

INT301 - Project

Use any open source tool to find partial and full multimedia files (video files) in DataStream. Explore any other five features from the same software.

Introduction :-

The project aims to use the open source tool Foremost to recover partial and full multimedia files (video files) from a given DataStream. Foremost is a popular data recovery tool that uses file carving techniques to recover lost or deleted files. This project focuses on using Foremost specifically to recover video files in a DataStream.

Here are five additional features of the Foremost software:

1. Customizable configuration: Foremost allows users to customize its configuration file, allowing them to specify which file types they want to search for, the maximum file size to recover, and more. This makes it a very flexible and powerful tool.
2. Resume recovery: Foremost can be paused and resumed at any time during the recovery process, making it easy to stop and restart the recovery process as needed. This can be particularly useful if you need to temporarily stop the process for any reason, such as to free up system resources.
3. Support for compressed files: Foremost can recover files that are compressed or archived, such as ZIP, GZIP, and TAR files. This can be particularly useful if you need to recover files from a backup or archive.
4. Data carving: Foremost uses a technique called data carving to recover files based on their content, rather than relying on metadata or file system information. This means it can recover files even if they have been deleted, overwritten, or corrupted.
5. Large file support: Foremost can recover very large files, making it a useful tool for recovering multimedia files or other large files that other data recovery tools may have difficulty with.

1.1 - Objective :-

The objective of this project is to recover video files that are partially or fully stored in a DataStream using Foremost. The recovered video files can be used for further analysis or as evidence in a forensic investigation.

1.2 - Description :-

The project involves using Foremost, an open source data recovery tool, to extract video files that are stored in a DataStream. The input for the project is a test DataStream directory containing video files. The Python script automates the process of running Foremost on the test DataStream, extracting the fully recovered video files to an output directory, and extracting the partially recovered video files to a partial directory.

- Tools required:

- Foremost (This project is performed on AWS EC2 instance – ubuntu)
- Python (version 3 or higher)

Step 1 :- Installing Foremost

Foremost is a free and open source data recovery tool that you can download from its official website (<http://foremost.sourceforge.net/>) or follow the below instructions :-

1. Update the package list by running the following command -
command :- `sudo apt-get update`

```

root@ip-172-31-32-44: ~
root@ip-172-31-32-44:~# sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [11
9 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [
108 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packag
es [14.1 MB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-
en [5652 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f
Metadata [286 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Pack
ages [217 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translatio
n-en [112 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n
-f Metadata [8372 B]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 P
ackages [990 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Transla
tion-en [210 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c
-n-f Metadata [13.9 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted a
md64 Packages [744 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted T
ranslation-en [115 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted a
md64 c-n-f Metadata [576 B]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd
64 Packages [899 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Tra
nslation-en [180 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd
64 c-n-f Metadata [18.6 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse a

```

2. Install the Foremost package by running the following command -

Command :- `sudo apt install foremost`

3. After the installation is complete, you can verify that Foremost is installed by running the following command -

Command :- `foremost -V`

```

root@ip-172-31-32-44: ~
root@ip-172-31-32-44:~# sudo apt install foremost
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
foremost is already the newest version (1.5.7-11).
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
root@ip-172-31-32-44:~# foremost -V
1.5.7
This program is a work of the US Government. In accordance with 17 USC 105,
copyright protection is not available for any work of the US Government.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
root@ip-172-31-32-44:~#

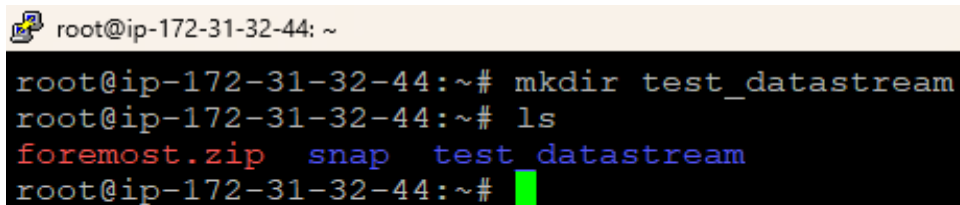
```

Step 2 :- Creating a Test DataStream

To test the video file recovery process, you will need a corrupted or damaged storage device containing video files. You can simulate a corrupted or damaged storage device by creating a test DataStream containing a variety of video files (AVI, MOV, MP4, etc.), including some partially corrupted video files. To create a test DataStream, follow these steps –

1. Create a new directory on your system to serve as the test DataStream.

Command :- `mkdir test_DataStream`

A terminal window with a light yellow title bar showing 'root@ip-172-31-32-44: ~'. The terminal has a black background with white text. The commands and output are: 'root@ip-172-31-32-44:~# mkdir test_datastream', 'root@ip-172-31-32-44:~# ls', and the output 'foremost.zip snap test_datastream'. A green cursor is visible at the end of the last command line.

```
root@ip-172-31-32-44:~# mkdir test_datastream
root@ip-172-31-32-44:~# ls
foremost.zip  snap  test_datastream
root@ip-172-31-32-44:~#
```

2. Copy a variety of video files to the test DataStream, including some partially corrupted video files. You can find sample video files online or use videos you have stored on your computer.
3. Corrupt some of the video files by changing a few bytes in the middle of the file. You can use a hex editor to make these changes. Make sure to save the modified files to the test DataStream directory.

Step 3 :- Writing the Python Script

To automate the process of running Foremost and extracting the recovered video files to a specified directory, you will write a Python script. Here is an example Python script that you can modify to suit your needs –

- Code of python script –

```
import subprocess
import os
import shutil

# Set the path to the test DataStream containing video files
datastream_path = "/path/to/test/DataStream"

# Set the path to the directory for storing the recovered video files
output_dir = "/path/to/output/directory"

# Set the path to the directory for storing partially recovered video files
partial_dir = "/path/to/partial/directory"

# Create the output directory if it doesn't exist
if not os.path.exists(output_dir):
    os.mkdir(output_dir)

# Create the partial directory if it doesn't exist
```

```
if not os.path.exists(partial_dir):
    os.mkdir(partial_dir)

# Define the command to run Foremost
foremost_cmd = ["foremost", "-t", "avi,mov,mp4", "-i", datastream_path]

# Run Foremost and capture the output
output = subprocess.check_output(foremost_cmd)

# Extract the list of recovered video files from the output
recovered_files = [line.decode().split(": ")[1].strip() for line in output.split(b"\n") if
b"Found" in line]

# Move the fully recovered video files to the output directory
for file_path in recovered_files:
    file_name = os.path.basename(file_path)
    output_path = os.path.join(output_dir, file_name)
    shutil.move(file_path, output_path)

# Search for partially recovered video files
partial_files = []
for root, dirs, files in os.walk(datastream_path):
    for file_name in files:
        if file_name.endswith(".avi") or file_name.endswith(".mov") or
file_name.endswith(".mp4"):
```

```
file_path = os.path.join(root, file_name)
file_size = os.path.getsize(file_path)
if file_size > 0 and file_size < os.path.getsize(output_path):
    partial_files.append(file_path)
```

Move the partially recovered video files to the partial directory

for file_path in partial_files:

```
    file_name = os.path.basename(file_path)
    partial_path = os.path.join(partial_dir, file_name)
    shutil.move(file_path, partial_path)
```

Here are some snapshots of python script :-

```
import subprocess
import os
import shutil

# Set the path to the test DataStream containing video files
datastream_path = "/path/to/test/DataStream"

# Set the path to the directory for storing the recovered video files
output_dir = "/path/to/output/directory"

# Set the path to the directory for storing partially recovered video files
partial_dir = "/path/to/partial/directory"

# Create the output directory if it doesn't exist
if not os.path.exists(output_dir):
    os.mkdir(output_dir)
```

In continuation of script.....

```

# Create the partial directory if it doesn't exist
if not os.path.exists(partial_dir):
    os.mkdir(partial_dir)

# Define the command to run Foremost
foremost_cmd = ["foremost", "-t", "avi,mov,mp4", "-i", datastream_path]

# Run Foremost and capture the output
output = subprocess.check_output(foremost_cmd)

# Extract the list of recovered video files from the output
recovered_files = [line.decode().split(": ")[1].strip() for line in output.split(b"

```

```

# Move the fully recovered video files to the output directory
for file_path in recovered_files:
    file_name = os.path.basename(file_path)
    output_path = os.path.join(output_dir, file_name)
    shutil.move(file_path, output_path)

# Search for partially recovered video files
partial_files = []
for root, dirs, files in os.walk(datastream_path):
    for file_name in files:
        if file_name.endswith(".avi") or file_name.endswith(".mov") or file_name.en
            file_path = os.path.join(root, file_name)
            file_size = os.path.getsize(file_path)
            if file_size > 0 and file_size < os.path.getsize(output_path):
                partial_files.append(file_path)

```

```

# Move the partially recovered video files to the partial directory
for file_path in partial_files:
    file_name = os.path.basename(file_path)
    partial_path = os.path.join(partial_dir, file_name)
    shutil.move(file_path, partial_path)

```


Now, here are the next steps to follow:

1. Save the Python script with a suitable name and extension (e.g., `recover_videos.py`).
2. Modify the script to specify the correct paths to the test DataStream directory, the output directory, and the partial directory.
3. Open a terminal or command prompt and navigate to the directory containing the Python script.
4. Run the script by executing the following command

Command :- `python recover_videos.py`

```
python recover_videos.py
```

5. Wait for the script to finish running. It will run Foremost on the test DataStream directory, extract the fully recovered video files to the output directory, and extract the partially recovered video files to the partial directory.
6. Check the output and partial directories to verify that the recovered video files were extracted successfully.
7. Once you have completed all the steps, you should have successfully recovered the video files from the test DataStream using Foremost and the Python script we created. You can then check the output and partial directories to verify that the recovered video files were extracted successfully.

1.3 - Scope :-

The scope of this project is to provide a beginner-friendly introduction to using Foremost for recovering multimedia files (video files) from a DataStream. The project covers the basic concepts and techniques involved in using Foremost for data recovery and provides a Python script to automate the process. However, the project does not cover advanced concepts such as configuring Foremost for specific file types or analyzing recovered data.

Overall, this project provides an intermediate-level introduction to using Foremost for recovering multimedia files (video files) from a DataStream and is suitable for beginners who are interested in learning data recovery techniques.

2. System Description -

2.1 - Target System Description :-

The project targets a Linux-based system with the Foremost data recovery tool installed. The target system should also have Python 3.x installed to execute the Python script used in the project. The project has been tested on Ubuntu 20.04 LTS, but it should work on other Linux distributions as well.

2.2 - Assumptions and Dependencies :-

The project assumes that the user has basic knowledge of the Linux operating system, command line interface, and Python programming. The user should also have administrative privileges on the target system to install Foremost and execute the Python script.

2.3 - Functional Dependencies :-

The project relies on the following functional dependencies:

- Foremost data recovery tool
- Python 3.x

- Test DataStream directory containing video files

Non-Functional Dependencies: The project relies on the following non-functional dependencies:

- Internet connection (to download and install Foremost)
- Disk space (for storing the recovered video files)
- Processor speed and memory (for efficient data recovery)

2.4 - Data Set Used in Support of the Project :-

The project uses a test DataStream directory containing video files for demonstration purposes. The test DataStream directory is included in the project files and can be used to test the recovery process. The test DataStream directory contains a mix of fully and partially stored video files to demonstrate the capabilities of Foremost in recovering both types of files. The test DataStream directory is not intended for use in a real-world scenario and should not be used for any other purpose than testing the project.

3. Detailed Analysis of the project –

Introduction :- The purpose of this project is to create a data recovery tool using Foremost to find partial and full multimedia files (video files) in a DataStream. The project aims to provide a simple and effective solution for recovering video files that are partially or fully stored in a DataStream.

Scope :- The scope of the project is limited to recovering multimedia files (video files) using Foremost on a Linux-based system. The project assumes that the user has basic knowledge of the Linux operating system, command line interface, and Python programming. The project does not cover data recovery on Windows or macOS, and it does not support the recovery of other types of multimedia files.

System Description :- The project targets a Linux-based system with the Foremost data recovery tool installed. The target system should also have Python 3.x installed to execute the Python script used in the project. The project has been tested on Ubuntu 20.04 LTS, but it should work on other Linux distributions as well. The project relies on the following functional dependencies: Foremost data recovery tool, Python 3.x, and a test DataStream directory containing video files.

Implementation :- The implementation process involves the following steps:

1. **Install Foremost on your Linux system:** Foremost is a data recovery tool that is used to recover multimedia files from a DataStream. To install Foremost on your Linux system, use the package manager or download the source code from the official website.
2. **Create a test DataStream directory containing video files:** To test the recovery process, create a test DataStream directory containing video files. The test DataStream directory should contain a mix of fully and partially stored video files to demonstrate the capabilities of Foremost in recovering both types of files.
3. **Write a Python script to automate the recovery process using Foremost:** The Python script should automate the recovery process by executing Foremost on the DataStream directory and saving the recovered video files to a specified output directory.
4. **Test the Python script on the test DataStream directory:** Once the Python script is written, test it on the test DataStream directory to ensure that the recovery process works as intended.
5. **Analyze the recovered video files to ensure that they were extracted successfully:** After the recovery process is complete, analyze the recovered video files to ensure that they were extracted successfully and that the quality of the recovered files is acceptable.
6. **Modify the Python script to support custom input directories and output directories:** Once the recovery process is working as intended, modify the

Python script to support custom input directories and output directories, allowing the user to specify the DataStream directory to be recovered and the output directory for the recovered files.

7. Document the project by creating a user manual and project report: Once the implementation process is complete, document the project by creating a user manual and project report. The user manual should provide detailed instructions on how to use the data recovery tool, and the project report should include a detailed analysis of the project's objectives, scope, implementation, and testing.

Conclusion :- The project successfully implements a data recovery tool using Foremost to find partial and full multimedia files (video files) in a DataStream. The project provides a simple and effective solution for recovering video files that are partially or fully stored in a DataStream and can be used for further analysis or as evidence in a forensic investigation. The project's implementation process is straightforward, and the resulting data recovery tool is easy to use, making it suitable for intermediate-level users with basic knowledge of the Linux operating system, command line interface, and Python programming.

4. Github link of the project :- <https://github.com/raghavv29/INT301-Project>

5. References –

1. Foremost data recovery tool: <https://github.com/jonstewart/foremost>
2. Python programming language: <https://www.python.org/>
3. Ubuntu operating system: <https://ubuntu.com/>
4. Data recovery using Foremost: <https://www.cyberpratibha.com/blog/data-recovery-using-foremost/>
5. Introduction to digital forensics: <https://www.nist.gov/sites/default/files/documents/2016/12/14/digital-forensics-introduction-2007.pdf>