WINTER OF CODE 8.0

UNISTENO: THE UNIVERSAL STEGANOGRAPHY TOOLKIT

Adhiraj Singh | 25JE0608

Introduction

Steganography is the practice of hiding information (payload) into a piece of digital media (like images, videos, audios, etc.)
Steganography is useful in areas like cybersecurity, digital forensics, CTF challenges, etc.
Unisteno is a unified and extensible open source program capable of analysing, embedding, and extracting hidden data across multiple media formats.
Unisteno is a developer-friendly framework for extending steganography methods.

Features of Unisteno

1. automatically detects file types and applies relevant steganalysis
2. Supports Analyze, Embed, and Extract workflows for
    a. Images
    b. Text Files
    c. PDFs
    d. Audio Files
    e. Videos
3. Provides a visual and statistical analysis
4. Uses a plugin-based architecture for extensibility
5. Offers an intuitive web-based GUI for both beginners and advanced users.
6. MIME type is detected using python-magic
7. File heuristics are applied (binary/text inspection) and the backend routes the file to only relevant plugins

System Architecture

- Frontend:
    - HTML5, CSS3, Bootstrap
    - Javascript for dynamic rendering
    - Drag-and-Drop upload and visualisation
- Backend:
    - Python (Flask)
    - Handles uploads, analysis, embedding and extraction.
    - Dynamically loads plugins
- Plugin System
    - Each steganographic method is isolated as a plugin.
    - Plugins implement:
        - can_handle(mime,path)
        - analyze()
        - embed()
        - extract()
    - The Separation makes scalability more possible
    - A corporation can code custom plugins for the file types they are going to use, making industrial applications possible.

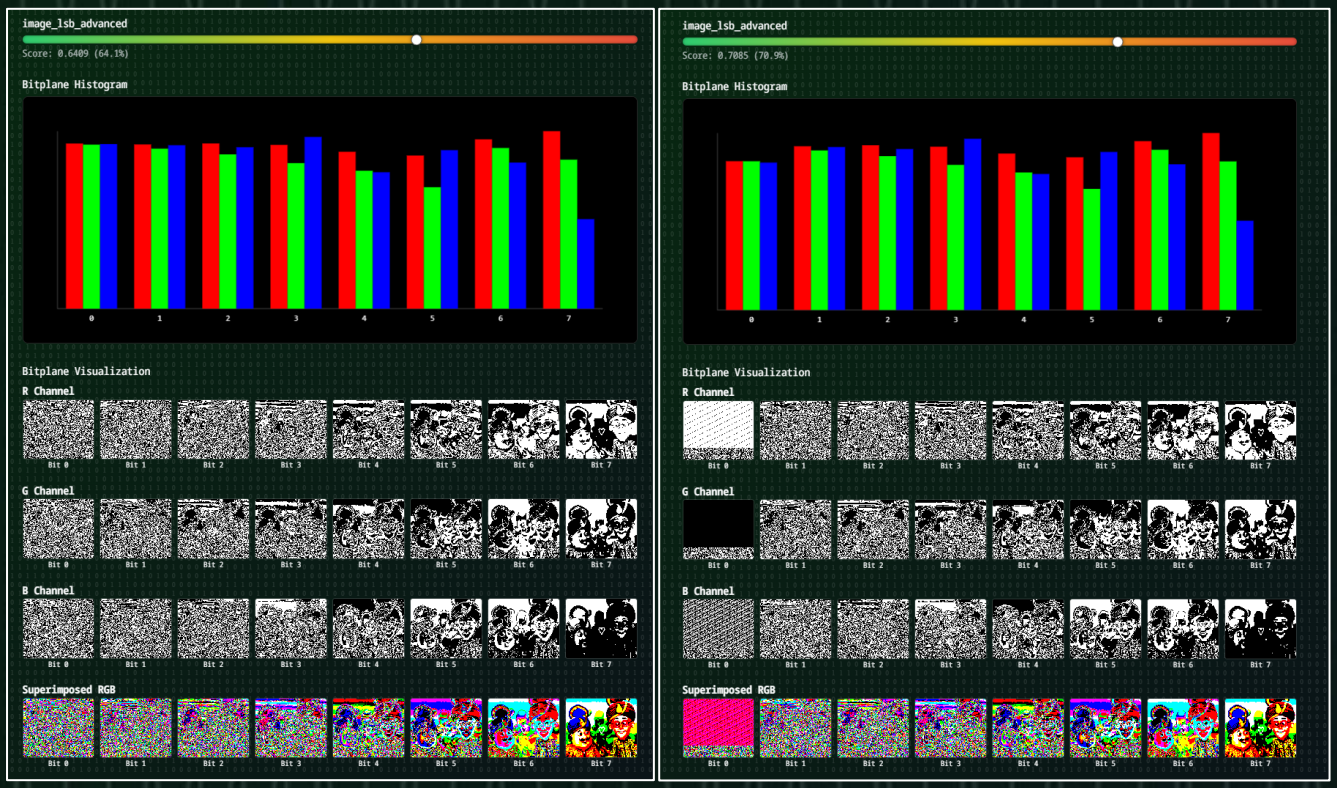
Images

Analyze

These are two images that look identical, but one of them is actually hiding a massive amount of data

To find out which, we can upload both to Unisteno.
Unisteno performs:
LSB Bitplane Extraction
Bitplane Histogram Analysis
Chi-Square Statistical Test
Metadata Inspection

These tools help identify both obvious and subtle hidden data.
We define the score as function of chi
(which we get from chi-square statistical test)
The score for one image is 64.1% and the other one is 70.9%
We can also see the patterned hidden in the least significant bit (bit 0) in the second file. (please zoom in)



Embed & Extract

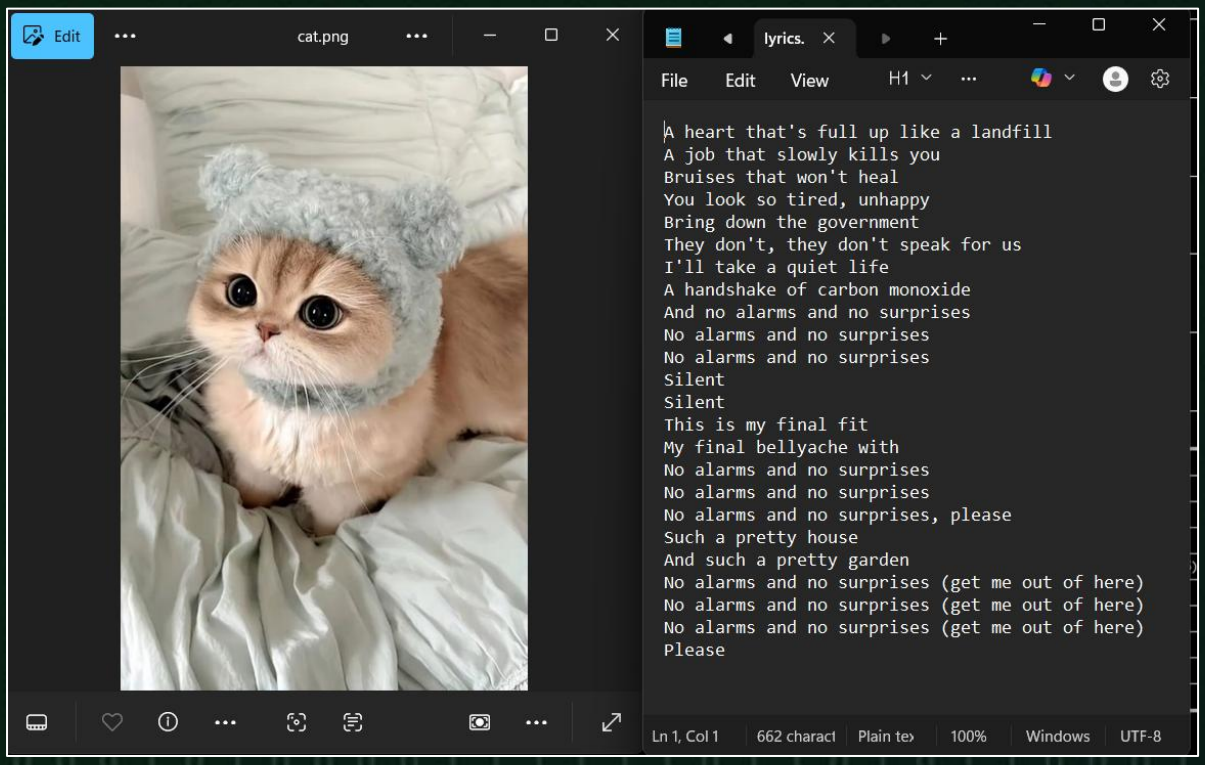
Unisteno implements password-seeded PRNG to randomise LSB embedding positions, which prevents sequential embedding detection (Like how we did in the Analyse example, that was intentionally bad embedding)

The embedder embeds: payload size, payload checksum, encrypted payload.

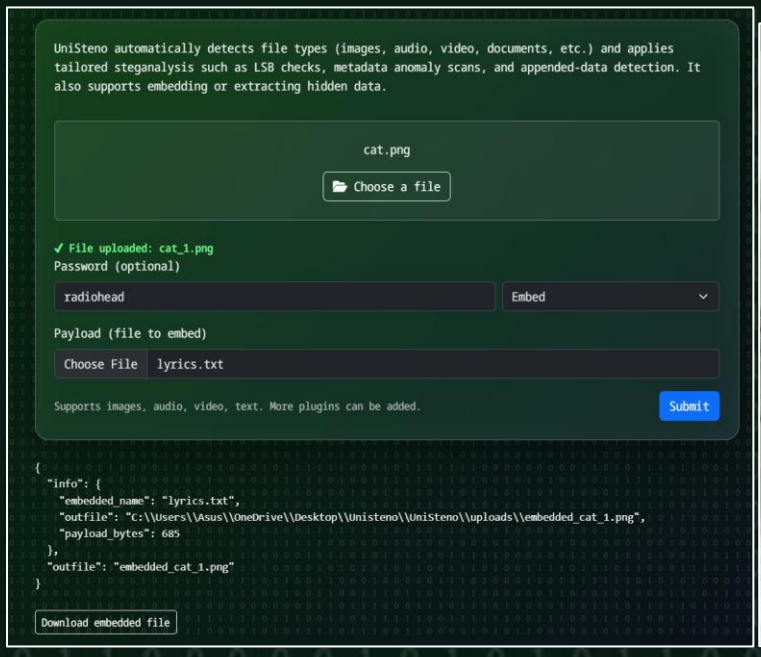
During extraction,

PRNG is re-seeded using the password, embedding bits are reconstructed, checksum is verified, and payload is safely recovered.

*UniSteno embedded file can only be extracted by UniSteno.



(This is the original file and payload to be embedded in it)



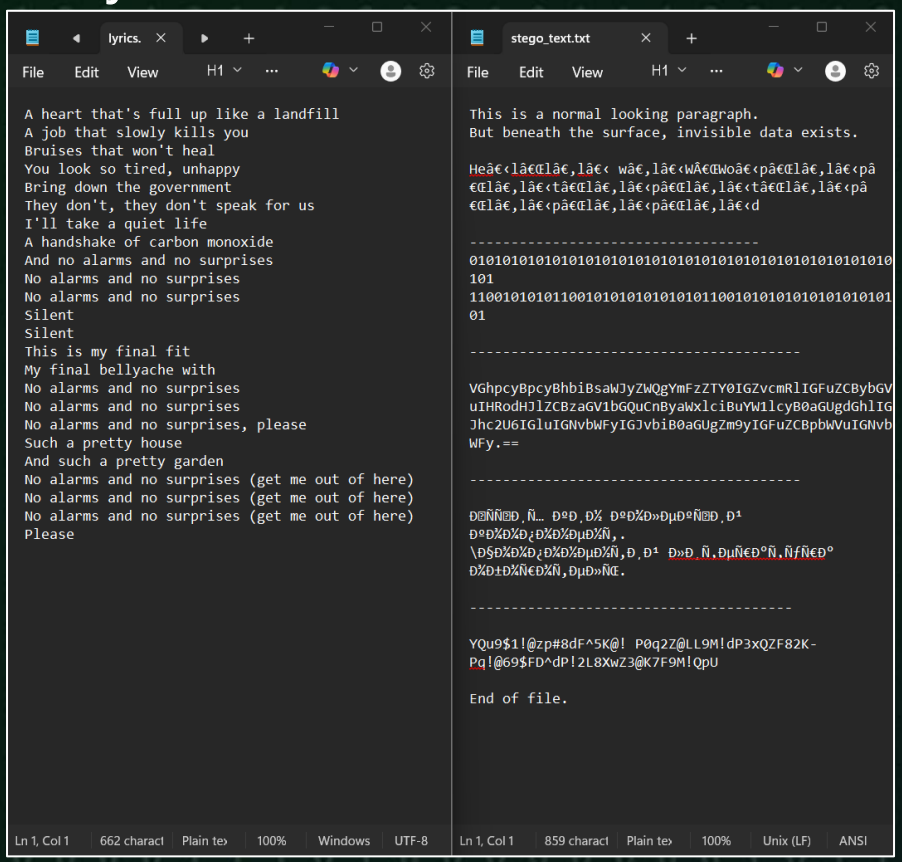
Embedding Method Along with generated embedded file

Extraction method along with successfully extracted payload

*To be noted, any file can be embedded within an image, a .txt file was used for demonstration purposes (except if the payload's size gets too large)
*To be noted, if the password during extraction is not the original password, nothing gets downloaded.
*If the password matches, the payload automatically gets downloaded.
*Lossless formats like .png work perfectly.

Text

Analysis

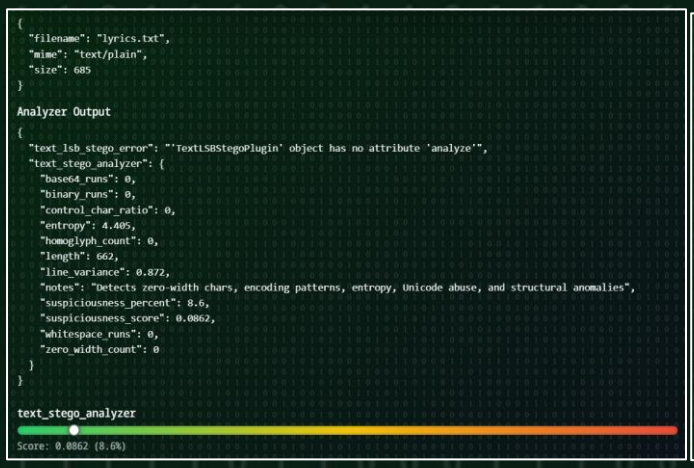


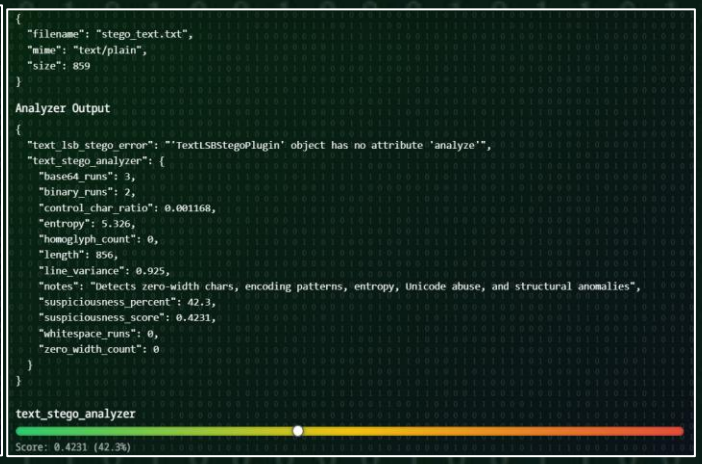
Comparing the two texts we will analyse.

UniSteno detects text-based steganography using:

- Zero Width Unicode character detection
  - Zero Width Space (ZWSP)
  - Zero Width Joiner (ZWJ)
- Homoglyph analysis
  - Cyrillic vs Latin lookalikes
- Entropy Estimation
  - Identifies unnatural character distributions

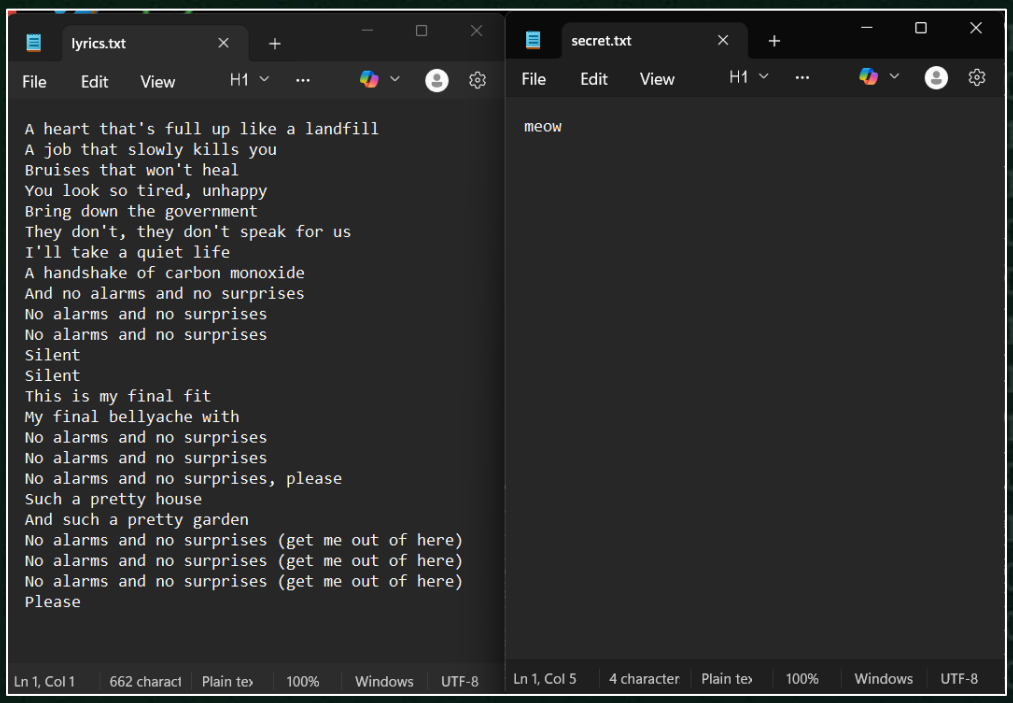
A weighted sum of these factors gives us the score.



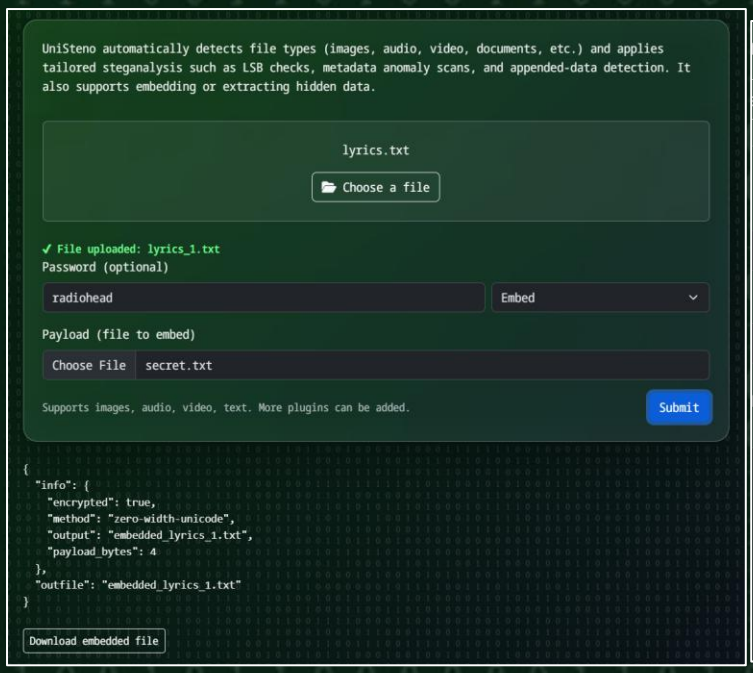


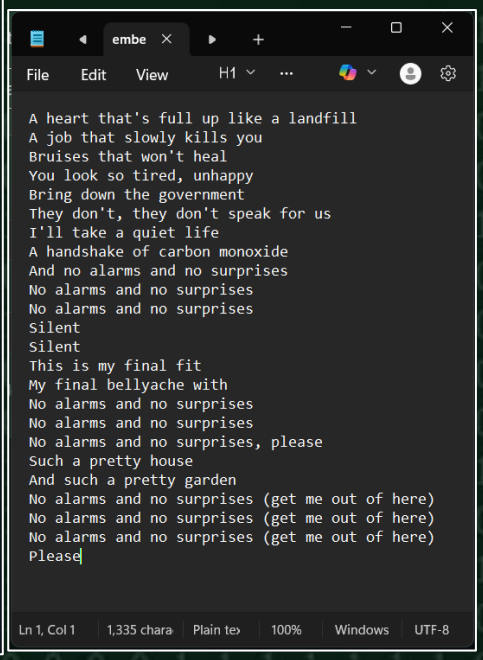
8.6% for normal English lyrics and **42.3%** for a bunch of suspicious stuff

Embed/Extract

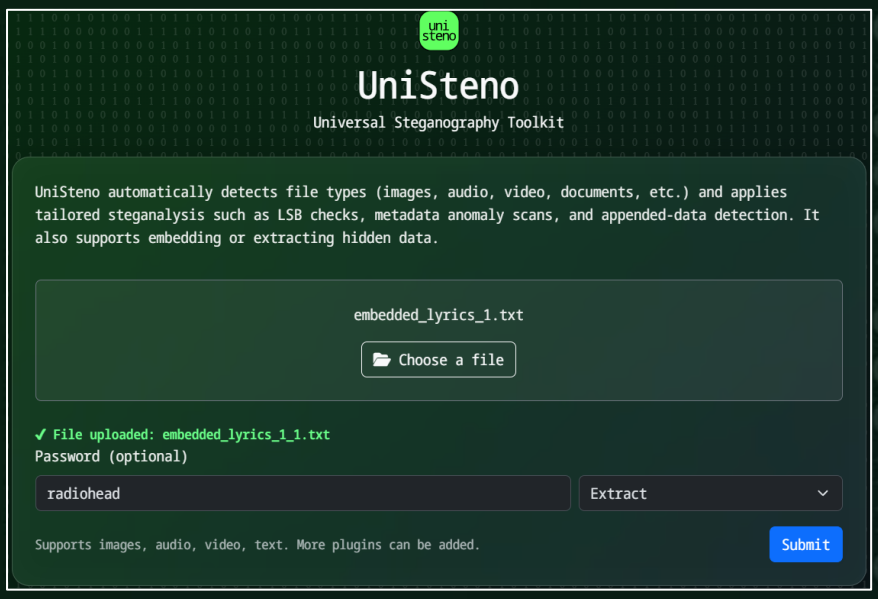


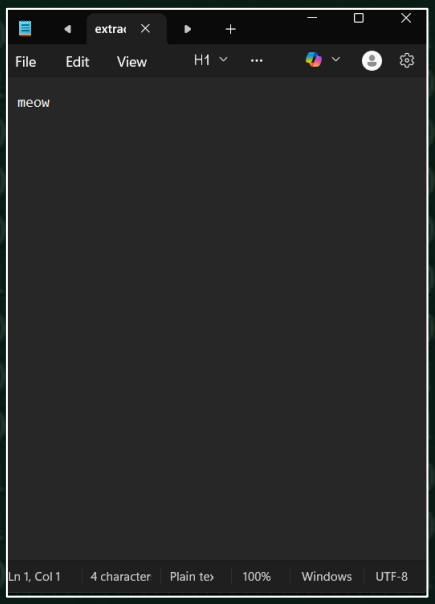
File and secret message to be embedded





Embedding Method Along with generated embedded file





Extraction method along with successfully extracted payload

Hidden text is encoded using zero-width characters, payload bits are invisibly inserted between words and text remains visibly unchanged.

During extraction, Zero-width characters are scanned and decoded, payload is reconstructed and validated.

Password Encryption is supported

PDFs

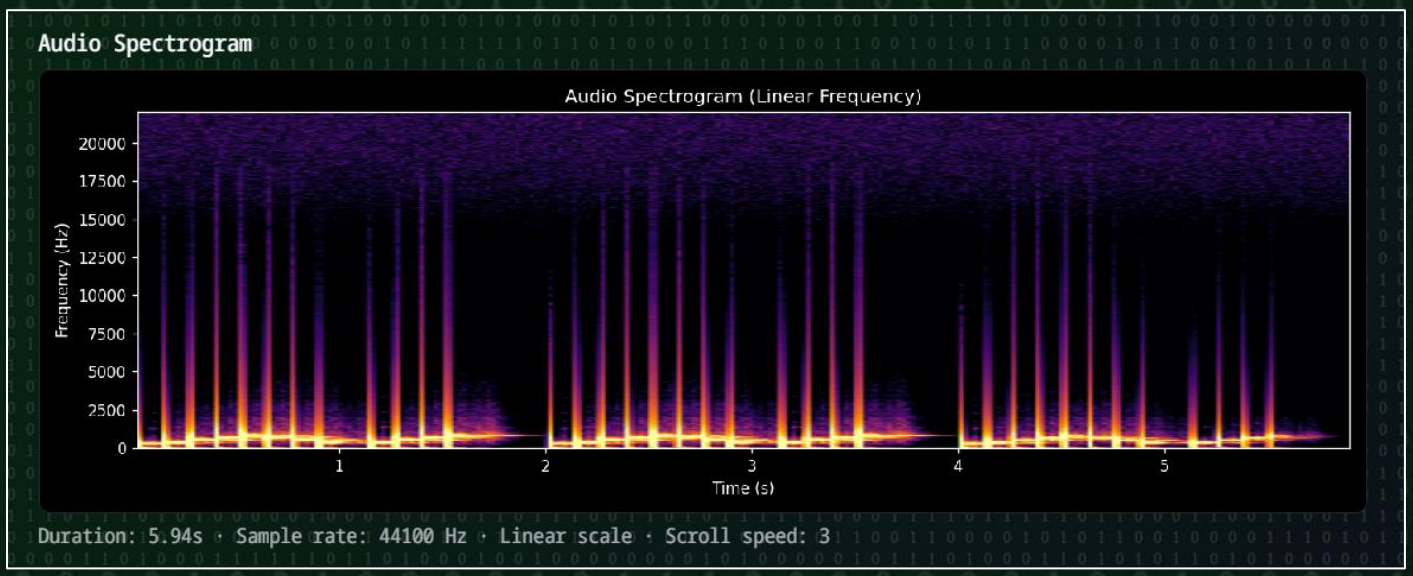
*Analyse, Embed, and Extract is the same as text.

*Unisteno ensures document integrity remains intact.

*PDFs were included because they are a common carrier in malware and data exfiltration.

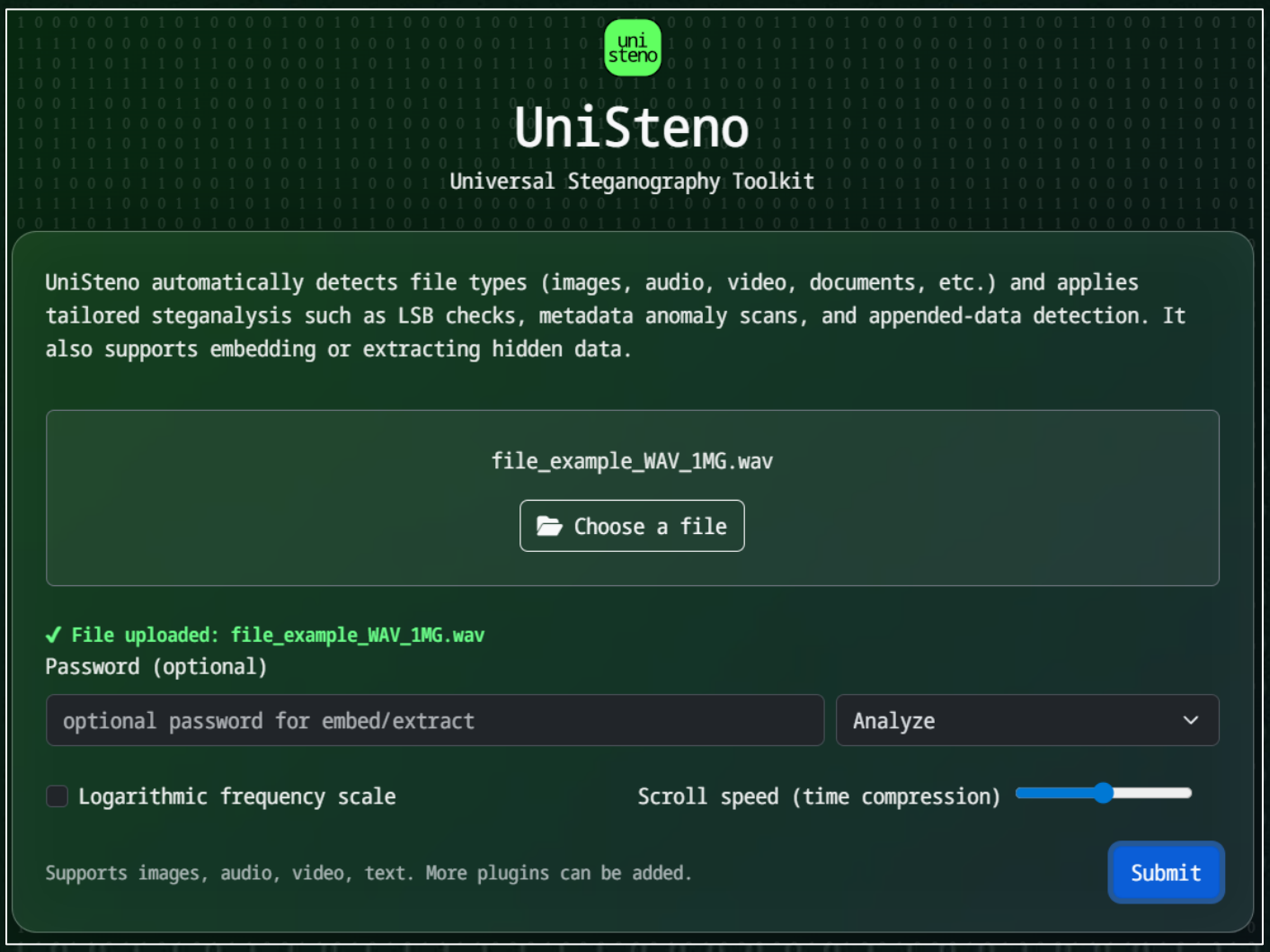
Audio

The spectrogram converts the audio signal to a time-frequency spectrogram. It helps visualise sudden frequency artifacts, high frequency noise bands, repetitive hidden patterns.



There is an option to toggle between Linear and Logarithmic Scale.

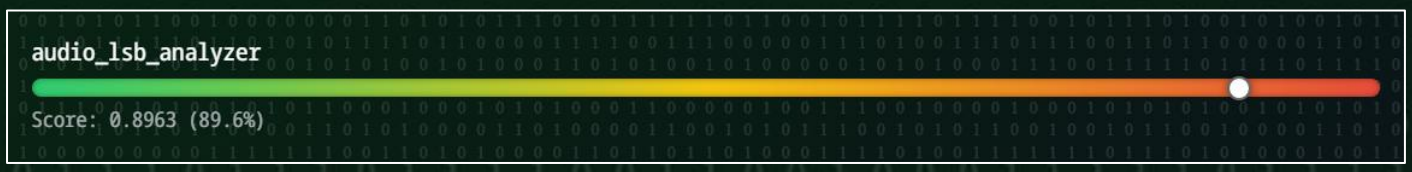
The scroll speed is also adjustable for lengthy audio files.



LSB and Statistical Inspection detects the least significant bits of audio samples.

It detects unnatural bit randomness and repeating bit patterns.

This is shown in form of a score:



Along with stats like:

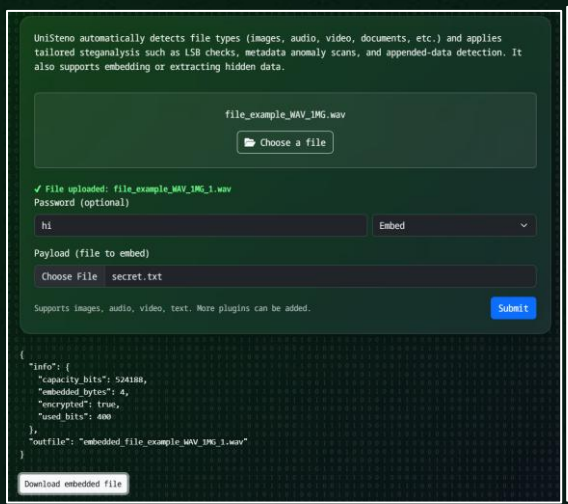
Analyzer Output

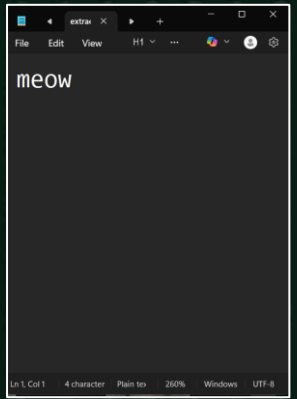

```
{
  "audio_lsb_analyzer": {
    "analyzed_samples": 220500,
    "chi_square": 4.041,
    "hf_energy_ratio": 0,
    "lsb_entropy": 1,
    "notes": "Detects LSB randomness and spectral noise consistent with audio steganography. Non-WAV formats are converted to WAV before analysis.",
    "sample_rate": 44100,
    "spectral_flatness": 0.003700000001117587,
    "spectral_variance": 1.1398999691009521,
    "suspiciousness_percent": 89.5999984741211,
    "suspiciousness_score": 0.8963000178337097
  },
```


Embedding/Extracting

The embedder uses a similar LSB embedder as the image LSB stego plugin.

We are trying to embed secret.txt in file_sample_WAV_1MG.wav.





(Extracted payload successfully)

Video

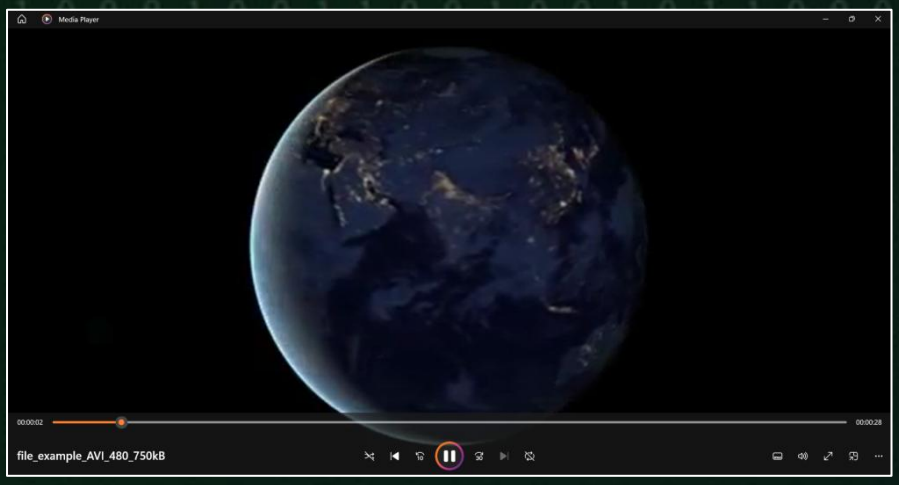
Analyse

Unisteno performs video steganalysis by:

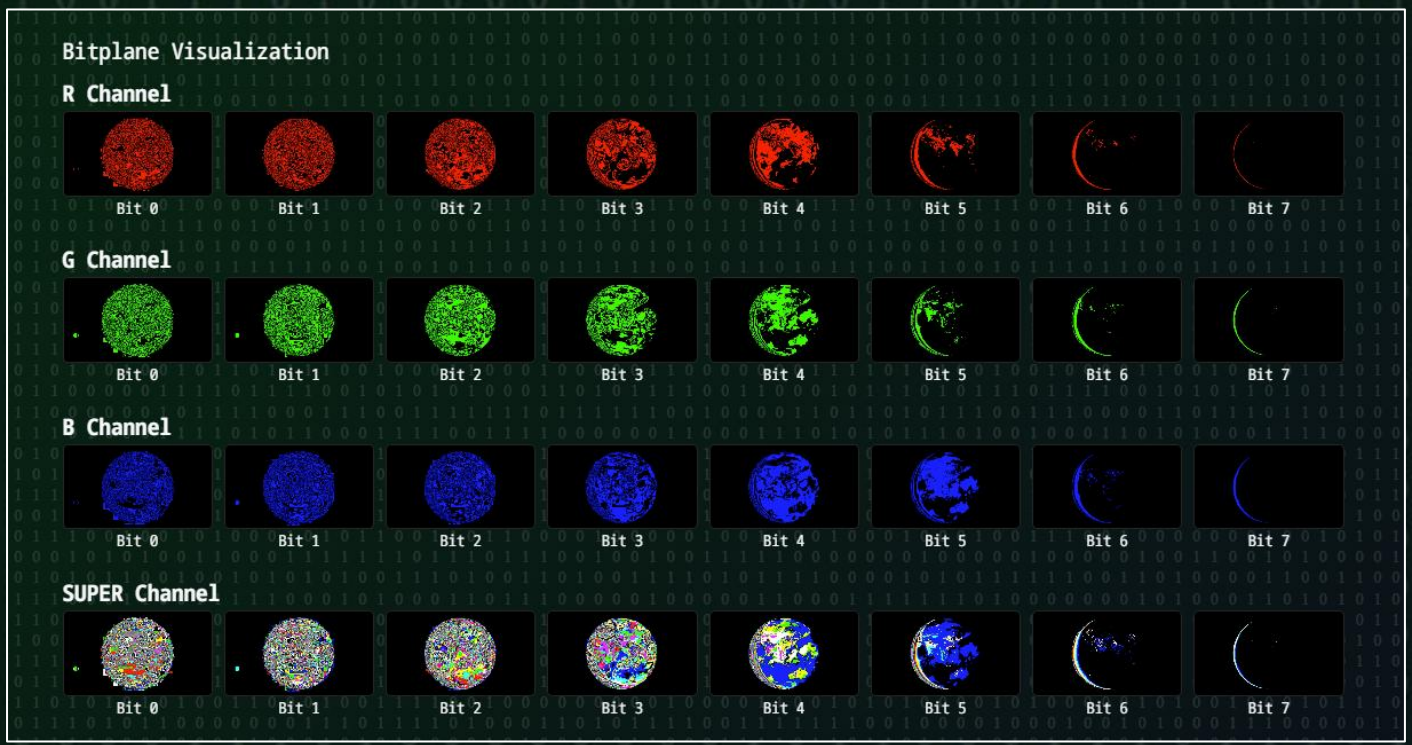
1. Converting all video's frames to images
2. Aggregating statistics across frames
3. Computing a suspiciousness score
4. Video bitplanes are calculated across R, G, B channels, and one channel, which is formed by superimposing the first three.
5. Each frame is then sewn together to form the original video, just in 32 different bitplanes.

Note that since we are processing hundreds, maybe thousands of images here, it is natural that Video analysis takes a lot of time

Shortcuts like only analysing say the first 50 frames, or analysing say every 5th frame is not taken, because of the responsibility of leakproof analysis.



Video to be analysed



Bitplane Visualisation

Embedding/Extracting

Since selection of all frames and embedding data randomly in any of those, took enormous time, Unisteno uses only the blue channel of the first frame of the video, this, however does not compromise the safety of the payload as it is still PRNG scattered (Similar to the image model)

*AVI file format needs to be used as it is lossless

To ensure accurate recovery, Unisteno embeds structured metadata along with the payload-

- Filename Length
- Filename
- Payload length
- Payload data
- CRC32 Checksum

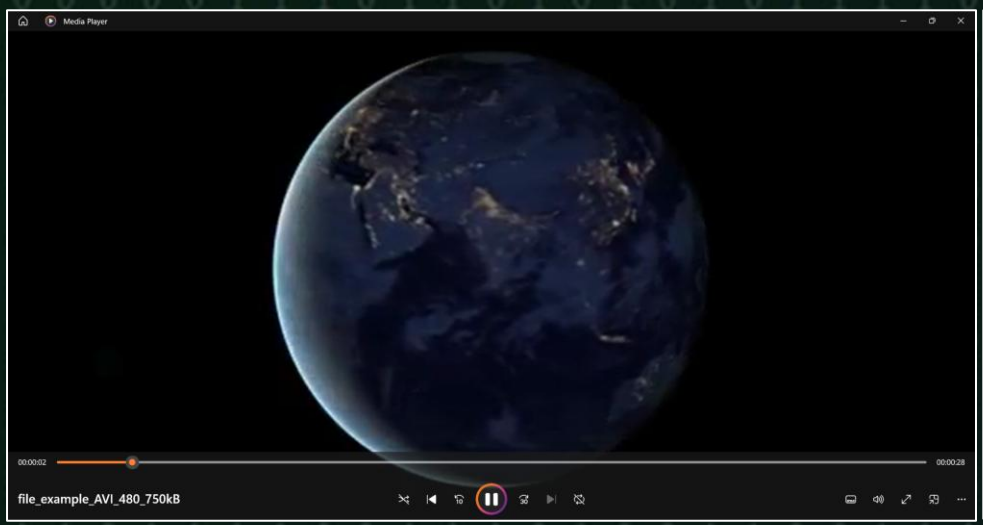
A cryptographically strong PRNG is seeded using a SHA-256 hash of the user password, As in Image encoding.

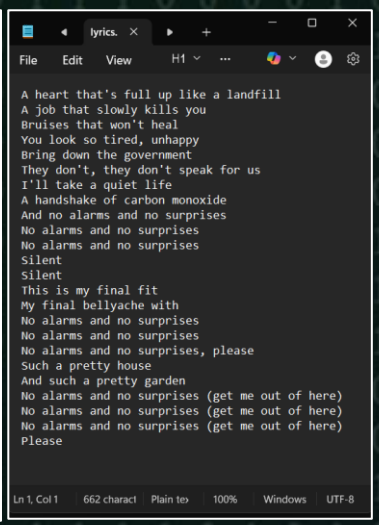
Even if the attacker knows the data is in the blue channel of the first frame, they cannot extract the data without the password.

Extraction is also done similarly to image extraction, first frame is read, blue channel is flattened into a bitstream, the same PRNG reconstructs embedding positions, metadata fields are decoded in sequence, payload bytes are reconstructed, and CRC32 verifications ensure correctness.

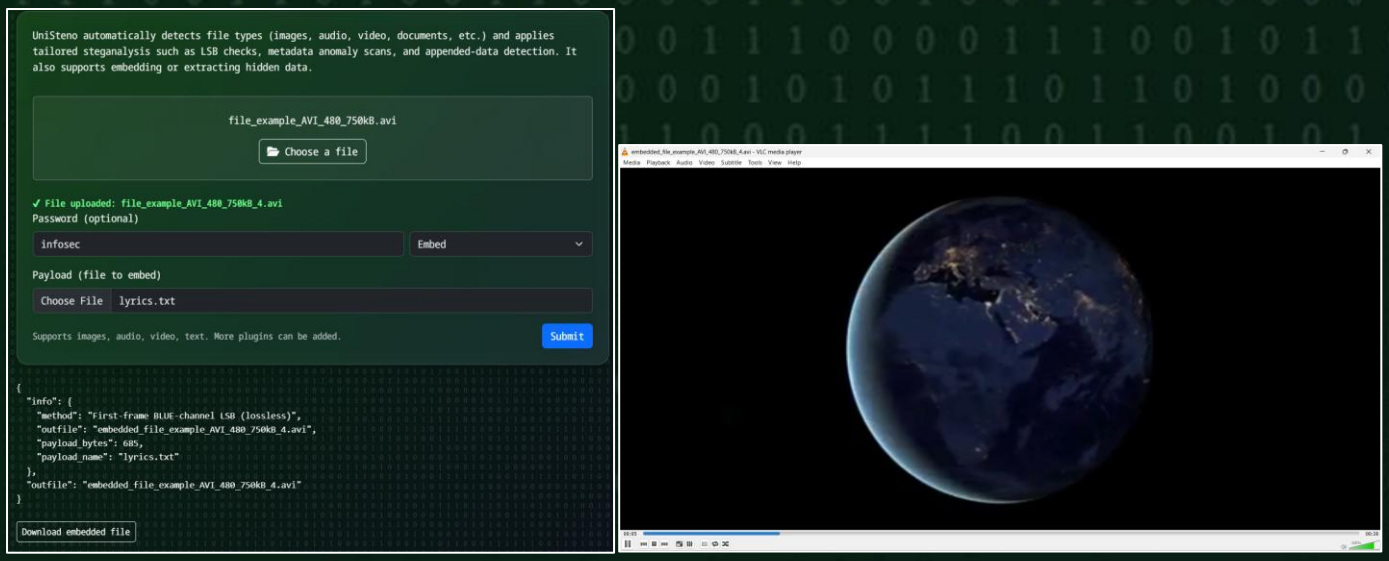
Performance wise, Time complexity is equivalent to image steganography because only one image in being used for embedding.

Unisteno's video embedding and extraction system provides a practical, secure and efficient steganography solution by leveraging first-frame LSB embedding combined with PRNG scattering. For the demo, we use an AVI file.

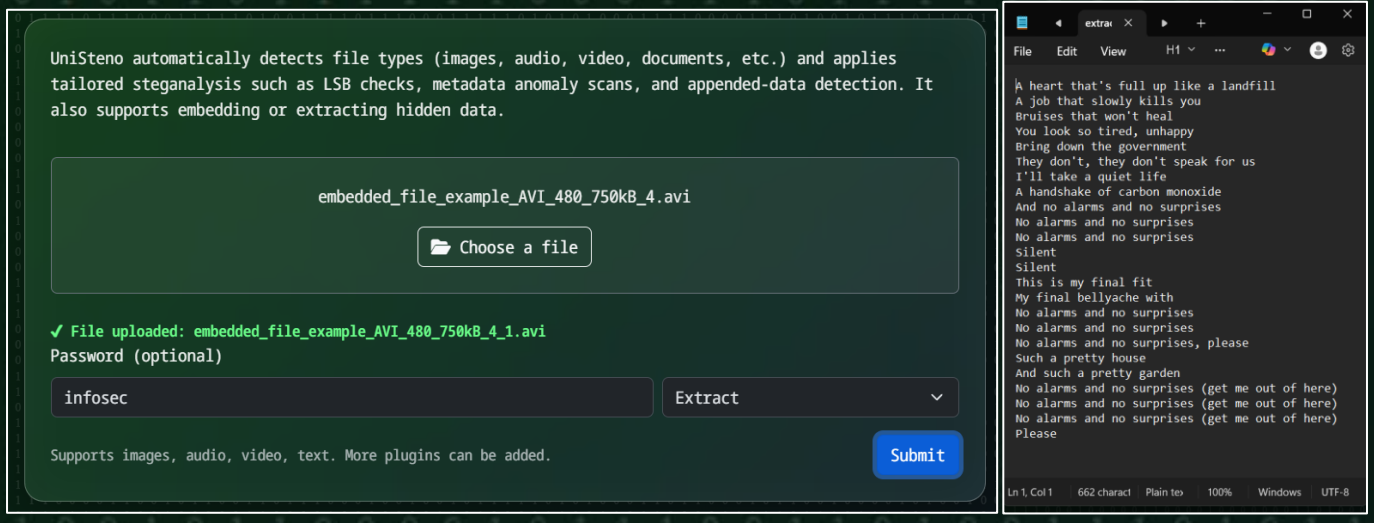




Video and the payload to be embedded in it



Embedding procedure and Embedded Video



Extraction procedure and successfully extracted payload

Graphical User Interface

1. Drag and drop file upload (Very convenient)
2. Analyse/Embed/Extract modes
3. Visual Bitplane Grids
4. Histogram Plots
5. Suspiciousness indicator bars
6. Audio Spectrograms (for audio files)

Technology Stack

Frontend:

- HTML5
- CSS3
- Bootstrap
- Javascript

Backend

- Python (Flask)

Libraries

- flask
- pillow
- numpy
- python-magic
- python-magic-bin
- PyPDF2
- scipy
- pydub
- matplotlib
- PyCryptodome
- OpenCV
- FFmpeg

Tools

- Github
- Docker

Challenges Faced

- Handling multiple media formats uniformly
- Designing a generic plugin interface
- Balancing performance with deep analysis
- Video codec compatibility across browsers
- Preventing false positives in statistical detection
- Biggest challenge was video analysing, embedding and extraction
- Javascript handling was a challenge too, getting to integrate frontend and backend

Conclusion

Unisteno successfully delivers a unified, extensible steganography toolkit.

Despite how complex it is to support multiple formats, Unisteno handles the issue very well with different plugins that do different jobs to different file types.

Unisteno can be used as a research and development tool because of how flexible it is.

About the Author

Adhiraj Singh

Bachelor of Technology in Chemical Engineering

IIT (ISM) Dhanbad

Interests:

Cybersecurity, CTFs, Digital Forensics, Full-Stack Development, Graphic Design, Video Editing

GitHub: https://github.com/adhiirraj

LinkedIn: https://www.linkedin.com/in/adhiraj-singh-861778260/

Email: 25je0608@iitism.ac.in