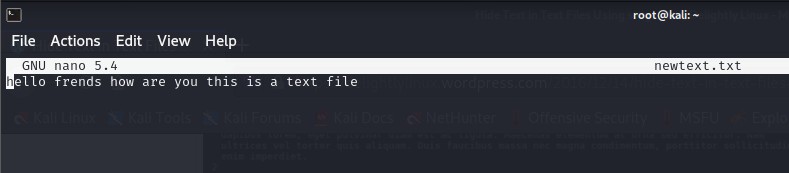


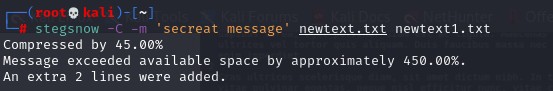
# Encrypting Algorithms:

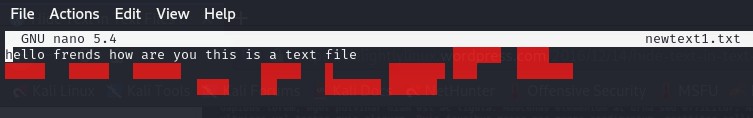
We can encrypt the data we are hiding using encryption techniques.



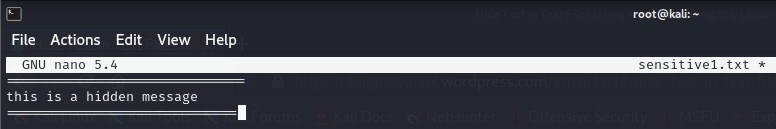
# Hiding text file under text file

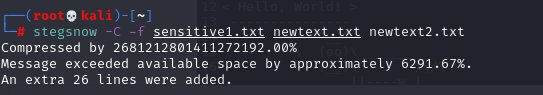
Creating a new text file

This command encodes the message inside newtext.txt and saves the resulting file that contains the message in newtext1.txt.

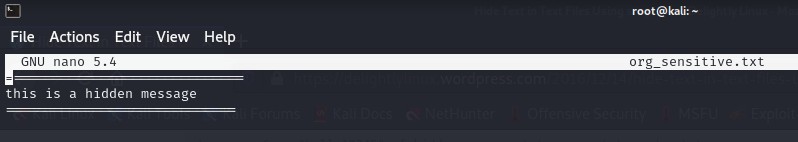
Checking newtext1 file

Checking using stegsnow

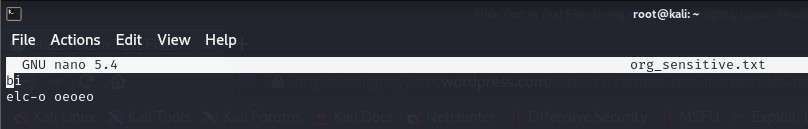
Creating sensitive1 file

Hiding text file under a layer text file

Encode newtext2 file inside org\_sensitive file

Checking org\_sensitive file

Encoding using password for keeping file secure

Checking org\_sensitive file