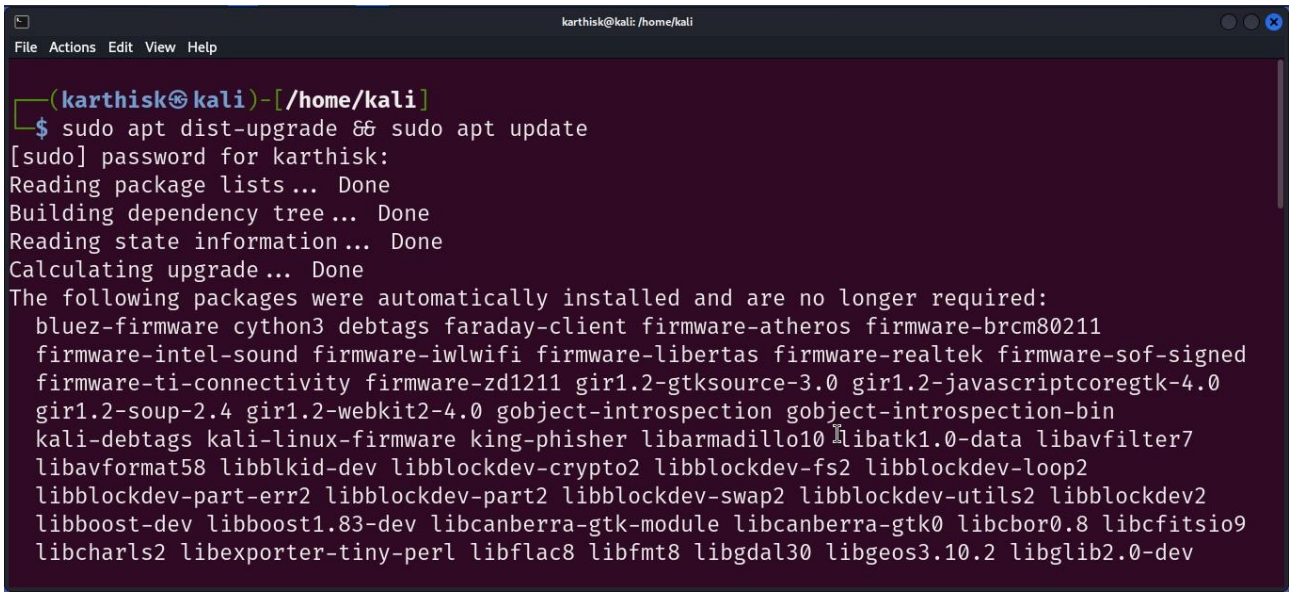
CONVERTING FILE TO STATIC VIDEO.

IMPLEMENTATION

STEP1:

Launch Linux machine. Open terminal and update and upgrade your Linux.

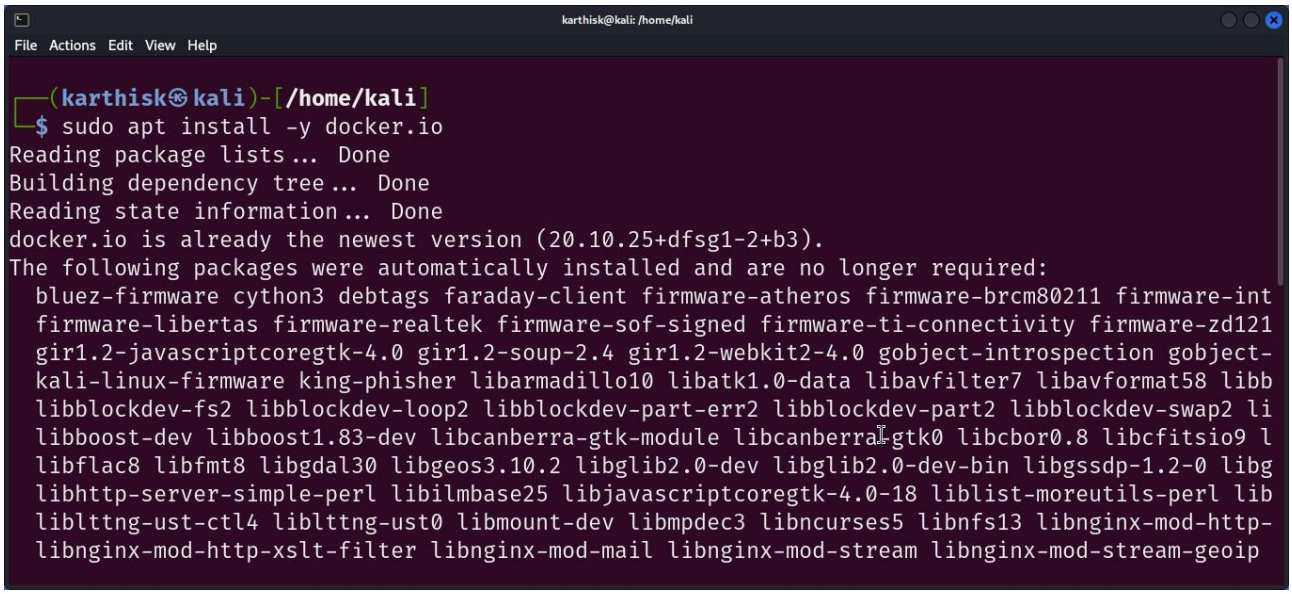
- $ sudo apt dist-upgrade && sudo apt update



STEP 2:

To run this project we need docker. Download Docker.

- $ sudo apt install -y docker.io



STEP 3:

Also need RUST for background process. Download RUST by following command

- $ sudo apt install rustup

```
karthisk@kali: /home/kali
File  Actions  Edit  View  Help

┌──(karthisk㉿kali)-[/home/kali]
└─$ sudo apt install rustup
[sudo] password for karthisk:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
rustup is already the newest version (1.26.0-4).
The following packages were automatically installed and are no longer required:
  bluez-firmware cython3 debtags faraday-client firmware-atheros firmware-brcm80211
  firmware-intel-sound firmware-iwlwifi firmware-libertas firmware-realtek
  firmware-sof-signed firmware-ti-connectivity firmware-zd1211 gir1.2-gtksource-3.0
  gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0 gobject-introspection
  gobject-introspection-bin kali-debtags kali-linux-firmware king-phisher
  libarmadillo10 libatk1.0-data libavfilter7 libavformat58 libblkid-dev
  libblockdev-crypto2 libblockdev-fs2 libblockdev-loop2 libblockdev-part-err2
  libblockdev-part2 libblockdev-swap2 libblockdev-utils2 libblockdev2 libboost-dev
```

STEP 4:

Clone the repository here. And change directory to /Infinite-Storage-Glitch

STEP 5:

In Infinite-Storage-Glitch directory run **sudo docker build -t isg .** to build the Docker Image.

```
karthisk@kali: /home/kali/Infinite-Storage-Glitch
File  Actions  Edit  View  Help

┌──(karthisk㉿kali)-[/home/kali]
└─$ sudo git clone https://github.com/DvorakDwarf/Infinite-Storage-Glitch.git

fatal: destination path 'Infinite-Storage-Glitch' already exists and is not an empty dire
ctory.

┌──(karthisk㉿kali)-[/home/kali]
└─$ cd Infinite-Storage-Glitch/

┌──(karthisk㉿kali)-[/home/kali/Infinite-Storage-Glitch]
└─$ sudo docker build -t isg .
Sending build context to Docker daemon    892MB
Step 1/15 : FROM ubuntu:22.04
 ---> ca2b0f26964c
Step 2/15 : ARG DEBIAN_FRONTEND=noninteractive
 ---> Using cache
 ---> 6c06d2bc9986
```

STEP 6:

Run the command in Infinite-Storage-Glitch directory **sudo docker run -it --rm -v ${PWD}:/home/Infinite-Storage-Glitch isg cargo build --release** to build the project.

```
File  Actions  Edit  View  Help

┌──(karthisk㉿kali)-[/home/kali/Infinite-Storage-Glitch]
└─$ sudo docker run -it --rm -v ${PWD}:/home/Infinite-Storage-Glitch isg cargo build --release
[sudo] password for karthisk:
    Updating crates.io index
  Downloaded autocfg v1.1.0
  Downloaded foreign-types-shared v0.1.1
  Downloaded futures-sink v0.3.25
  Downloaded foreign-types v0.3.2
  Downloaded cfg-if v1.0.0
  Downloaded bitflags v1.3.2
  Downloaded fnv v1.0.7
  Downloaded newline-converter v0.2.2
  Downloaded openssl-probe v0.1.5
  Downloaded futures-task v0.3.25
  Downloaded http-body v0.4.5
  Downloaded hyper-tls v0.5.0
  Downloaded heck v0.4.1
  Downloaded clang-sys v1.4.0
  Downloaded clap_derive v4.1.0
  Downloaded tracing v0.1.37
   Compiling wait-timeout v0.2.0
   Compiling time v0.1.45
   Compiling iana-time-zone v0.1.53
   Compiling dyn-clone v1.0.10
   Compiling unicode-width v0.1.10
   Compiling strsim v0.10.0
   Compiling lazy_static v1.4.0
   Compiling termcolor v1.2.0
   Compiling clap v4.1.6
   Compiling inquire v0.5.3
   Compiling chrono v0.4.23
   Compiling youtube_dl v0.8.0
   Compiling isg_4real v0.1.0 (/home/Infinite-Storage-Glitch)
    Finished release [optimized] target(s) in 4m 03s
```

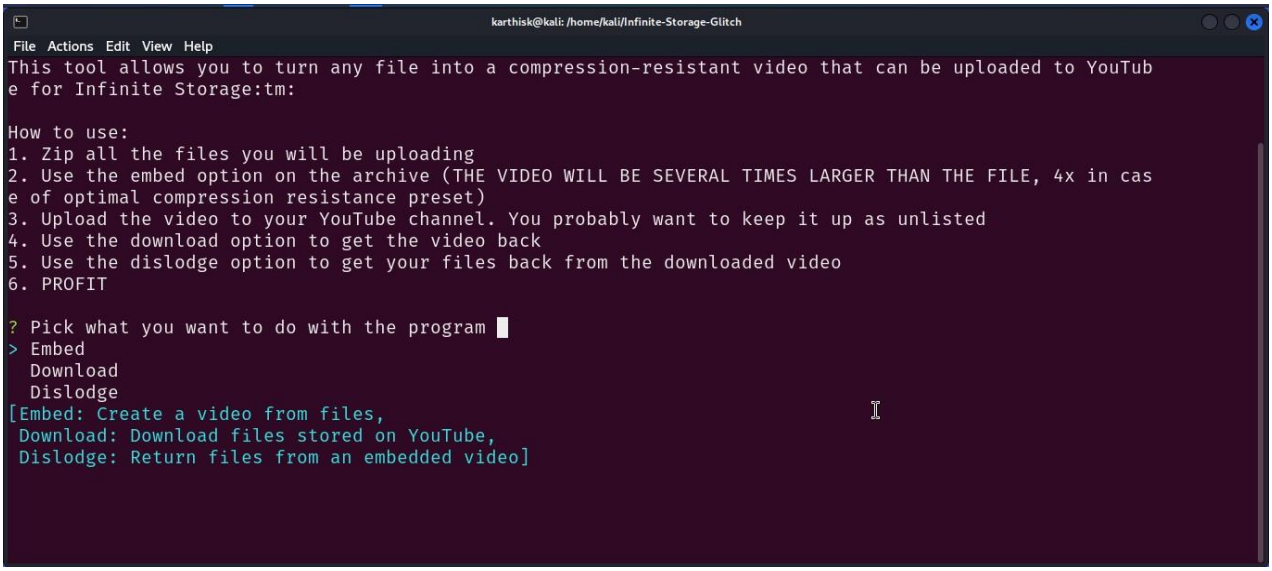
STEP 7:

After executing the command we will find the executable **target/release** directory in Infinite-Storage-Glitch directory. By performing cd **target/.**In target directory cd **release/** in release directory we find **isg_4real** to run the program.

```
karthisk@kali: /home/kali/Infinite-Storage-Glitch/target/release
File  Actions  Edit  View  Help

┌──(karthisk㉿kali)-[/home/kali/Infinite-Storage-Glitch]
└─$ cd target/

┌──(karthisk㉿kali)-[/home/kali/Infinite-Storage-Glitch/target]
└─$ cd release/

┌──(karthisk㉿kali)-[/home/kali/Infinite-Storage-Glitch/target/release]
└─$ ls
build  deps  examples  incremental  isg_4real  isg_4real.d

┌──(karthisk㉿kali)-[/home/kali/Infinite-Storage-Glitch/target/release]
└─$
```

STEP 8:

Run the final command in terminal **sudo docker run -it --rm -v ${PWD}:/home/Infinite-Storage-Glitch isg ./target/release/isg_4real** in Infinite-Storage-Glitch directory to reveal the art of changing any type of file to Static video form.

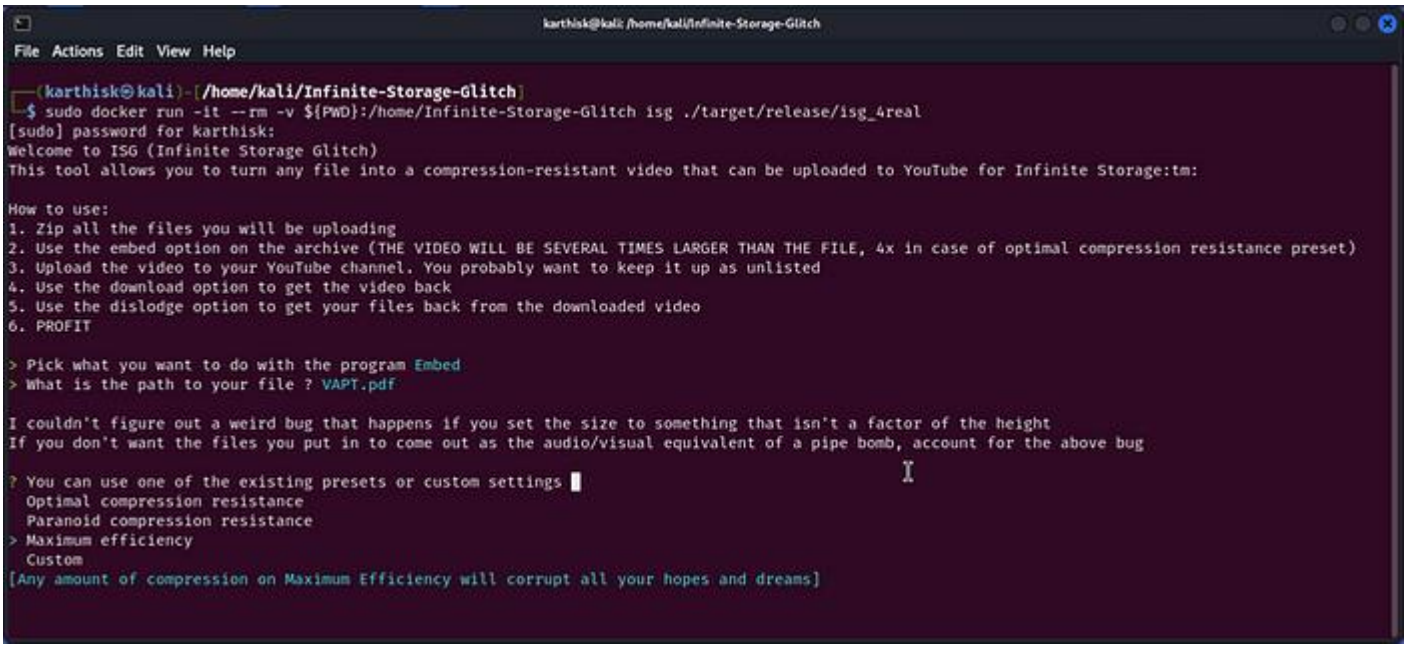


STEP 9:

After that it will display three options:

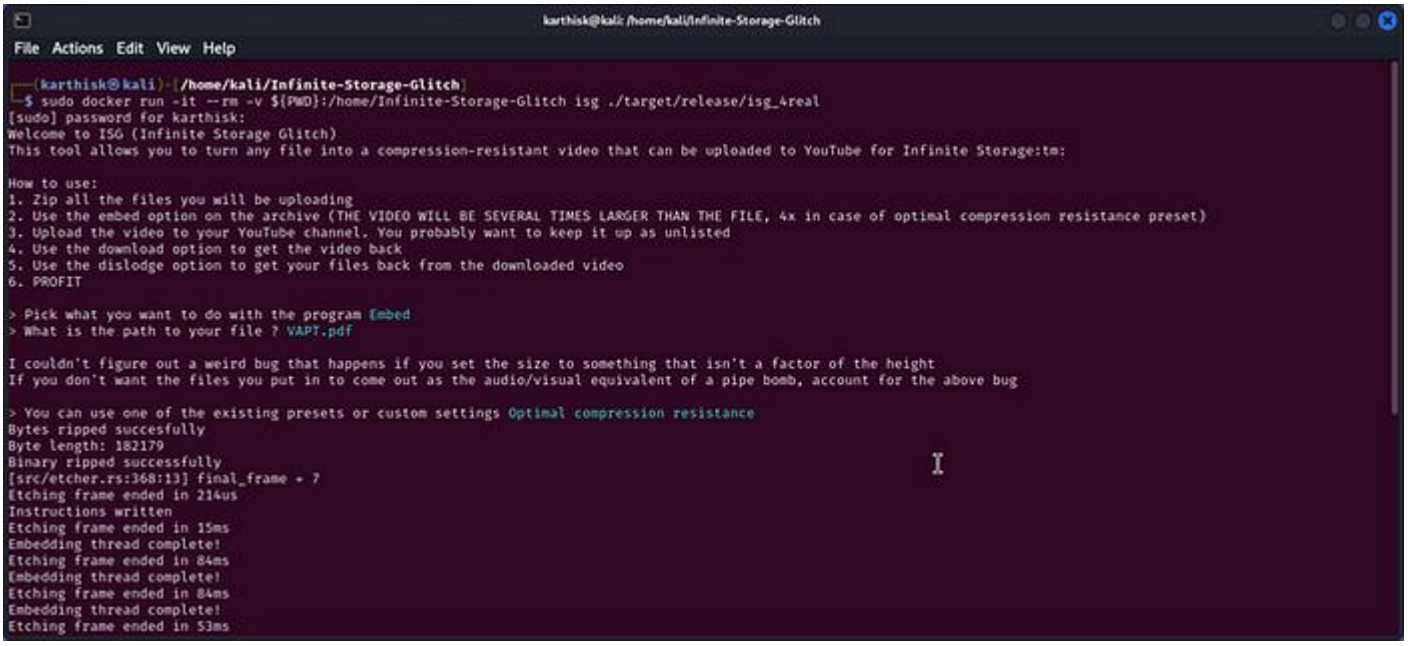
- **Embed:** Create a video from files.
- **Download:** Download files stored on YouTube.
- **Dislodge:** Return files from an embedded video.

Enter the Embed option to Create a video from files and Give the path to your file. Put the pdf file (VAPT.pdf) in the Infinite-Storage-Glitch directory which you can change file into static video form to avoid entering the path.



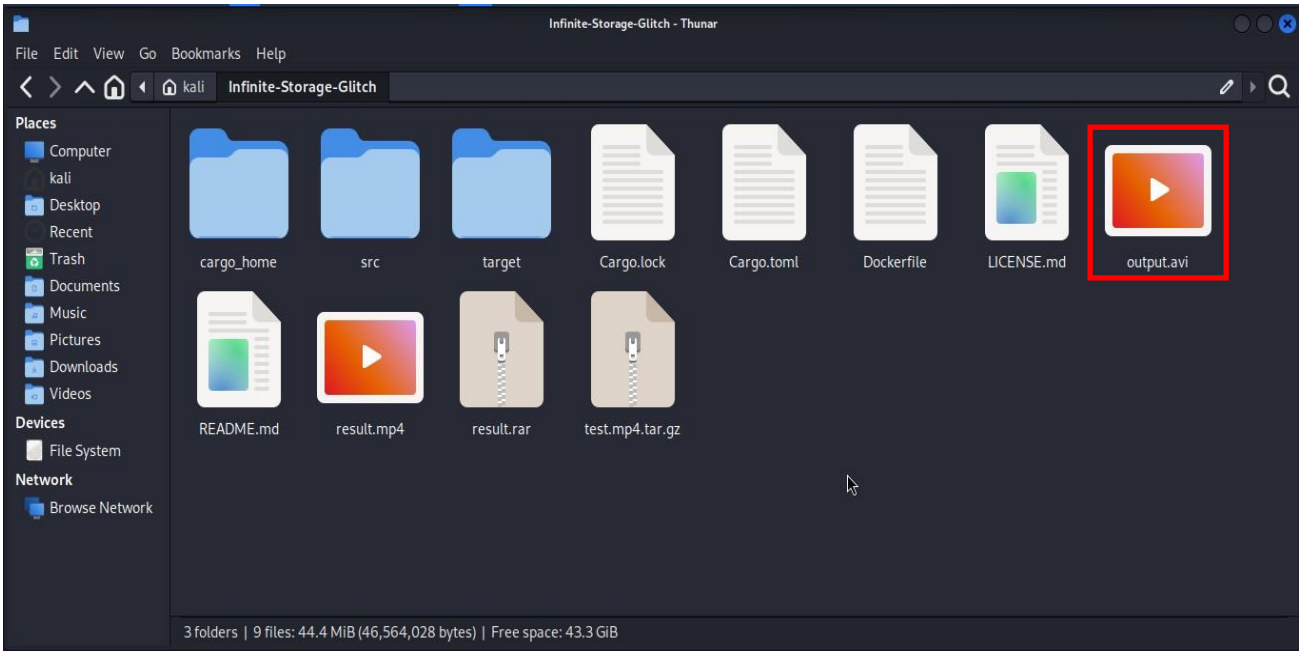
STEP 10:

Once enter the path for the file click enter. Click enter to **Optimal compression resistance** has a default option. Then it will construct Etching frames. Etching frame in the sense converting **binary bits** into **digital frames** and that digital frames convert to **static video** form.



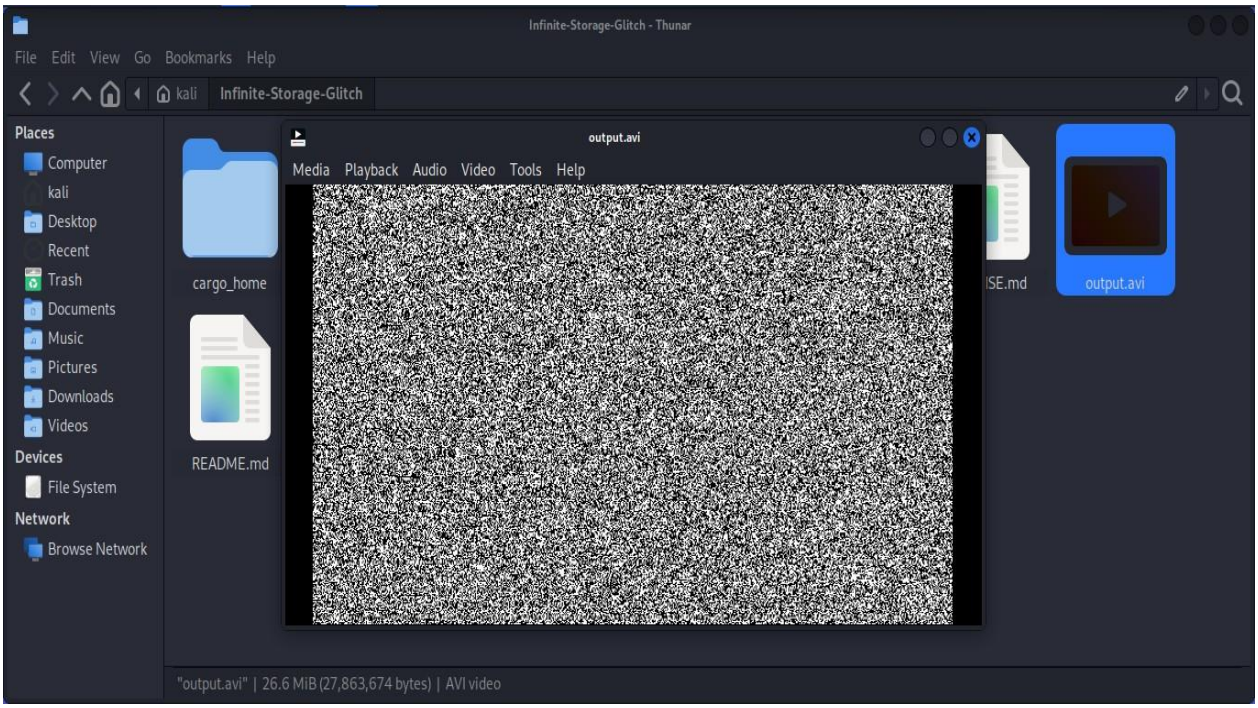
STEP 11:

Then VAPT.pdf file is changed into static video form in the name of **output.avi** which will found in your Infinite-Storage-Glitch folder.



STEP 12:

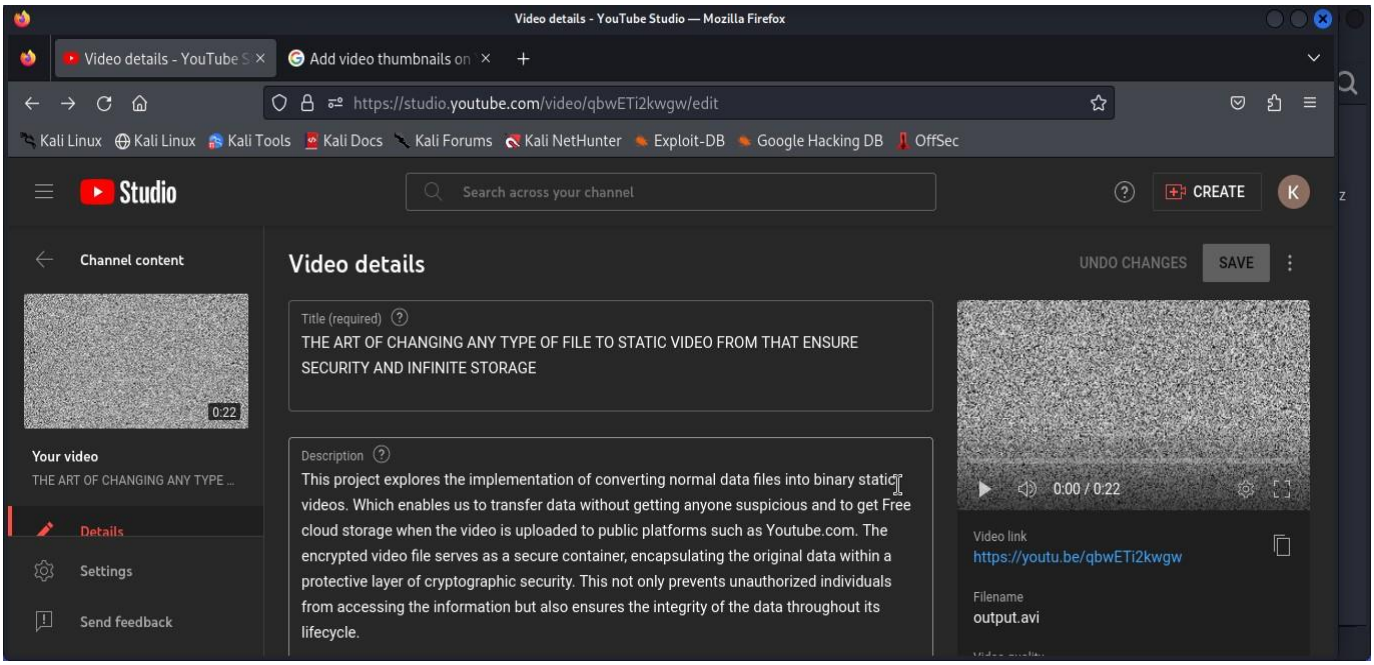
Displaying video of output.avi **STATIC VIDEO.**



Converting various file types — audio, video, text, or documents — into a static video format provides an added layer of security, ensuring that the original content is protected. By employing steganographic techniques, the data can be discreetly embedded within the video frames, making it difficult to detect and extract the concealed information, thereby enhancing confidentiality and data integrity.

STEP 13:

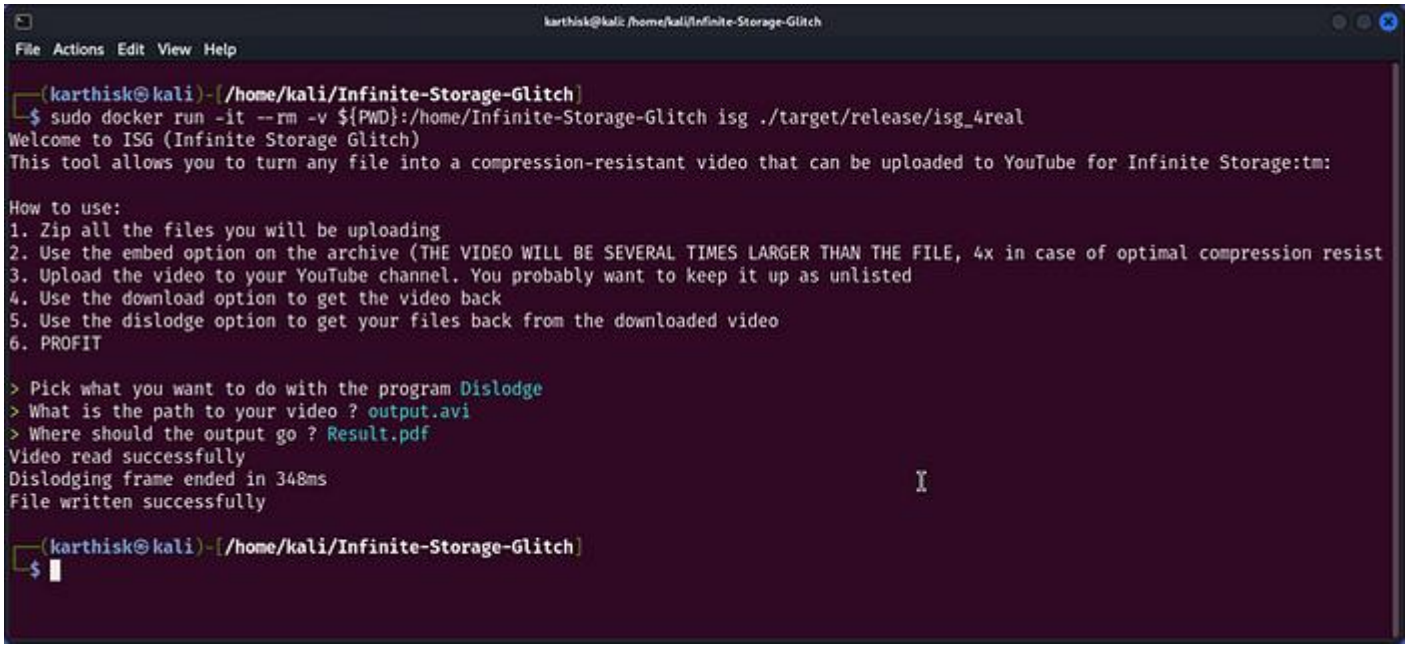
Then upload the **output.avi** static video file to You Tube that act as Infinite Cloud Storage.



LET'S KNOW HOW TO RETRIVE ORIGINAL FORM FILE FROM STATIC VIDEO.

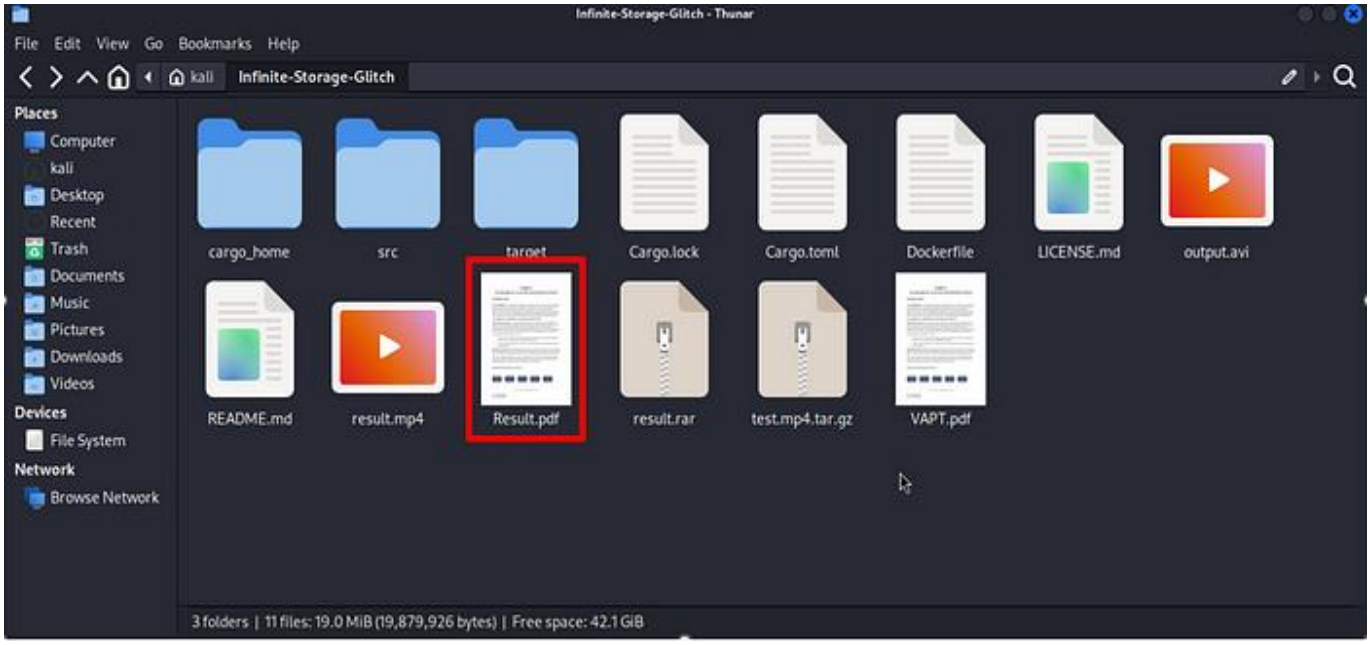
STEP 14:

Once the static video uploaded to YouTube we can access the file remotely. Also we again convert the **static video** to its original format for that give **dislodge** option and Give the path of the static video file. And also include the name of the file and extension which you want to store in your folder.



STEP 16:

After completed above process successfully **Result.pdf** file is stored in yourInfinite-Storage-Glitch folder.



STEP 17:

Displaying video of **Result.mp4** which we have converted static video into its original form using **Dislodge** option.

CONCLUSION:

Using steganographic methods to embed data into a static video file offers a unique approach to secure data storage and transmission. Ensuring that the data remains safe and retrievable in its original form. This method combines the advantages of video data's inherent complexity with the principles of steganography, making it difficult for unauthorized parties to detect and extract the hidden information.