# Spam Mail Detection

# using Machine Learning (ML)

**Introduction:**

Spam detection is a crucial task in information security to protect users from unwanted and potentially harmful messages. Email and SMS remain popular communication channels, but they are often targets for spam campaigns. Traditional rule-based filters are limited in their effectiveness, making machine learning-based approaches more appealing due to their ability to adapt and generalize.

**Architecture:**

The architecture of our spam detection system comprises several key components:

1. Data Preprocessing and Feature Extraction:

Text Cleaning: Removing HTML tags, special characters, and stopwords.
Tokenization: Breaking down messages into individual words (tokens).
Feature Engineering: Using TF-IDF (Term Frequency-Inverse Document Frequency) to convert text into numerical vectors.

2. Machine Learning Models:

Multinomial Naive Bayes: A probabilistic classifier based on Bayes' theorem with strong performance for text classification tasks.
Support Vector Machines (SVM): Effective for high-dimensional data, SVM seeks to find the hyperplane that best separates classes.
Ensemble Methods: Combining multiple models (e.g., Voting Classifier) to improve overall performance.
Real-time Detection System:

3. Integration: Deploying the trained models into a real-time application using Flask.

Continuous Monitoring: Monitoring incoming messages and applying the detection system to classify as spam or not.

**Modules:**

1. Data Preprocessing:

Text Cleaning: Use regular expressions to remove unwanted characters and symbols.
Tokenization: Split text into tokens (words or n-grams) using libraries like NLTK (Natural Language Toolkit) or spaCy.
TF-IDF Vectorization: Convert text data into numerical vectors using the TF-IDF approach.

2. Machine Learning Models:

Model Selection: Choose appropriate algorithms like Naive Bayes, SVM, or ensemble methods based on performance metrics.
Model Training: Fit the selected models on the preprocessed data.
Model Evaluation: Assess performance using metrics like accuracy, precision, recall, and F1-score.

3. Real-time Detection System:

Web Application Development: Use Flask to create a web-based interface for the spam detection system.
Integration: Incorporate the trained models into the application for real-time prediction.

**User Interface:**

Design an intuitive interface to input messages and display spam classification results.
Implementation:

1. Data Preprocessing:

**Conclusion:**

The implementation of a spam detection system using machine learning models such as Naive Bayes, SVM, and ensemble methods demonstrates the effectiveness of these techniques in classifying spam and non-spam messages. By integrating the models into a real-time detection system using Flask, we create a practical application that can be deployed to protect users from unwanted messages.

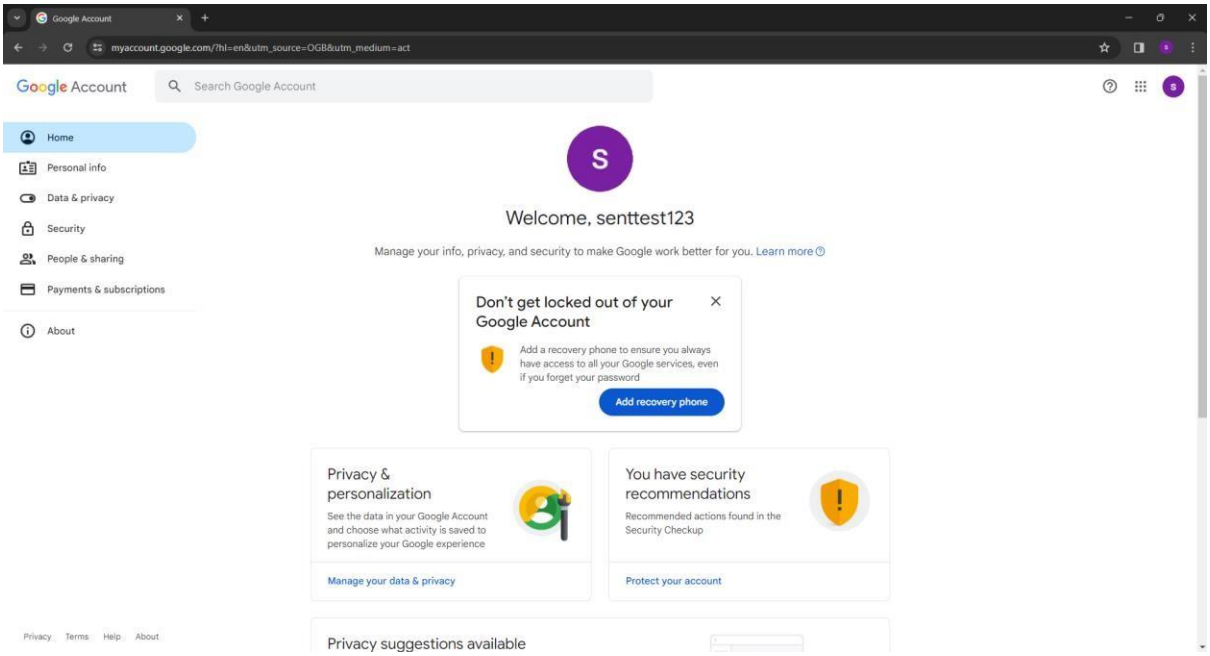
This expanded implementation plan includes detailed code snippets for data preprocessing, model training, evaluation, and integration into a Flask-based real-time detection system. Adjust and customize the code to fit your specific project requirements and dataset. The provided architecture and modules outline a structured approach towards building an effective spam detection system.

## Sending Spam Email

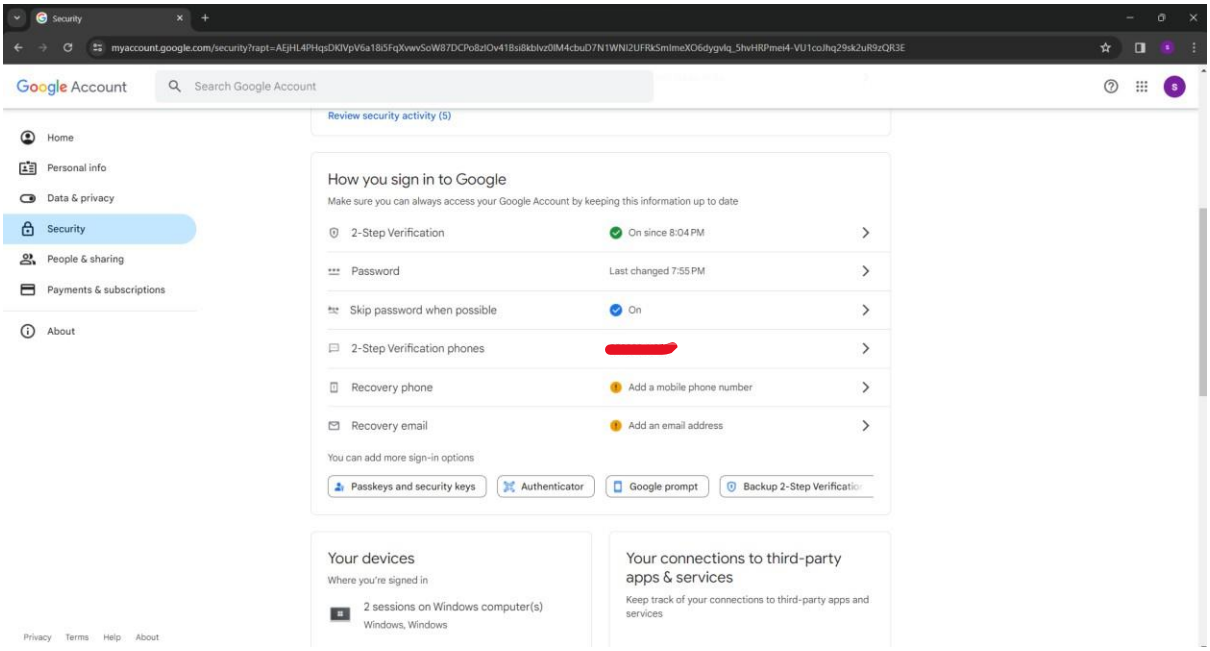
- Sending spam email from duplicate Gmail to a duplicate Gmail that have been created.

**Sender Gmail:** senttest43@gmail.com

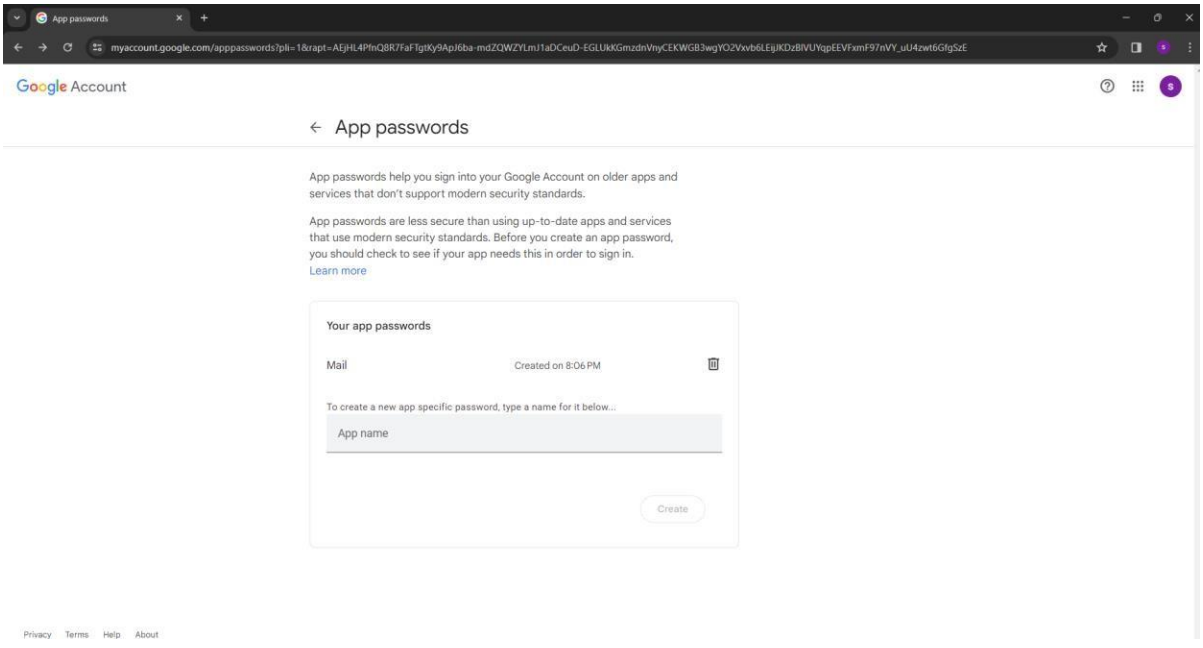
**Receiver Gmail:** receivetest123@gmail.com



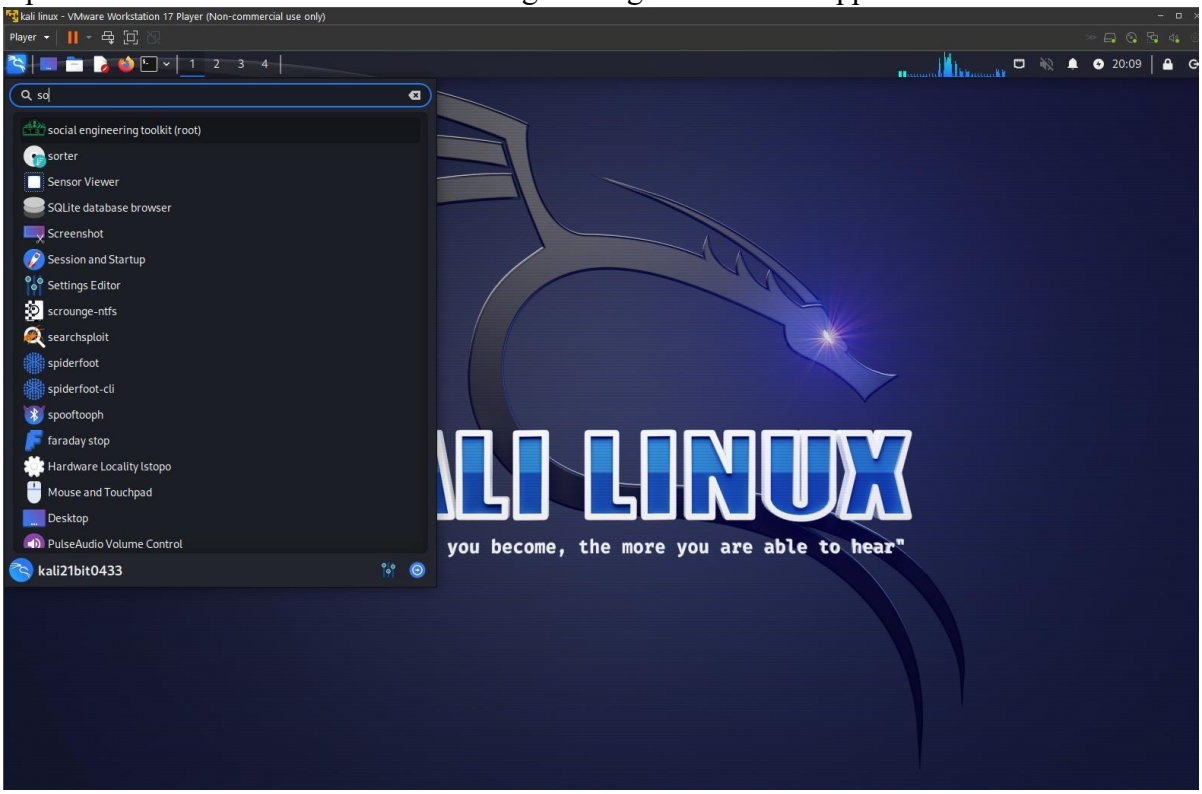
- Changing the setting in the sender Gmail for switching on the two-step verification



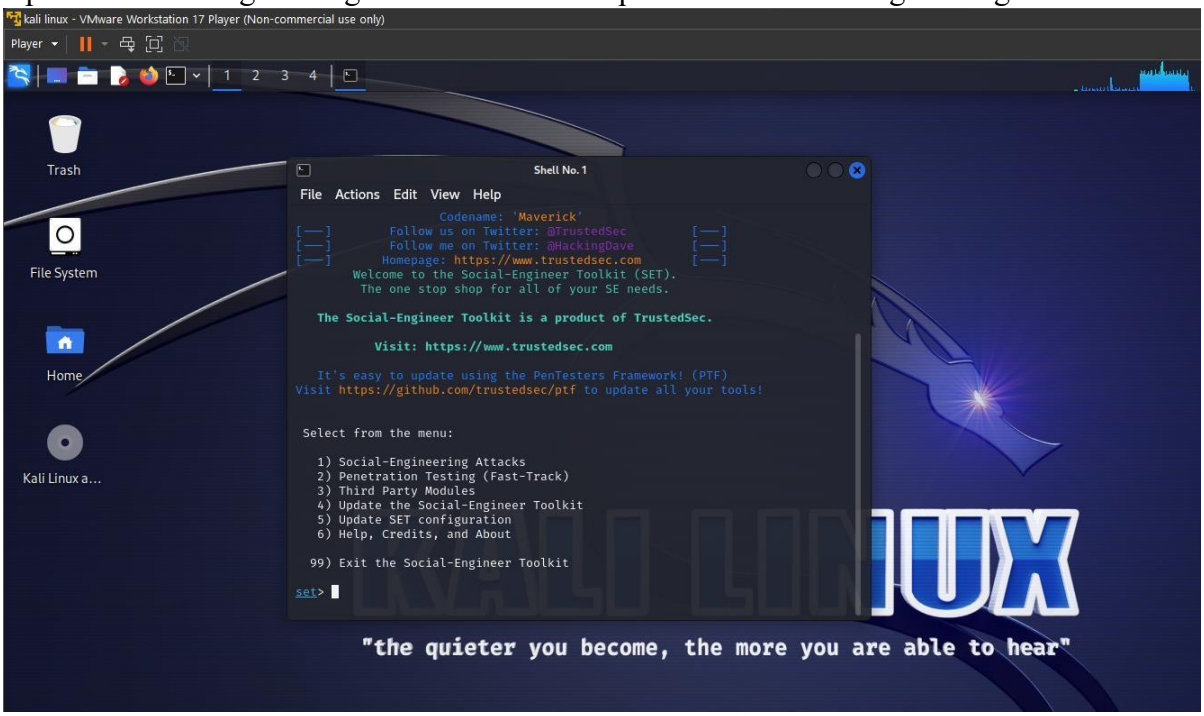

- Creating an App password for sending the spam email from the kali Linux. This password is used in the kali Linux while sending the email to the receiver.



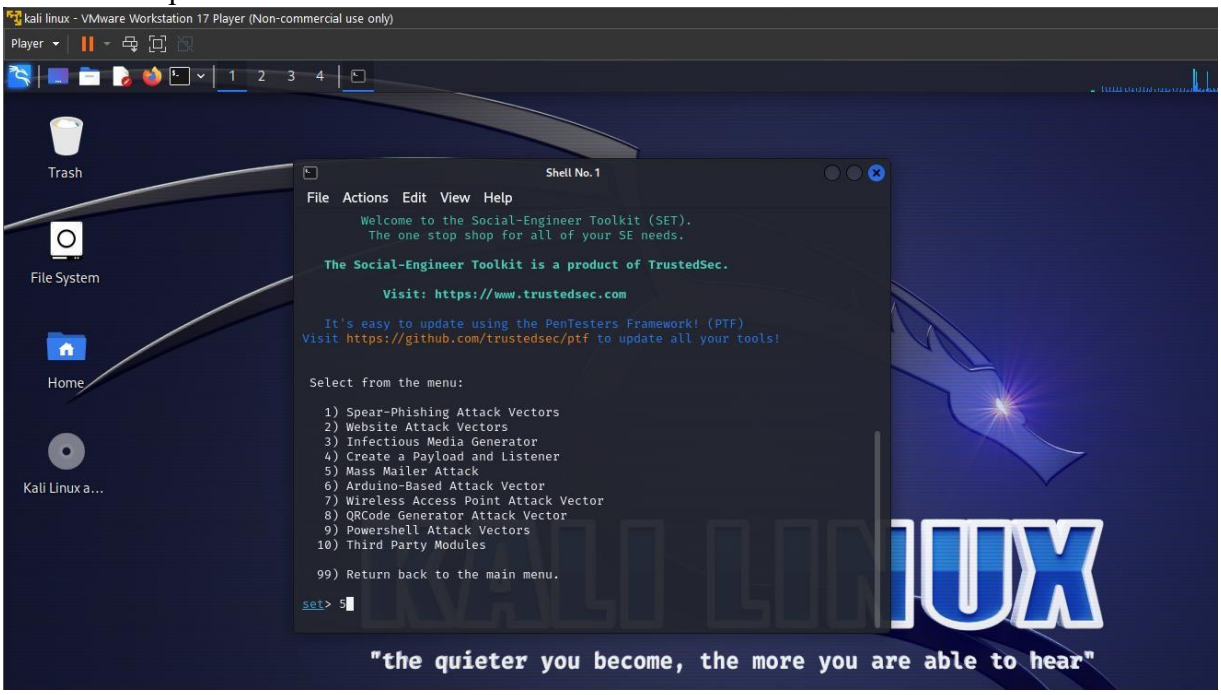
- Open kali Linux and search for social engineering toolkit in the applications.



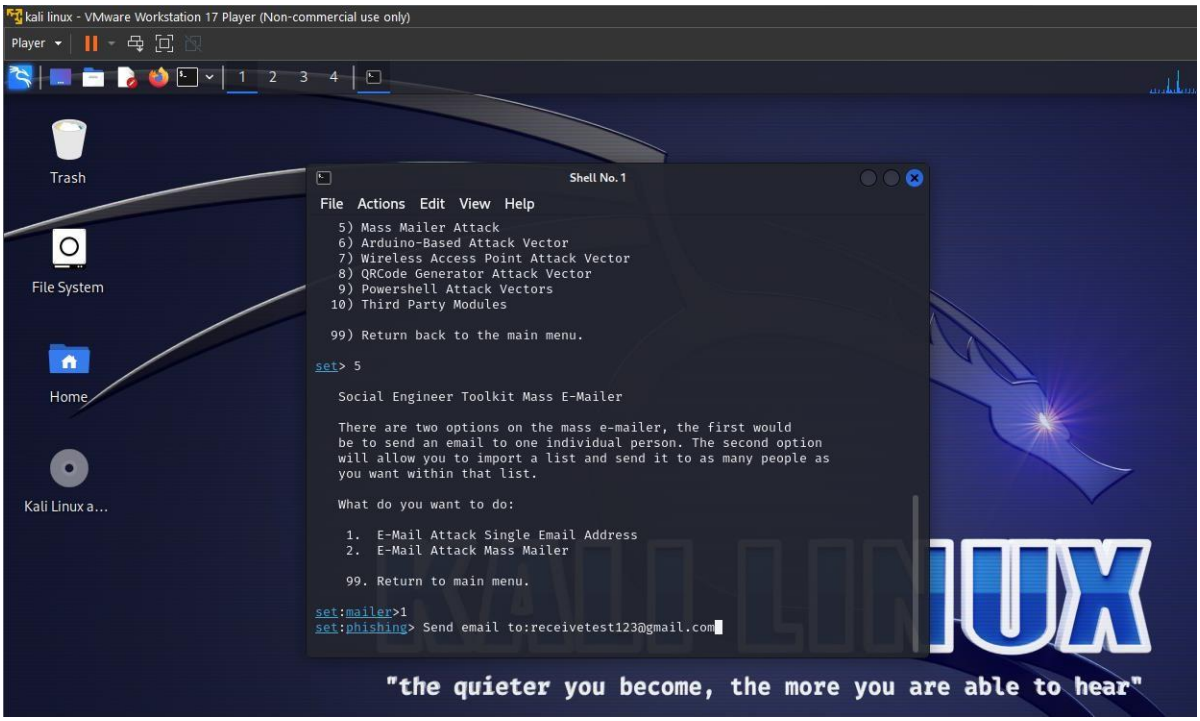

- Open the social engineering tool kit and select option 1 i.e. Social-Engineering attacks



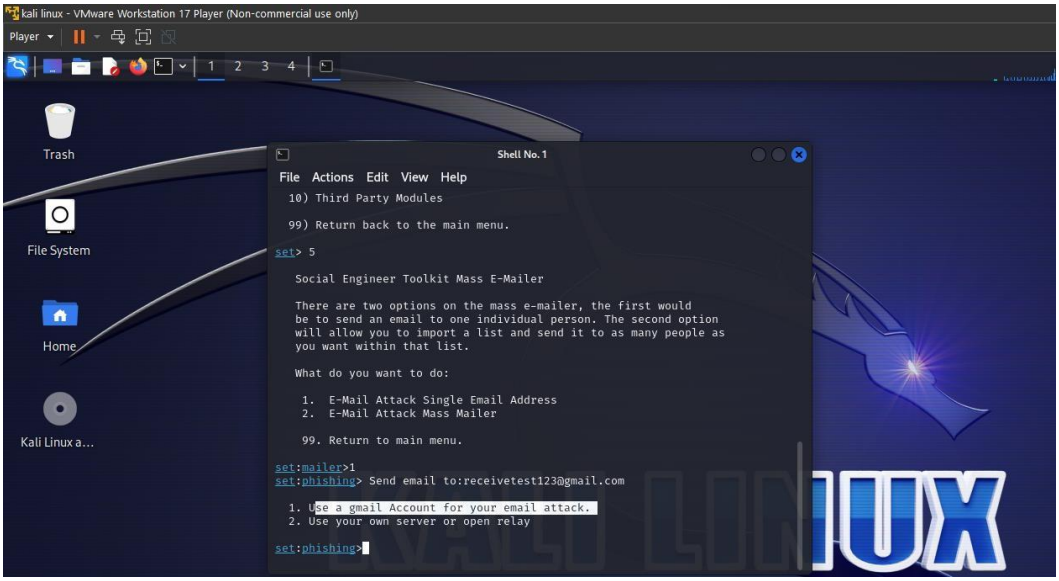
- Now select option 5 i.e. Mass Mailer Attack



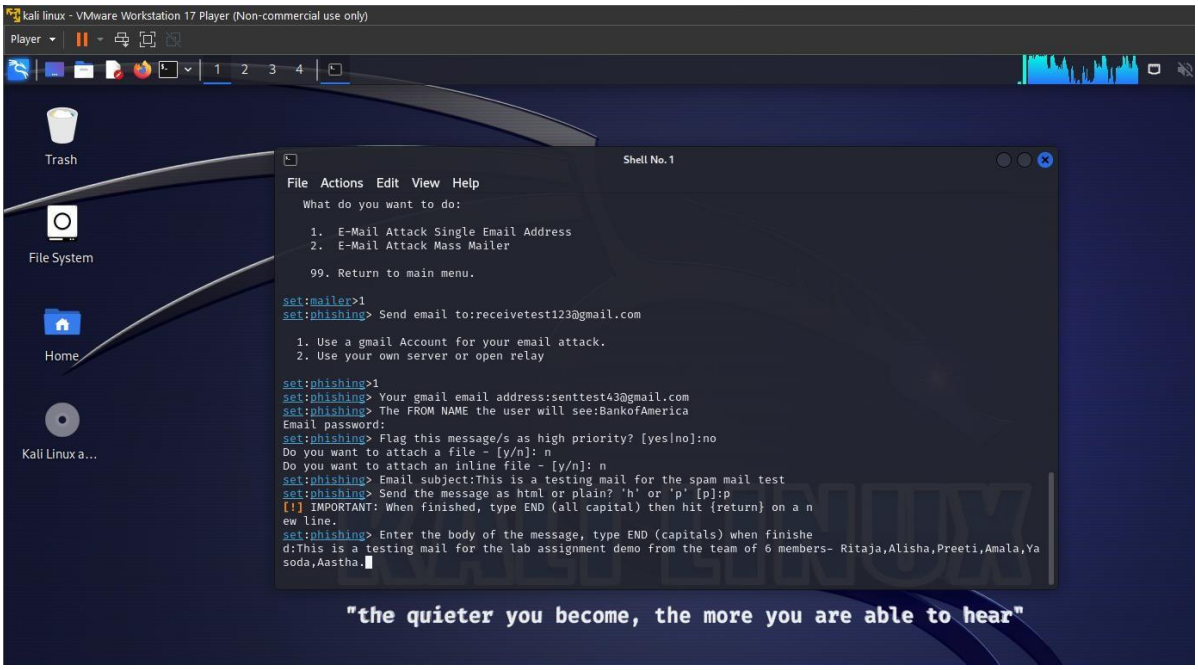

- Now select option 1 i.e. E-mail attack single email address.
- Now it asks for the receiver's email. Give the mail id that has to be received email.



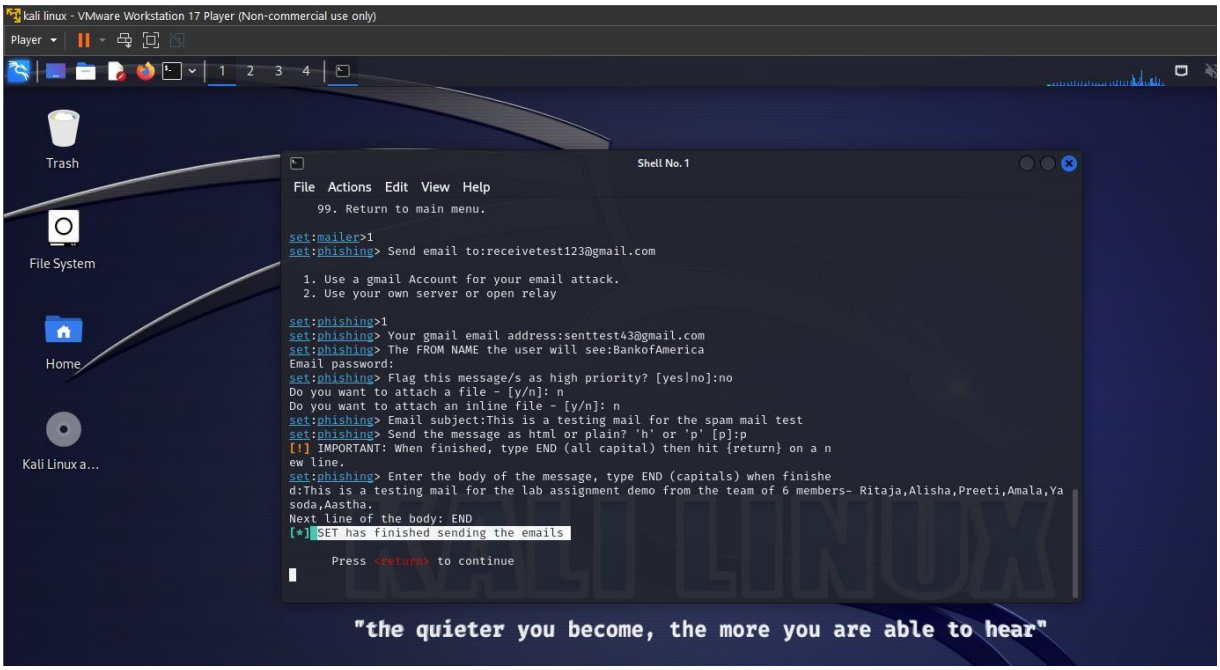
- Now it asks for the mode of email sending. Two options of sending one is using an email address and the other is using a server. Here we selection the option 1 i.e. use a Gmail account for your email attack.



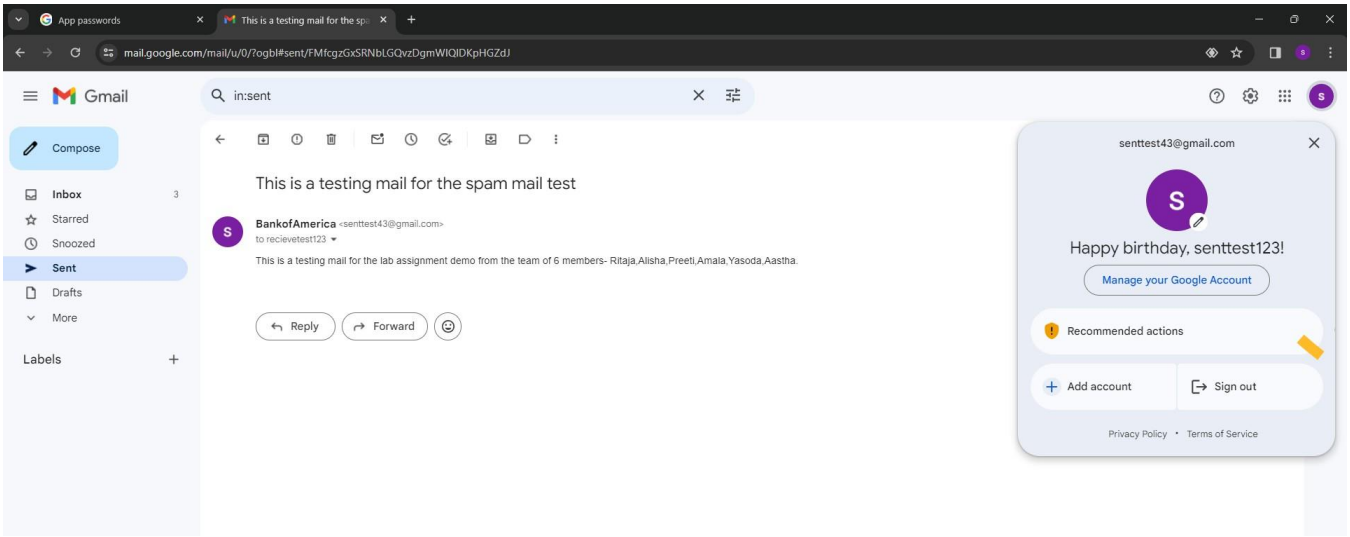

- It asks for the Gmail of the sender.
- It asks for the information that has to be sent in the email.
- They are FROM NAME the user will see, file attachments, Email subject, body of the message.
- It also asks for the option whether the message should be sent in the html format or as a plain.



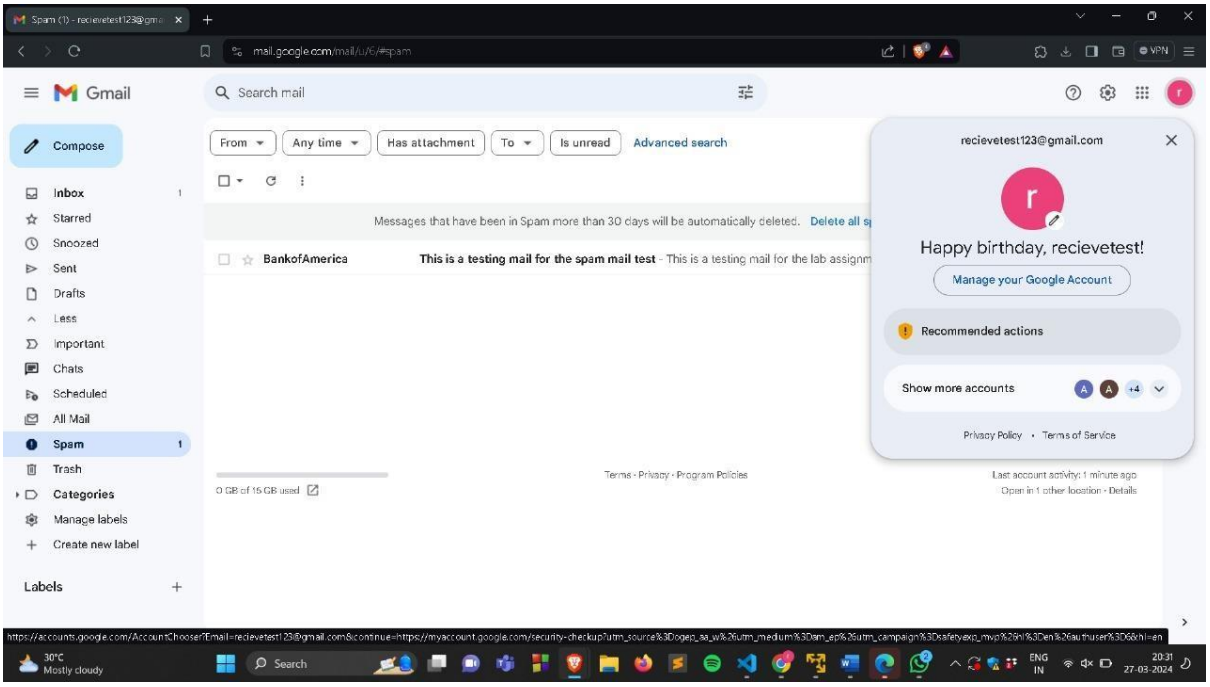
- Type END when you are finished with the body of the message.




- The mail is sent to the receiver email and can be seen in the sent box of the sender email.



- The mail is received to the receivers Gmail and it can be seen in the spam section.




- Here the email is identified as a spam email in the receivers inbox mails.