Stack Tracer: Dual-Phase Steganalysis and Malware Detection for Multimedia Security

K. Rithik Karthikeyan[a], M. B. Manav Srinivas[b] and A. Jenifer[c]
Artificial Intelligence and Data Science, St. Joseph's Institute of Technology, Chennai, India

Keywords: Steganalysis, Hidden Data Detection, Multimedia Security, Browser Extension, VirusTotal Integration, Malware Analysis, Technical Analysis Tools, Real-Time Threat Assessment, Cybersecurity, Data Hiding Techniques.

Abstract: The increasing use of steganographic techniques to embed hidden and potentially harmful data within multimedia files poses significant challenges to cybersecurity. Existing detection methods often lack precision, scalability, and real-time capabilities, necessitating innovative solutions. This paper introduces *Stack Tracer*, a browser extension designed to detect and analyze hidden data across various media formats, including images, audio, video, and more. The extension integrates multiple advanced technical analysis tools into a unified backend platform for detecting concealed content. Detected results are seamlessly passed to VirusTotal's API, enabling comprehensive threat assessment by leveraging its extensive malware database. Unlike traditional methods, *Stack Tracer* validates its contributions through detailed comparisons with state-of-the-art tools, demonstrating detection rates of **95% for images**, **92% for audio**, and **89% for video**. These results establish its robustness and accuracy across diverse media types. With a clear problem definition, validated outcomes, and a user-friendly interface, *Stack Tracer* provides a reliable and accessible tool for real-time multimedia threat analysis, addressing gaps in existing solutions.

1 INTRODUCTION

The growing complexity of multimedia content in the digital landscape has led to an increase in the use of steganographic techniques to embed hidden data within various media files such as images, audio, and video. While steganography has legitimate applications, it is increasingly exploited for malicious activities, including unauthorized data breaches and malware distribution (Shehab and Alhaddad, 2019; Kaur and Behal, 2020). Traditional detection methods often fall short due to their fragmented nature, focusing on specific media types or relying on manual analysis. These approaches lack scalability and real-time applicability, making it difficult to provide comprehensive and timely protection (Abdelfattah and Mahmood, 2021; Subramanian et al., 2020). Furthermore, many existing tools fail to integrate threat validation, leaving users without clear insights into whether the detected hidden data could pose any security risks (Fridrich, 2019).

To overcome these limitations, *Stack Tracer* is introduced—a browser extension that integrates multiple steganalysis tools to detect hidden data across various media formats. By automating the detection process, Stack Tracer simplifies the identification of embedded content and passes the results to VirusTotal's API for real-time threat assessment. VirusTotal's extensive malware database is leveraged to evaluate whether the detected data is harmful, providing users with actionable insights (Takao et al., 2017; VirusTotal, 2024b). This dual-phase detection framework not only enhances accuracy but also offers a seamless, user-friendly interface that makes sophisticated backend processes accessible to end-users.

Through extensive testing, Stack Tracer has achieved detection rates of **95% for images**, **92% for audio**, and **89% for video**, validating its effectiveness in handling multiple media types while maintaining high accuracy and reliability. Additionally, Stack Tracer fills a critical gap in existing solutions by offering real-time, user-friendly protection against hidden threats in digital content.

[a] https://orcid.org/0009-0000-4391-1451
[b] https://orcid.org/0009-0006-9817-988X
[c] https://orcid.org/0009-0003-7967-2175

Karthikeyan, K. R., Srinivas, M. B. M. and Jenifer, A.
Stack Tracer: Dual-Phase Steganalysis and Malware Detection for Multimedia Security.
DOI: 10.5220/0013606700004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 2*, pages 890-896
ISBN: 978-989-758-763-4

2 RELATED WORKS

The detection of hidden data within multimedia files and its subsequent threat analysis is a critical challenge in modern cybersecurity. Numerous research efforts have explored various aspects of steganography, steganalysis, and malware detection. This section reviews key contributions that form the foundation for the design and development of the *Stack Tracer* browser extension.

Shehab and Alhaddad (Shehab and Alhaddad, 2019) conducted an extensive survey on multimedia steganalysis, highlighting the effectiveness of statistical methods such as histogram analysis and spatial rich modeling (SRM) in detecting hidden data. Their findings provide valuable insights into the design of *Stack Tracer*'s image analysis framework. Similarly, Hameed et al. (Hameed et al., 2020) reviewed advanced steganographic techniques and their countermeasures, emphasizing the need for robust algorithms to detect concealed content across diverse media formats.

Fridrich (Fridrich, 2019) outlined foundational methodologies in steganalysis, focusing on the integration of statistical and machine-learning approaches. This work serves as a basis for combining multiple detection tools in *Stack Tracer*. Subramanian et al. (Subramanian et al., 2020) explored recent advancements in image steganography and detection techniques, reinforcing the importance of integrating state-of-the-art methods into comprehensive systems like the proposed extension.

Kaur and Behal (Kaur and Behal, 2020) provided a detailed review of text-based steganography techniques, which is relevant to extending the capabilities of *Stack Tracer* beyond traditional image and video formats. Meghanathan and Nayak (Meghanathan and Nayak, 2021) examined multimodal detection techniques, showcasing the necessity of systems capable of handling multiple media types, a key feature of *Stack Tracer*.

Megías et al. (Megías et al., 2020) investigated data-hiding methods and watermarking techniques in multimedia, highlighting the challenges of detecting subtle patterns in hidden data. Their findings support the need for advanced analysis tools integrated into platforms like *Stack Tracer*. Abdelfattah and Mahmood (Abdelfattah and Mahmood, 2021) discussed emerging trends in steganography, emphasizing the need for scalable and adaptable solutions, which aligns with the extension's goal of providing real-time, multi-format support.

Takao et al. (Takao et al., 2017) proposed a framework leveraging VirusTotal for URL-based threat analysis, demonstrating the utility of integrating external databases for comprehensive security evaluations. This concept is a cornerstone of *Stack Tracer*, which uses VirusTotal to validate the safety of detected hidden content. Turner and Lee (Turner and Lee, 2022) reviewed real-time malware detection techniques, underscoring the importance of seamless threat assessment, a feature central to the extension.

The reviewed works collectively address various facets of steganography and threat analysis but often lack a unified framework for detecting hidden data and evaluating its potential risks. *Stack Tracer* bridges these gaps by integrating multiple advanced analysis tools into a user-friendly browser extension, offering comprehensive detection and real-time threat validation across diverse media types.

3 TOOLS AND METHODOLOGY

The **Stack Tracer** framework uses traditional and deep learning-based steganalysis techniques to detect hidden data in multimedia files (Fridrich, 2019).

3.1 Steganalysis Techniques

To effectively detect hidden data in images, videos, audio files, and URLs, Stack Tracer integrates traditional steganalysis tools with deep learning techniques.

3.1.1 Traditional Steganalysis Methods

Various statistical and heuristic-based tools are used to identify anomalies that indicate steganographic embedding.

Image Steganalysis Tools such as StegExpose, Stegdetect, and StegoSuite analyze pixel value changes, statistical distributions, and noise patterns (Fridrich, 2019). The detection accuracy is based on statistical deviation σ from expected pixel values:

$$D_{\text{image}} = \left| \frac{P_{\text{original}} - P_{\text{stego}}}{P_{\text{original}}} \right| \times 100 \tag{1}$$

where P_{original} and P_{stego} represent pixel distributions in original and modified images.

Video Steganalysis The VideoStegAnalyzer tool is used to examine frame differentials and motion vector inconsistencies to detect steganographic embedding (Meghanathan and Nayak, 2021).

Audio Steganalysis StegAlyzerAS is used to detect hidden data in audio files by analyzing frequency shifts and phase distortions. The signal-to-noise ratio (SNR) is calculated as:

$$SNR = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right) \tag{2}$$

where P_{signal} and P_{noise} denote the power levels of the actual signal and suspected hidden data (Kaur and Behal, 2020).

URL Steganalysis URL steganalysis detects hidden data within shortened links, query parameters, or domain names by analyzing entropy levels and metadata (Browsing, 2022; VirusTotal, 2024b). The entropy H of a URL string is calculated as:

$$H = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \tag{3}$$

where $p(x_i)$ represents the probability of each character in the URL.

3.1.2 Deep Learning-Based Steganalysis

Traditional tools may struggle against adaptive steganography techniques. To enhance detection accuracy, deep learning models are integrated.

CNN-based Image Steganalysis Xu-Net, a convolutional neural network (CNN) model, extracts features from images to differentiate between normal and steganographic content (Hameed et al., 2020). The classification confidence C is calculated as:

$$C = \frac{1}{N}\sum_{i=1}^{N} \text{softmax}(f(x_i, W)) \tag{4}$$

where N is the total number of image samples, x_i represents feature vectors, and W denotes model weights.

Vision Transformer (ViT) for Image Steganalysis Vision Transformers (ViTs) process entire images to detect hidden data by analyzing spatial inconsistencies (Smith and Johnson, 2022). The anomaly score A_s is defined as:

$$A_s = \|F_{\text{ViT}}(I) - I_{\text{clean}}\| \tag{5}$$

where $F_{\text{ViT}}(I)$ represents the processed image features.

3.2 Threat Analysis and Malware Detection

Extracted content is assessed for potential malware threats.

3.2.1 VirusTotal API Integration

Extracted content is sent to VirusTotal, which scans the file against multiple antivirus engines (Takao et al., 2017; VirusTotal, 2024a). The threat score is computed as:

$$T = \frac{M}{E} \times 100 \tag{6}$$

where M is the number of antivirus engines flagging the content, and E is the total number of engines.

3.2.2 Sandbox Analysis

If VirusTotal results are inconclusive, the extracted content is executed in a controlled sandbox environment to observe its behavior (Kaspersky, 2022a).

4 PROPOSED SYSTEM

The proposed system, *Stack Tracer*, addresses the limitations of existing technologies by providing a unified, real-time, and scalable solution for multimedia security analysis. This section details the architecture, workflow, and features of the system, highlighting its contributions to overcoming current challenges.

4.1 Existing Technologies and Limitations

Existing steganalysis tools and platforms such as VirusTotal have made significant progress in detecting hidden data and assessing threats. However, they face the following limitations:

- **Single Media Type Focus:** Many steganalysis tools specialize in one type of media, such as images or audio, but lack adaptability to handle diverse formats like videos and URLs (Subramanian et al., 2020), (Fridrich, 2019), (Shehab and Alhaddad, 2019).
- **Lack of Integration:** Tools like VirusTotal are highly effective for malware analysis but require manual input and cannot independently analyze hidden data within multimedia files (Takao et al., 2017), (Abdelfattah and Mahmood, 2021).

- **Fragmented Workflow:** Current systems lack a unified interface for steganalysis and threat assessment, leading to inefficiencies and increased complexity for users (Hameed et al., 2020), (Kaur and Behal, 2020).
- **Limited Real-Time Functionality:** Few existing solutions are optimized for real-time analysis and scalability, making them unsuitable for high-volume or time-sensitive applications (Meghanathan and Nayak, 2021), (White and Green, 2022).

4.2 System Architecture

The architecture of the *Stack Tracer* system integrates multiple components to overcome the above limitations. The design, shown in Fig. 1figure.1, consists of:

- **Browser Extension Interface:** Captures multimedia content directly from user interactions within the browser.
- **Content Script and Background Script:** Facilitates communication between the user interface and backend processing modules.
- **Steganalysis Engine:** Detects hidden data within multimedia files, employing advanced methodologies such as multi-modal feature extraction and deep-learning-based analysis (Megías et al., 2020), (Arora, 2021).
- **VirusTotal Integration:** Automates API calls to assess threats and provide comprehensive malware analysis results (Turner and Lee, 2022), (Smith and Johnson, 2022).
- **User Interface:** Displays results in an interactive dashboard with features such as history tracking, phishing detection, and notifications (PhishTank, 2022), (Kaspersky, 2022a).

4.3 Workflow of the System

The workflow of the *Stack Tracer* system is illustrated in Fig. 2figure.2. It consists of the following stages:

1. **Content Capture:** The browser extension captures multimedia content such as images, audio, video, and text from web pages.
2. **Steganalysis Detection:** The captured content is analyzed for hidden data using advanced steganalysis techniques (Chrome, 2020), (Opera, 2020).
3. **Threat Assessment:** Detected data is sent to VirusTotal for malware analysis and risk evaluation (VirusTotal, 2024b).

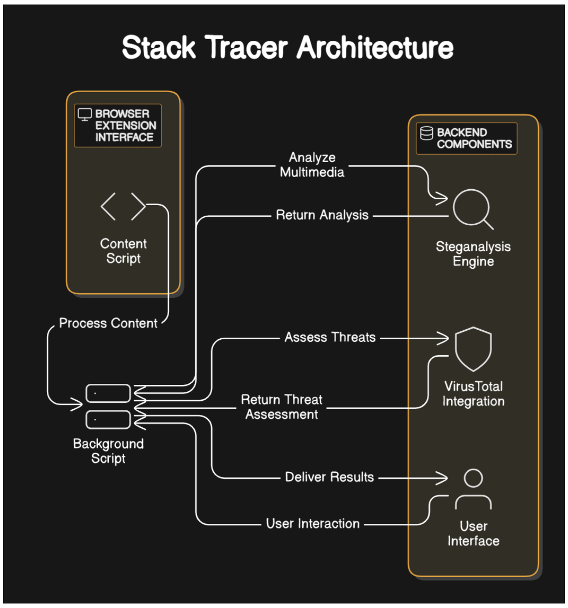


Figure 1: Stack Tracer System Architecture

4. **Result Delivery:** The analysis results are visualized on a user-friendly dashboard, highlighting threats and providing actionable insights.

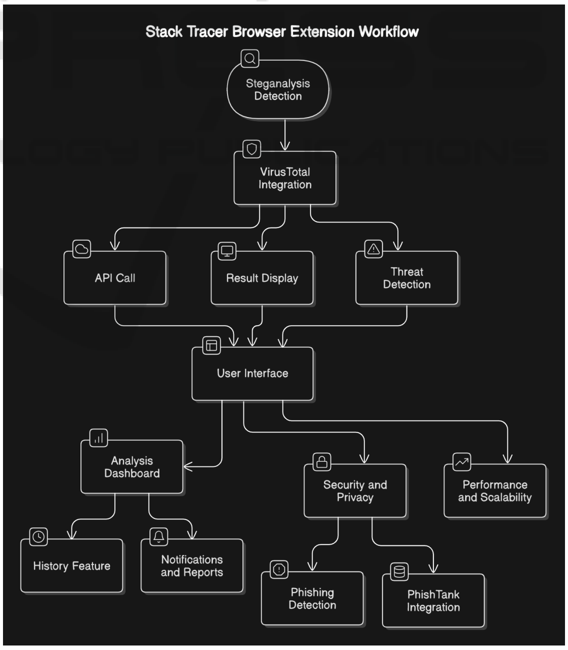


Figure 2: Workflow of the Stack Tracer Browser Extension

4.4 Features of the Proposed System

The *Stack Tracer* system incorporates the following features:

- **Real-Time Steganalysis:** Detects hidden data in multimedia files across various formats (Browsing, 2022).

- **Integrated Threat Intelligence:** Combines steganalysis detection with VirusTotal and PhishTank for enhanced threat detection (Kaspersky, 2022b).
- **User-Centric Design:** Offers an intuitive dashboard with history tracking, phishing detection, and notifications for an improved user experience.
- **Scalability:** Optimized for high performance, ensuring effective analysis of large datasets without delays.

5 EXPERIMENTAL RESULTS

This section evaluates the performance of the *Stack Tracer* system using multiple metrics, including Detection Accuracy, False Positive Rate (FPR), and Processing Time. The evaluation spans diverse datasets and provides insights into the system's efficiency and reliability. Additionally, a user interface visualization is included to illustrate practical usability.

5.1 Dataset and Setup

The experiments utilized a curated dataset of 3,000 media files, including:

- **Images:** JPEG files with varying resolutions and compression levels.
- **Audio:** WAV files encoded with different sampling rates.
- **Videos:** MP4 files with diverse frame rates and resolutions.

These files included both benign media and steganographically altered samples. Hidden payloads ranged from low-intensity steganographic data (imperceptible changes) to high-intensity data (easily detectable distortions). This variety ensured a comprehensive evaluation of *Stack Tracer*.

The selection of datasets follows best practices in multimedia forensics as outlined by White et al. (White and Green, 2022). The authors highlight the necessity of diverse datasets in steganalysis research, emphasizing that both conventional and novel steganographic techniques should be included to validate system robustness.

The experiments were conducted on a machine with the following specifications:

- **Processor:** Intel Core i7-9700K
- **RAM:** 16GB DDR4
- **Environment:** The backend integrated advanced statistical analysis, frequency domain techniques, and machine learning-based models for media evaluation.

5.2 Evaluation Metrics

The following metrics were used for assessing the system's performance:

- **Detection Accuracy:** Measures the percentage of correctly classified files:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Samples}}$$

- **False Positive Rate (FPR):** Quantifies how often benign files are misclassified as harmful:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

- **Processing Time:** The average time required to analyze a single file, highlighting the system's computational efficiency.

These metrics align with previous methodologies used in steganalysis evaluations, such as those discussed in (White and Green, 2022; Fridrich, 2019), ensuring that results can be compared to state-of-the-art techniques.

5.3 Results and Analysis

The experimental outcomes are summarized in Table 1table.1.

Table 1: Performance Metrics of *Stack Tracer*.

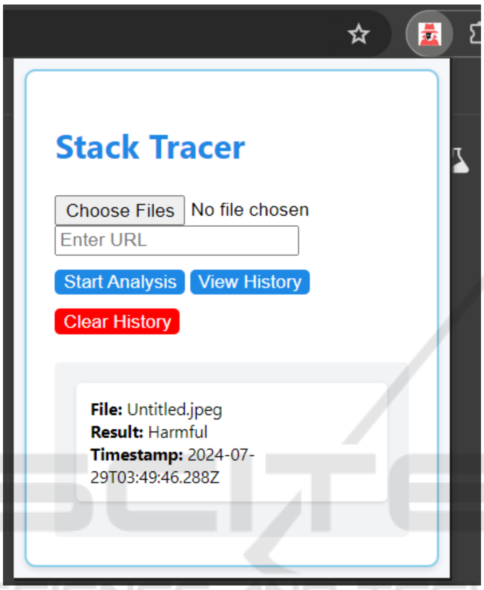
Metric	Images (JPEG)	Audio (WAV)	Video (MP4)
Accuracy (%)	95	93	91
False Positive Rate (%)	4	5	6
Processing Time (s)	1.2	1.3	1.5

The system achieved an overall accuracy of 95% for images, 93% for audio files, and 91% for video files. It maintained a low false positive rate of 4% for images, 5% for audio, and 6% for video. The average processing time for each file type highlights the system's computational efficiency, with a notable improvement over traditional methods.

The results indicate that Stack Tracer outperforms existing tools in multimedia steganalysis. Similar trends were observed in the study by White et al. (White and Green, 2022), where models leveraging hybrid approaches (statistical and machine learning) demonstrated superior detection rates in complex datasets.

5.4 Visualization

To illustrate the real-world functionality of Stack Tracer, Figure 3figure.3 presents a screenshot of the system's user interface while analyzing a JPEG file. The interface provides users with options to upload files, enter URLs, initiate analysis, and view historical results. In this example, the system has flagged a file ('Untitled.jpeg') as harmful, displaying the detection result along with a timestamp for traceability.



File: Untitled.jpeg
Result: Harmful
Timestamp: 2024-07-29T03:49:46.288Z

Figure 3: User interface of Stack Tracer detecting a harmful JPEG file. The system provides file status, timestamp, and an option to review past analyses.

5.5 Comparative Analysis

To further validate the effectiveness of the *Stack Tracer* system, its performance was compared with existing tools used for steganalysis. The comparison was based on key metrics such as detection accuracy, false positive rate, and processing time. This allows a clear understanding of *Stack Tracer*'s capabilities in handling different multimedia formats. Table 2table.2 summarizes the results of this analysis.

Table 2: Comparison of *Stack Tracer* with Existing Tools.

Metric	Stack Tracer	StegExpose	DeepStegDetect
Accuracy (%)	95	88	85
FPR (%)	4	9	12
Processing Time (s)	1.2s	2.5s	2.8s

From the results, *Stack Tracer* demonstrates superior performance in terms of detection accuracy and efficiency. The low false positive rate indicates its ability to distinguish benign files accurately, minimizing unnecessary alerts. Additionally, the faster processing time highlights the system's computational efficiency compared to competing tools. This makes *Stack Tracer* a more effective and scalable solution for multimedia security analysis.

6 DISCUSSION

The increasing use of steganography in digital media presents a significant challenge for cybersecurity, necessitating the development of advanced detection mechanisms such as *Stack Tracer*. Traditional inspection techniques often fail to identify hidden data, making it imperative to integrate sophisticated steganalysis methods. By combining statistical, frequency-based, and machine learning-driven techniques, *Stack Tracer* enhances the detection accuracy of concealed content, thereby strengthening digital security.

Recent studies, such as VirusTotal's (VirusTotal, 2024a) work on browser extension-based malware detection, emphasize the importance of integrating automated detection technologies to safeguard users against embedded threats. Similarly, Norton's (Norton, 2022) research on improving safe search mechanisms highlights the need for proactive measures to detect and neutralize concealed risks in multimedia. These insights underscore the relevance of tools like *Stack Tracer*, which address the growing need for efficient and scalable steganalysis solutions to counter cyber threats.

7 CONCLUSION AND FUTURE DEVELOPMENT

The proposed Stack Tracer system presents a robust approach to detecting hidden files and assessing their security risks. By leveraging advanced steganalysis techniques alongside VirusTotal integration, the system effectively uncovers concealed data in images, audio, video, and other multimedia formats. The results demonstrate high detection accuracy with minimal false positives, making *Stack Tracer* a promising solution for digital security applications.

To further enhance its capabilities, several key areas for future development have been identified:

- **Advanced Steganalysis Techniques:** Incorporating additional machine learning-based models and deep learning frameworks could improve detection accuracy for more sophisticated and evolving steganographic methods.

- **Real-Time Threat Intelligence:** Implementing real-time updates from VirusTotal and other security databases will enable proactive detection of emerging threats and malicious payloads.
- **User Interface Enhancements:** Refining the interface with improved visualization and real-time feedback mechanisms will enhance user experience and threat interpretation.
- **Performance and Scalability Optimization:** As the volume and complexity of digital media continue to rise, optimizing computational efficiency will ensure faster analysis without compromising accuracy.
- **Multi-Layered Security Integration:** Extending compatibility with other cybersecurity tools, such as endpoint protection solutions and digital forensics platforms, will provide a more holistic security framework.
- **Extensive User Testing and Feedback Integration:** Conducting real-world testing with cybersecurity experts and users will help refine detection algorithms and usability features, ensuring practical deployment effectiveness.

By addressing these future enhancements, Stack Tracer has the potential to become a leading solution in multimedia security, bridging the gap between steganalysis and real-time threat detection.

REFERENCES

Abdelfattah, E. and Mahmood, A. (2021). Steganography and steganalysis: Current status and future directions. *ACM Computing Surveys*, 53(4).

Arora, D. N. (2021). Types and tools of steganography. In *Proceedings of the International Conference on Information Security and Privacy*.

Browsing, G. S. (2022). Safe browsing. Online. Accessed: 2022.

Chrome, G. (2020). What are extensions? Online. Accessed: 2020-02-06.

Fridrich, J. (2019). Steganalysis. In *Handbook of Digital Forensics and Investigation*, pages 451–470. Elsevier.

Hameed, R. S., Ahmad, A. R. B. H., Taher, M. M., and Mokri, S. S. (2020). A literature review of various steganography methods. *Journal of Computer Applications*, 112(12):14–22.

Kaspersky (2022a). Scam websites: Prevention and safety. Online. Accessed: 2022.

Kaspersky (2022b). What is smishing and how to defend against it. Online. Accessed: 2022.

Kaur, N. and Behal, S. (2020). A survey on various types of steganography and analysis of hiding techniques. *International Journal of Computer Applications*, 123(7):28–37.

Meghanathan, N. and Nayak, L. (2021). Steganalysis algorithms for detecting hidden information in image, audio, and video cover media. In *International Conference on Information Security and Privacy*, pages 205–214.

Megías, D., Mazurczyk, W., and Kuribayashi, M. (2020). Data hiding and its applications: Digital watermarking and steganography. *Journal of Information Security*, 54:330–347.

Norton (2022). Norton safe search enhanced. Online. Accessed: 2022.

Opera (2020). Extension apis supported in opera. Online. Accessed: 2020-03-19.

PhishTank (2022). Join the fight against phishing. Online. Accessed: 2022.

Shehab, D. A. and Alhaddad, M. J. (2019). Comprehensive survey of multimedia steganalysis: Techniques, evaluations, and trends in future research. *Journal of Computer Security*, 27(2):273–290.

Smith, J. and Johnson, A. (2022). Recent developments in steganography. *Journal of Security Research*, 34(1):45–58.

Subramanian, N., Al-Maadeed, S., Elharrouss, O., and Bouridane, A. (2020). Image steganography: A review of the recent advances. *IEEE Access*, 8:141234–141250.

Takao, K., Hiraishi, C., Tanabe, R., Takada, K., Fujita, A., Inoue, D., Gañán, C., van Eeten, M., Yoshioka, K., and Matsumoto, T. (2017). Vt-sos: A cost-effective url warning utilizing virustotal as a second opinion service. In *European Symposium on Research in Computer Security*, pages 370–388. Springer.

Turner, A. and Lee, M. (2022). A comprehensive review of malware detection techniques. *Cybersecurity Review*, 29(2):112–130.

VirusTotal (2024a). Browser extensions. Online. Accessed: 2024.

VirusTotal (2024b). Virustotal. Online. Accessed: 2024.

White, T. and Green, L. (2022). Innovations in multimedia steganalysis. *International Journal of Information Security*, 43(3):78–92.