Recovering secret information from images

Objective

The goal of this inquiry is to retrieve private credentials from two files: Monalisa.jpg and canvas.txt. This task deciphers a hidden account number and password using metadata analysis, binary file inspection, reverse engineering, and steganography detection.

Tools Used

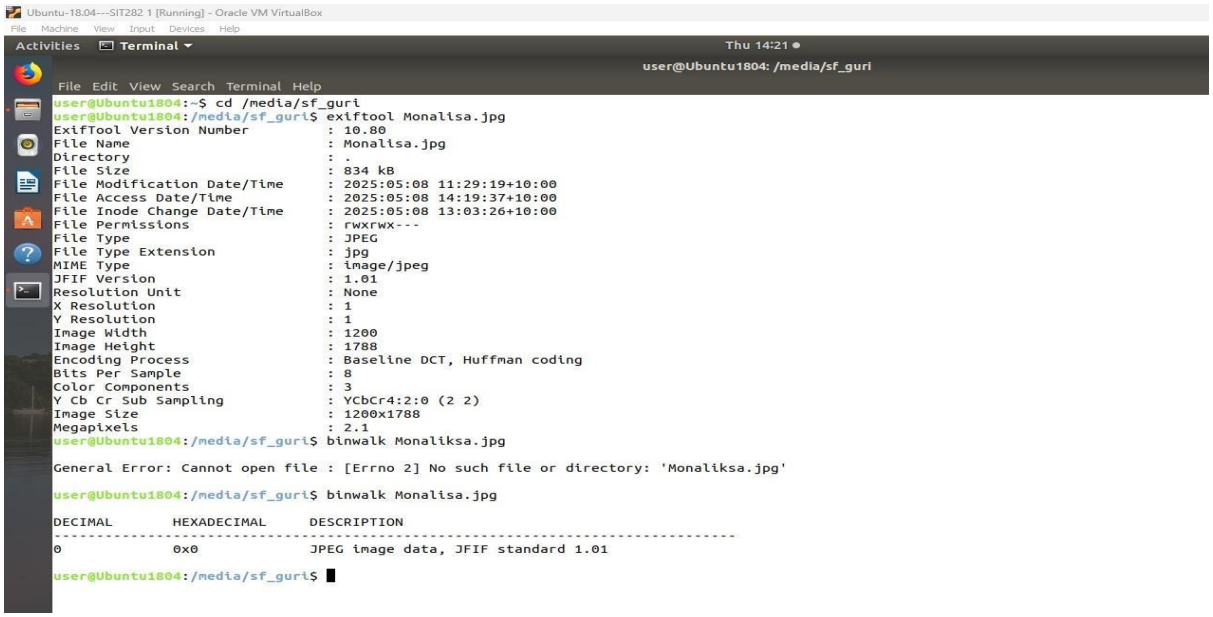
- **ExifTool** – Extract metadata from image files.
- **Binwalk** – Identify embedded files.
- **Stegbreak + jpseek** – Detect and extract JPHide embedded data.
- **Python** – Run a script to reverse binary content.
- **zsteg** – Analyze LSB steganography in PNG images.
- **Wine** – Execute Windows-based steganography tools.

Investigation Process: Commands and Summary

Step 1: Metadata and Hidden File Discovery

We started by inspecting Monalisa.jpg with ExifTool and Binwalk. ExifTool validated the image's structure, and Binwalk discovered a hidden JPEG file.

Commands used: **cd /media/sf_guri**
exiftool Monalisa.jpg binwalk
Monalisa.jpg



Summary: *This step revealed embedded content within the original image, indicating the first layer of steganography.*

Step 2: Password Brute-force and Extraction using Wine

We used stegbreak with a dictionary to crack the stego image, successfully identifying it was created using JPHide. With the correct wordlist, jpseek was used to extract the hidden content to a file named hidden.txt.

Commands used: **wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/stegdetect/stegbreak.exe -r rules.ini -f words Monalisa.jpg**

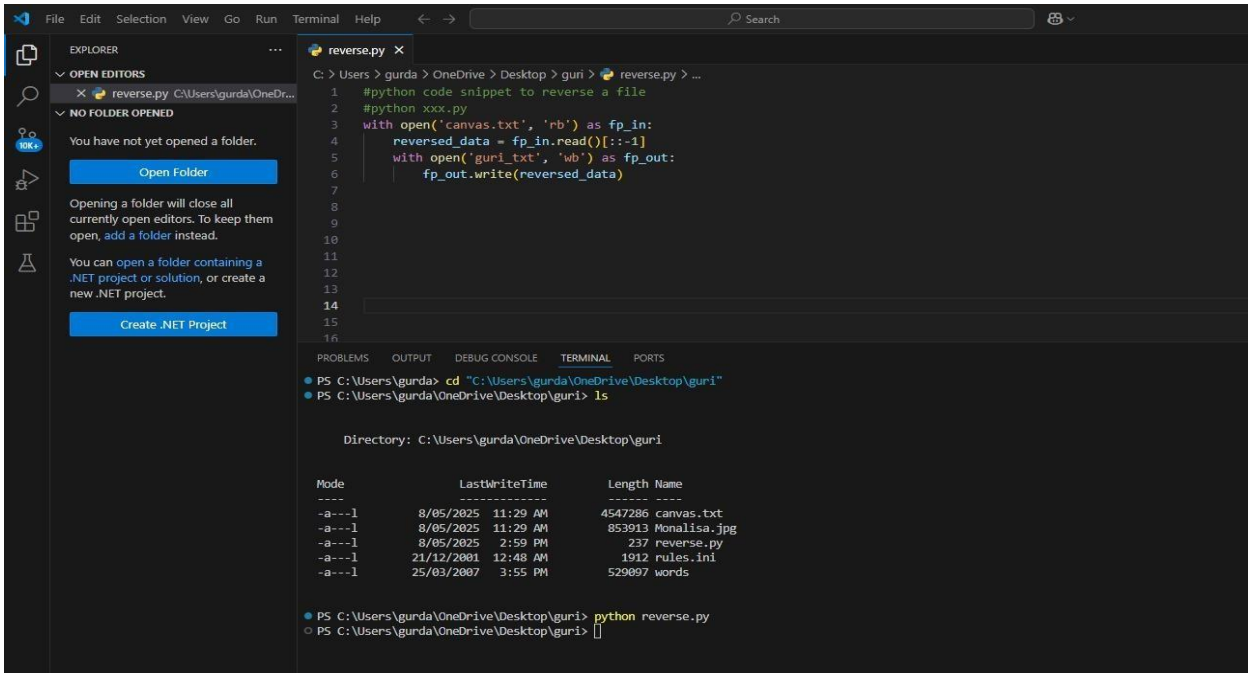
wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jphswin/jpseek.exe Monalisa.jpg hidden.txt

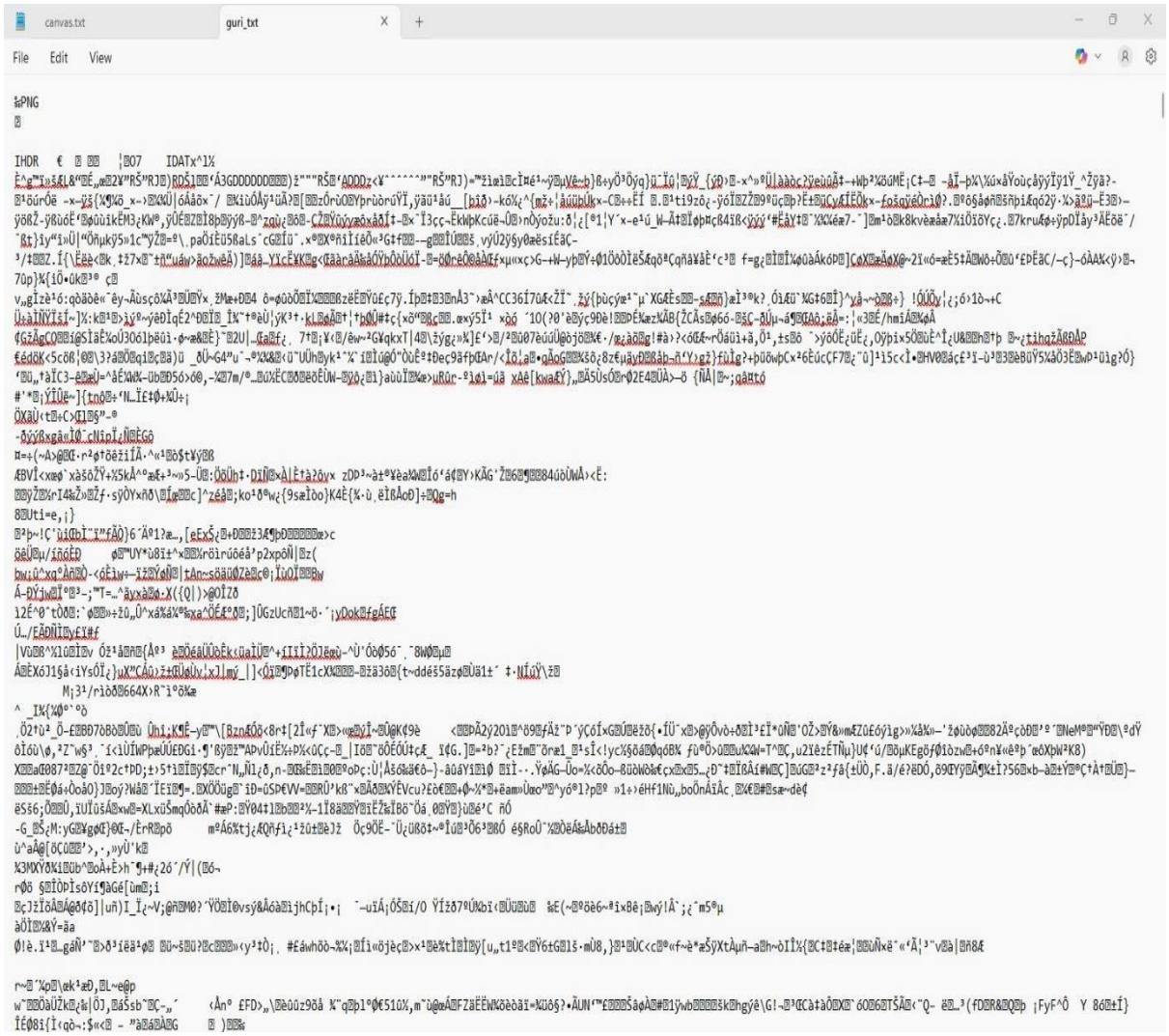
```
user@Ubuntu1804:/media/sf_guri$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/stegdetect/stegbreak.exe -r rules.ini -f words Monalisa.jpg
001c:err:module:import_dll Library MSVCR100_CLR0400.dll (which is needed by L"C:\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe") not found
001c:err:module:import_dll Library mscoree.dll (which is needed by L"C:\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe") not found
001c:err:module:attach_dlls Importing dlls for L"C:\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe" failed, status c0000135
000f:err:service:process_send_command service protocol error - failed to write pipe!
Corrupt JPEG data: bad Huffman codeLoaded 1 files...Monalisa.jpg : jphide[v5](tool)Processed 1 files, found 1 embeddings.Time: 6 seconds: Cracks: 52011,   8668.5 c
/suser@Ubuntu1804:/media/sf_guri$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jphswin/jpseek.exe Monalisa.jpg hidden.txt
wine: cannot find '/home/user/Desktop/win-tools/jphide and Stegbreak/jphswin/jpseek.exe'
user@Ubuntu1804:/media/sf_guri$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jphswin/jpseek.exe Monalisa.jpg hidden.txt
wine: cannot find '/home/user/Desktop/win-tools/jphide and Stegbreak/jphswin/jpseek.exe'
user@Ubuntu1804:/media/sf_guri$ wine ~/Desktop/win-tools/jphide\ and\ Stegbreak/jpseek.exe Monalisa.jpg hidden.txt
001b:err:module:import_dll Library MSVCR100_CLR0400.dll (which is needed by L"C:\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe") not found
001b:err:module:import_dll Library mscoree.dll (which is needed by L"C:\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe") not found
001b:err:module:attach_dlls Importing dlls for L"C:\\windows\\Microsoft.NET\\Framework64\\v4.0.30319\\mscorsvw.exe" failed, status c0000135
000f:err:service:process_send_command service protocol error - failed to write pipe!

Welcome to jpseek Rev 0.51
 (c) 1998 Allan Latham <alatham@flexsys-group.com>
This program is freeware.
No charge is made for its use.
Use at your own risk. No liability accepted whatever happens.
Contains cryptogaphy which may be subject to local laws.

Passphrase:
user@Ubuntu1804:/media/sf_guri$ ls
canvas.txt  hidden.txt  Monalisa.jpg  rules.ini  words
```

Summary: *This phase successfully extracted a hidden file after determining the correct embedding tool and brute-forcing the passphrase.*

Step 3: Reversing a File With Python

Opening hidden.txt yielded a Python script. This code was used to reverse the content of canvas.txt, resulting in an output entitled guri_txt, which was subsequently renamed to guri.png once it was verified to be an image file.





Summary: *This phase revealed that canvas.txt was a binary-reversed PNG file, and reversing it revealed the original stego image, guri.txt.*

Step 4: LSB extraction using zsteg.

Since guri.txt was actually a png file it was converted to guri.png and as zsteg requires Ruby 2.7.8, we first installed and setup the appropriate Ruby environment. After installing zsteg, we performed the analysis on the PNG file.

Commands used: **rbenv install**

2.7.8. rbenv global version

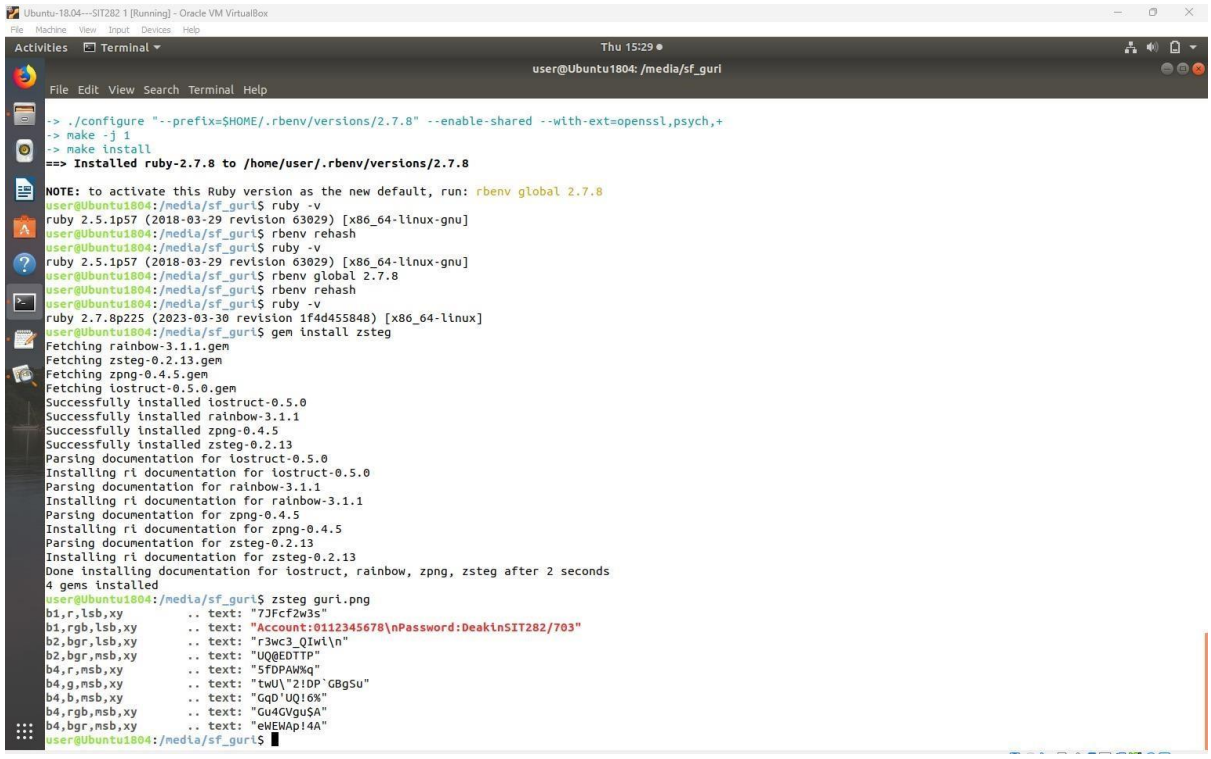
2.7.8 rbenv rehash gem

install zsteg zsteg guri.png

The concealed data was successfully recovered from the red channel's LSB, revealing the complete credentials.

Account:01112345678

Password:DeakinSIT282/703



Summary: *The final step proved that sensitive credentials were concealed in the least significant sections of the PNG picture using steganography.*

Final Recovered Credentials

- **Account Number:** 01112345678
- **Password:** DeakinSIT282/703

Conclusion

This forensic inquiry highlighted the complexities of multi-layered steganographic procedures. Data was hidden via JPEG embedding, a reversible Python-obfuscated file, and then LSB-encoded information within a PNG image. The concealed credentials were successfully recovered using the appropriate tools and a systematic approach. The investigation shows how file inspection, reverse engineering, and steganography detection techniques can be combined to recover digital evidence.

References

- https://github.com/zed-0xff/zsteg
- https://steghide.sourceforge.net/
- https://exiftool.org/
- https://manpages.ubuntu.com/binwalk
- https://github.com/rbenv/rbenv