Project Report

on

PHISHING MAIL DETECTION

Submitted by

Tim Johny (12160086)
Sooraj Eswaradas (12160078)
Sreelakshmi R Syam (12160080)
Sruthy Annie Santhosh (12160081)

In partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

APRIL 2019

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

Certified that this is a bonafide record of the Project titled

PHISHING MAIL DETECTION

done by
Tim Johny (12160086)
Sooraj Eswaradas(12160078)
Sreelakshmi R Syam (12160080)
Sruthy Annie Santhosh (12160081)

of VIII Semester, Computer Science and Engineering in the year 2019 in partial fulfillment requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology.

Dr.Latha R Nair	Preetha S	Dr.Sudheep Elayidom
Head of Division	Project Coordinator	Project Guide

Acknowledgement

We take this opportunity to thank the supreme being, the source of all knowledge whose blessings are our guiding light in any venture we take up. Were in short of words to express our gratitude to Dr. Sudheep Elayidom, our project guide who guided us and helped us constantly with his inputs and suggestion without which we couldnt have implemented this project the way it is working today. We are highly indebted Mrs Preetha S, our project cordinator for their constant supervision and support in completing this project.A bouquet of gratitude to Dr. Latha R Nair, head of the division of Computer Science and Engineering for all kinds of encouragement extended to us.

Tim Johny(12160086)
Sooraj Eswaradas(12160078)
Sreelakshmi R Syam(12160080)
Sruthy Annie Santhosh(12160081)

Declaration

We, Mr. Sooraj Eswaradas , Mr.Tim Johny Ms. Sreelakshmi R Syam, Ms. Sruthy Annie Santhosh, hereby declare that this project is the record of authentic work carried out by us during the academic year 2018 - 2019 and has not been submitted to any other University or Institute towards the award of any degree.

Abstract

Phishing is the combination of social engineering and technical methods to convince the user to reveal their personal data.Phishing is typically carried out by Email spoofing or instant messaging. It targets the user who has no knowledge about social engineering attacks, and internet security, like persons who do not take care of privacy of their accounts details such as Facebook, Gmail, credit banks accounts and other financial accounts.Detecting any phishing website is really a complex and dynamic problem involving many factors and criteria .Because of the ambiguities involved in phishing detection, classification techniques can be an effective tool in detecting phishy mails.In this project we propose a system which detects phishing mails and seperate them from inbox.A mail system is developed and the incoming mails are classified and seperated.For this we use the regression algorithm and decision tree method for classification.Fuzzy logic is also incorporated to classifying the phishing mails which contain phishing URLs that are not yet blacklisted. We are also integrating Phishtank API so that as the blacklists will be updated in the Phishtank it will be easier for our system to detect phishing mails.

Contents

List of Figures

Chapter 1

Introduction

Phishing is a form of fraud in which the attacker tries to learn sensitive information such as login credentials or account information by sending as a reputable entity or person in email or other communication channels. Typically a victim receives a message that appears to have been sent by a known contact or organization. The message contains malicious software targeting the users computer or has links to direct victims to malicious websites in order to trick them into divulging personal and financial information, such as passwords, account IDs or credit card details..

Phishing is popular among attackers, since it is easier to trick someone into clicking a malicious link which seems legitimate than trying to break through a computers defense systems. The malicious links within the body of the message are designed to make it appear that they go to the spoofed organization using that organizations logos and other legitimate contents.The main reason is the lack of awareness of users. But security defenders must take precautions to prevent users from confronting these harmful sites. Preventing these huge costs can start with making people conscious in addition to building strong security mechanisms which are able to detect and prevent phishing domains from reaching the user.

Chapter 2

System Analysis

2.1 Existing System

The already existing system for phishing detection has lot many drawbacks which makes them incompetent in today's world. The false positive rates are too high which is a major concern associated with user operations. Passive warning alone is not enough to control phishing sites. The existing models make use of the Blacklist based approach which becomes futile when a new URL which has not been already blacklisted comes to play.

2.2 Proposed System

This approach intends to amend the phishing attacks at the email level. The main concept is that when a spoofed email is not received by its victims, they cannot fall for the scam. Filters and content analysis techniques are often used to detect phishing emails before they can be delivered to users.For instance, by using training filters, an enormous number of phishing emails can be thwarted.

Normally, phishing is done via sending mails to thousands of users, urging them to visit the fake website through the link or URL present in it. In order to embezzle sensitive information from potential victims, phishers generally try to persuade the users to click on the hyperlink embedded in the spoofed email. LinkGuard works by analyzing the features of the URL. It also calculates the similarities of the URL with the characteristics of known phishing URLs.

Chapter 3

System Study

In this section, we are going to present the SRS, system objectives and hardware and software tool requirements.

3.1 Software Requirements Specification

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

Purpose

The purpose of this project is to detect phishing mails in the mail system.If any mail is found to be containing phishing URLs they are seperated.For classification we are using algorithms like regression and decision tree.There is a site called phishtank which contains a list of blacklisted URLs .We are integrating phishtank API in this system ,so that the suspicious URLs can be compared with the existing blacklisted URLs.A mail system is created.Users can sign in to the mail system.They can send and receive mails.The received mails are checked for malicious URLs. The mails that contain malicious URLs are seperated to another folder.There is also a graphical representation of the amount of non phishing and phishing mails that the user receives.

Project Overview

The project has following functionalities:

Functional Requirements

Functional requirements are associated with specific functions, tasks or behaviors the system must support. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements

- Sign Up
  Purpose: This feature is used to register new users and create accounts.
  Actor: Users.
  Input: name,username,password.
  Output: After these values are stored in the database registration will be successfully completed.

- LogIn
  Purpose: Used by users to log in to the system.
  Actor: Users.
  Input: Username ,password.
  Output: If correct username and password is given then log in is successful.

- Composing Mails
  Purpose: From the user homepage user can send mails.
  Actor: user
  Input: receivers address,subject,message
  Output: Mail will be sent to the receiver.

- Receiving Mails
  Purpose: From the users inbox received mails can be viewed.
  Actor: Users
  Output: User can read the mail.

- Phishing Detection
  Purpose: To detect mails with phishing URLs and classify them as phishing mails.
  Input: URLs from the mail will be fetched automatically.

Output: Phishing mails will be seperated to phishing folder.

Non Functional Requirements

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. In general, functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements".

- Scalability
  The network-deployment cost for scaling up these systems must be manageable merely having the technology to provide a user service is not sufficient. The service-provider involvement requires that different infrastructure services be available. This information helps service providers to determine where to invest next. The data-collection facility is that service want to integrate into their service and system.

- Interoperability
  It is important that the interface is simple and intuitive Instead of making products and services ever more sophisticated, they must be made intuitive, simple, and useful in solving problems

- Reliability
  In order to be more reliable django is used. It enhances the performance as well as the reliability of the system and communicate with more number of people at a time. It is imperative that the service reaches thousands of people, and that it is absolutely reliable.

- Portability
  In order to be more portable we use IDE Pycharm.

- Extensibility
  The application should be widely extensible, where we can include many services like fax can be added in to UMS server. Also many call routing mechanism can be included.

- Efficiency
  The system should function in an efficient manner with proper acknowledgements and responses at high speed.

3.2 Hardware and Software Requirements

Hardware Requirements

- Hard Disk:20 GB and above
- RAM: 256 MB
- Processor speed: 1.6 GHz and above

Software Requirements

- Front end: CSS,Bootstrap
- IDE:Pycharm
- Framework:Django
- Languages:Python
- Database:Sqlite3
- Operating System: Windows

Chapter 4

System Design

4.1 Introduction

Designing requires a careful planning and thinking on the part of the system designer. Designing a system means to plan how the various parts of it are going to achieve the desired goal. After the software requirements have been analysed and specified, design is the first of the three technical activities. Designing, coding and testing are required to build and verify the mobile application.

4.2 Data Flow Diagrams

Data Flow Diagram is a pictorial way of showing the flow of data into/within the system, around the system and out of the system. It is a graphical representation of flow of data within a system. Unlike flowcharts, DFDs do not give detailed descriptions of modules but graphically describe data and how the data interact with the system. The DFD enable us to visualize how the system operates, its final output and the implementation of the system as a whole including modification if any. The purpose of data flow diagram is to provide a semantic bridge between users and system developers.

4.3 LEVEL 0 DFD

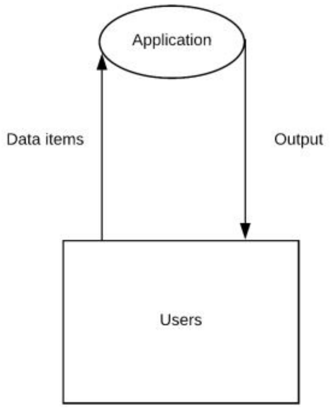


Figure 4.1: LEVEL 0

4.4 LEVEL 1 DFD

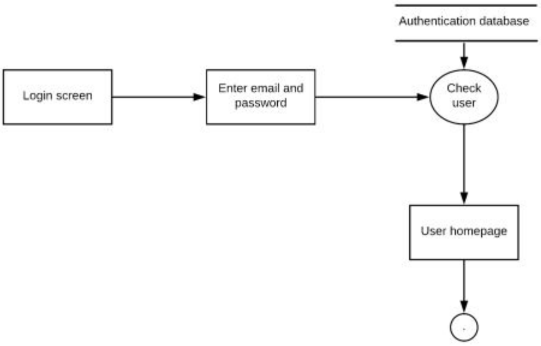


Figure 4.2: LEVEL 1

4.5 LEVEL 2 DFD

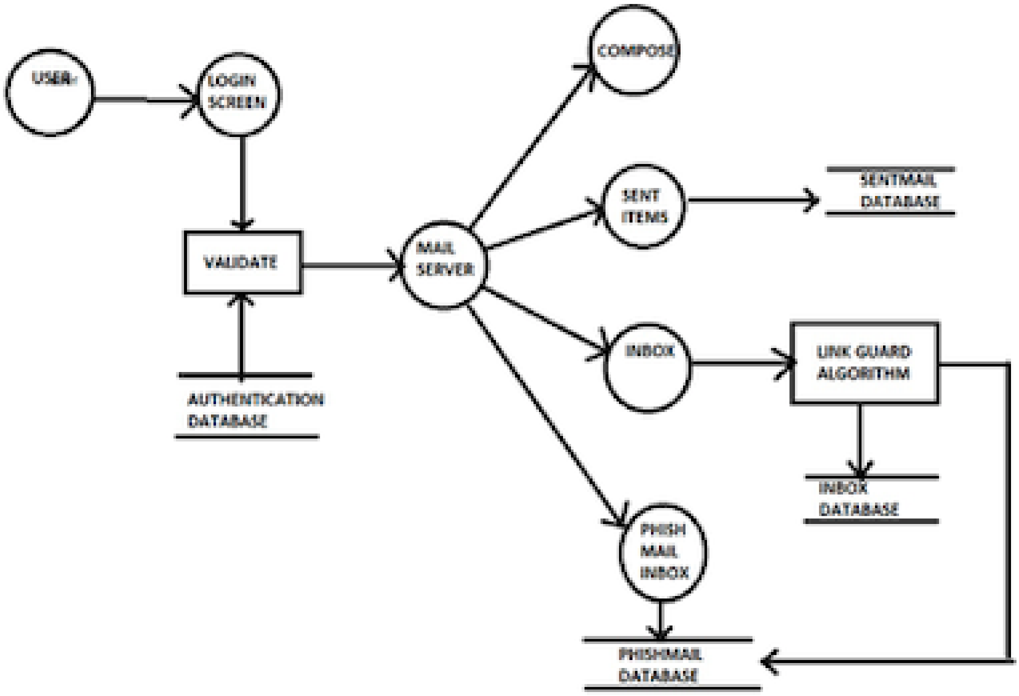


Figure 4.3: LEVEL 2

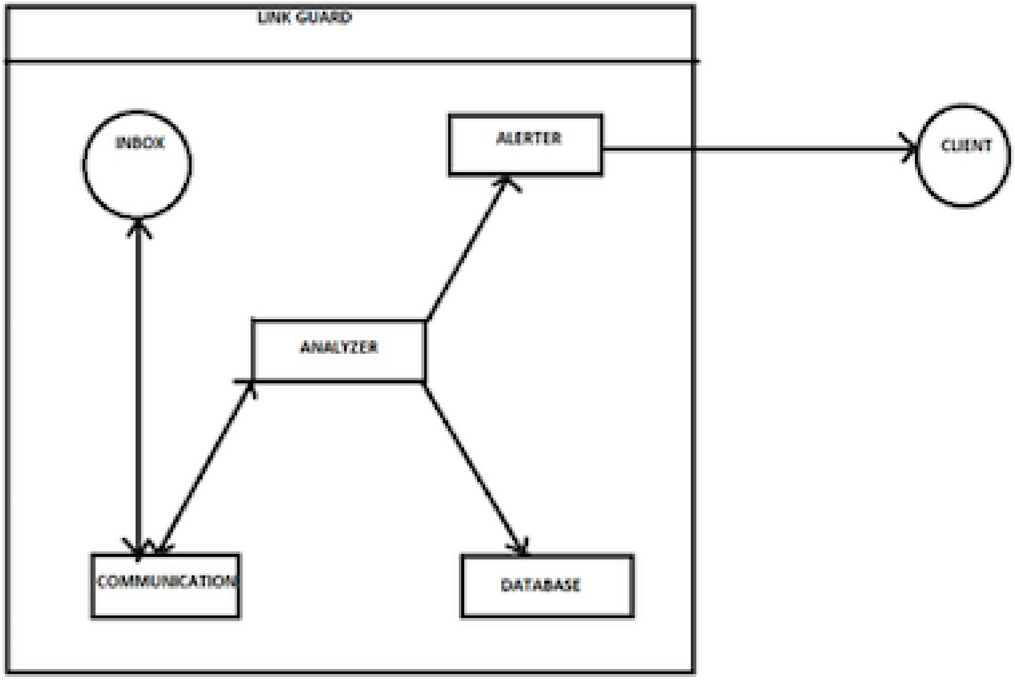


Figure 4.4: LINK GUARD

4.6 Tables Used

The system has the following tables:

- Inbox
- User database
- Phishing database

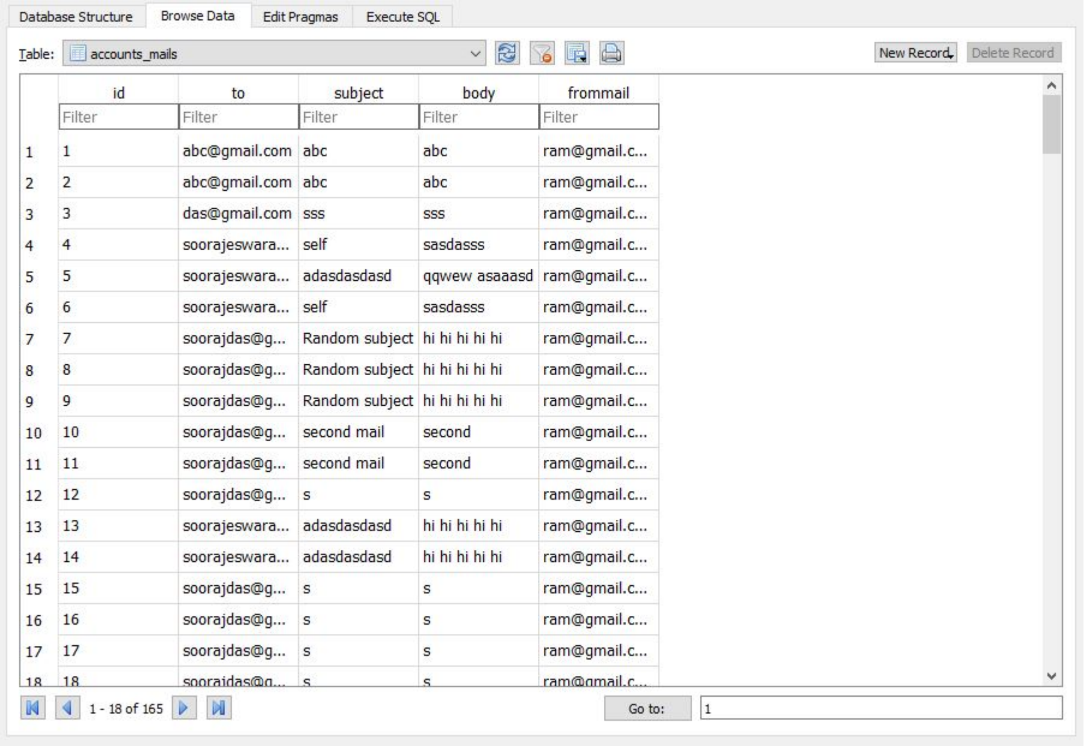


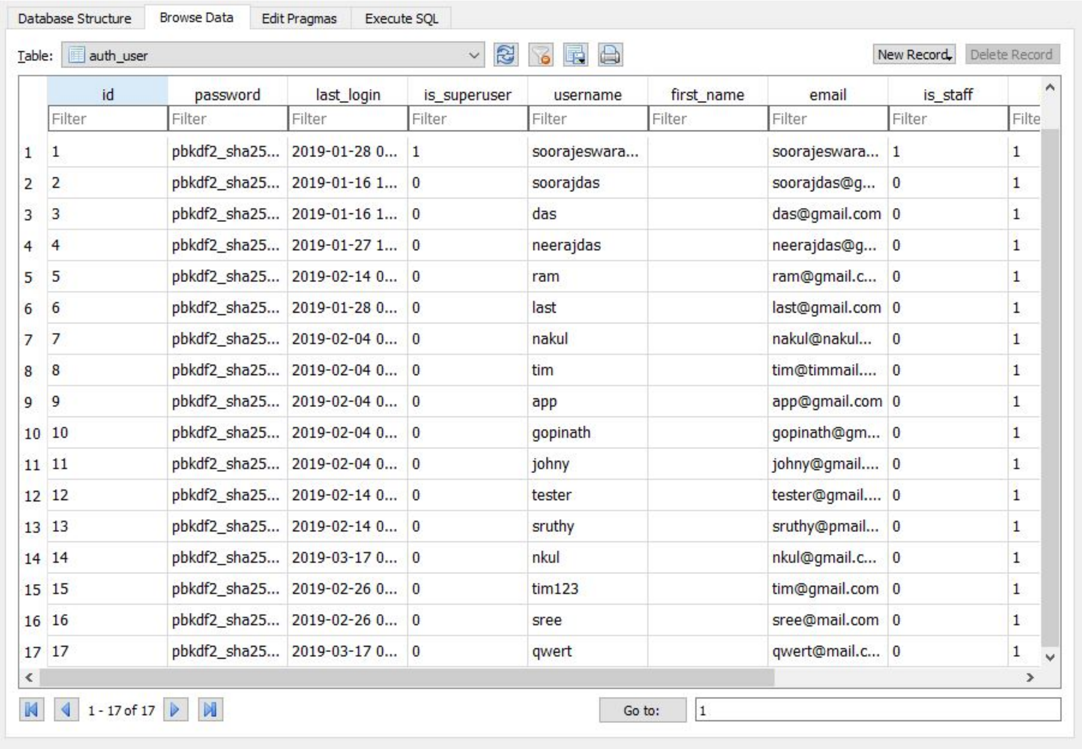
Table: accounts_mails

	id	to	subject	body	frommail
	Filter	Filter	Filter	Filter	Filter
1	1	abc@gmail.com	abc	abc	ram@gmail.c...
2	2	abc@gmail.com	abc	abc	ram@gmail.c...
3	3	das@gmail.com	sss	sss	ram@gmail.c...
4	4	soorajeswara...	self	sasdasss	ram@gmail.c...
5	5	soorajeswara...	adasdasdasd	qqwew asaaasd	ram@gmail.c...
6	6	soorajeswara...	self	sasdasss	ram@gmail.c...
7	7	soorajdas@g...	Random subject	hi hi hi hi hi	ram@gmail.c...
8	8	soorajdas@g...	Random subject	hi hi hi hi hi	ram@gmail.c...
9	9	soorajdas@g...	Random subject	hi hi hi hi hi	ram@gmail.c...
10	10	soorajdas@g...	second mail	second	ram@gmail.c...
11	11	soorajdas@g...	second mail	second	ram@gmail.c...
12	12	soorajdas@g...	s	s	ram@gmail.c...
13	13	soorajeswara...	adasdasdasd	hi hi hi hi hi	ram@gmail.c...
14	14	soorajeswara...	adasdasdasd	hi hi hi hi hi	ram@gmail.c...
15	15	soorajdas@g...	s	s	ram@gmail.c...
16	16	soorajdas@g...	s	s	ram@gmail.c...
17	17	soorajdas@g...	s	s	ram@gmail.c...
18	18	soorajdas@g...	s	s	ram@gmail.c...

1 - 18 of 165

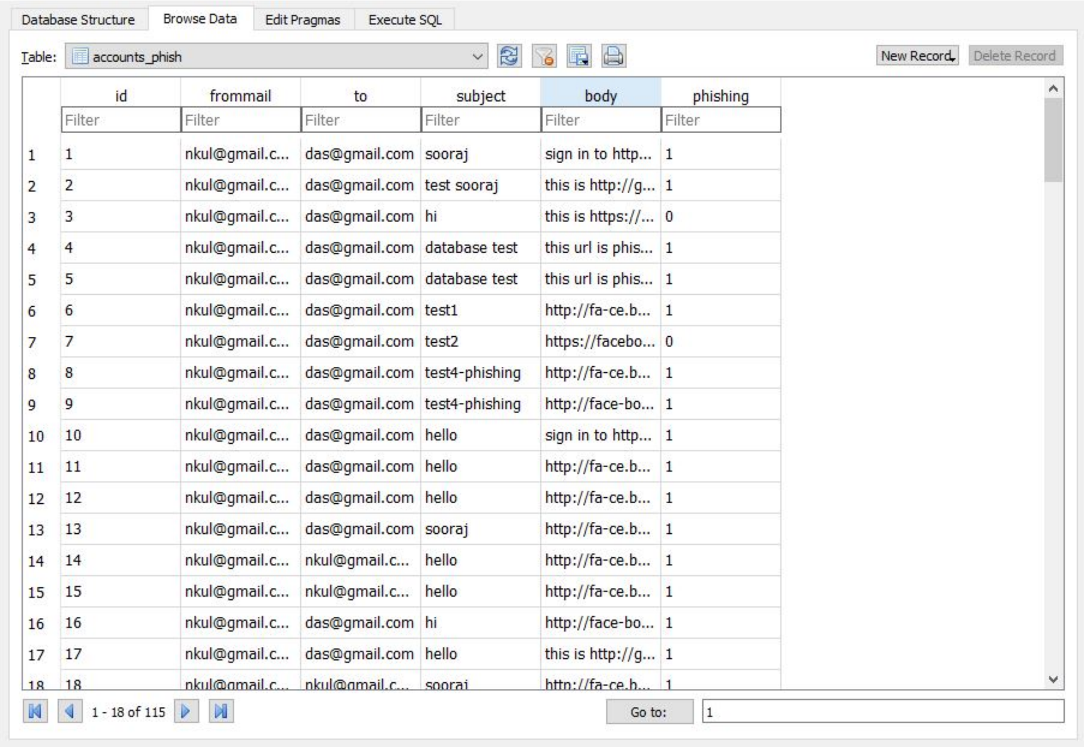
Go to: 1

Figure 4.5: TABLE for incoming mails



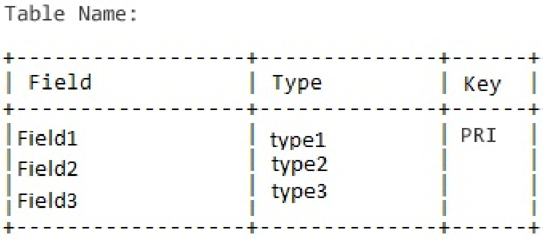
	id	password	last_login	is_superuser	username	first_name	email	is_staff	
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filte
1	1	pbkdf2_sha25...	2019-01-28 0...	1	soorajeswara...		soorajeswara...	1	1
2	2	pbkdf2_sha25...	2019-01-16 1...	0	soorajdas		soorajdas@g...	0	1
3	3	pbkdf2_sha25...	2019-01-16 1...	0	das		das@gmail.com	0	1
4	4	pbkdf2_sha25...	2019-01-27 1...	0	neerajdas		neerajdas@g...	0	1
5	5	pbkdf2_sha25...	2019-02-14 0...	0	ram		ram@gmail.c...	0	1
6	6	pbkdf2_sha25...	2019-01-28 0...	0	last		last@gmail.com	0	1
7	7	pbkdf2_sha25...	2019-02-04 0...	0	nakul		nakul@nakul...	0	1
8	8	pbkdf2_sha25...	2019-02-04 0...	0	tim		tim@timmail....	0	1
9	9	pbkdf2_sha25...	2019-02-04 0...	0	app		app@gmail.com	0	1
10	10	pbkdf2_sha25...	2019-02-04 0...	0	gopinath		gopinath@gm...	0	1
11	11	pbkdf2_sha25...	2019-02-04 0...	0	johny		johny@gmail....	0	1
12	12	pbkdf2_sha25...	2019-02-14 0...	0	tester		tester@gmail....	0	1
13	13	pbkdf2_sha25...	2019-02-14 0...	0	sruthy		sruthy@pmail...	0	1
14	14	pbkdf2_sha25...	2019-03-17 0...	0	nkul		nkul@gmail.c...	0	1
15	15	pbkdf2_sha25...	2019-02-26 0...	0	tim123		tim@gmail.com	0	1
16	16	pbkdf2_sha25...	2019-02-26 0...	0	sree		sree@mail.com	0	1
17	17	pbkdf2_sha25...	2019-03-17 0...	0	qwert		qwert@mail.c...	0	1

Figure 4.6: TABLE FOR USER DATA



	id	frommail	to	subject	body	phishing
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	nkul@gmail.c...	das@gmail.com	sooraj	sign in to http...	1
2	2	nkul@gmail.c...	das@gmail.com	test sooraj	this is http://g...	1
3	3	nkul@gmail.c...	das@gmail.com	hi	this is https://...	0
4	4	nkul@gmail.c...	das@gmail.com	database test	this url is phis...	1
5	5	nkul@gmail.c...	das@gmail.com	database test	this url is phis...	1
6	6	nkul@gmail.c...	das@gmail.com	test1	http://fa-ce.b...	1
7	7	nkul@gmail.c...	das@gmail.com	test2	https://facebo...	0
8	8	nkul@gmail.c...	das@gmail.com	test4-phishing	http://fa-ce.b...	1
9	9	nkul@gmail.c...	das@gmail.com	test4-phishing	http://face-bo...	1
10	10	nkul@gmail.c...	das@gmail.com	hello	sign in to http...	1
11	11	nkul@gmail.c...	das@gmail.com	hello	http://fa-ce.b...	1
12	12	nkul@gmail.c...	das@gmail.com	hello	http://fa-ce.b...	1
13	13	nkul@gmail.c...	das@gmail.com	sooraj	http://fa-ce.b...	1
14	14	nkul@gmail.c...	nkul@gmail.c...	hello	http://fa-ce.b...	1
15	15	nkul@gmail.c...	nkul@gmail.c...	hello	http://fa-ce.b...	1
16	16	nkul@gmail.c...	das@gmail.com	hi	http://face-bo...	1
17	17	nkul@gmail.c...	das@gmail.com	hello	this is http://g...	1
18	18	nkul@gmail.c...	nkul@gmail.c...	sooraj	http://fa-ce.b...	1

Figure 4.7: TABLE FOR DETECTED PHISHING MAILS



```
Table Name:

+------------------+--------------+------+
| Field            | Type         | Key  |
+------------------+--------------+------+
|Field1            | type1        | PRI  |
|Field2            | type2        |      |
|Field3            | type3        |      |
+------------------+--------------+------+
```

Figure 4.8: STRUCTURE OF TABLE

Chapter 5

System Implementation

The project aims giving the user provision to send and receive mails and checks whether the incoming mails have Phishing URLs.If a mail is suspected to have a phishing URL it is classified as phishing mail.Once a user receives an email containing a link,the link is extracted from the mail and is compared with the already present blacklist, if the link is present in the blacklist then the mail is reverted to the phishing mails inbox and will be moved to the phishing database. If its not present in the blacklist then with the help of a Decision Tree Classifier and Logistic Regression we analyze the URL. 8 Characters of a URL are used to identify the genuinity of the URL. The following are the set of charecteristics used:

- Presence of IP address
- URL length
- Presence of @ symbol
- Presence of special characters
- Presence of //
- Subdomain count
- Absence of https at the beginning
- Presence of https in domain

Platform and Tools

The development of this project was done in Windows environment.

Platform

System - Windows is preferred for its user friendliness and vast set of available tools. It provides an easy way to configure a web server to run the application

Database SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. It is very fast and lightweight, and the entire database is stored in a single disk file. It is used in a lot of applications as internal data storage. The Python Standard Library includes a module called "sqlite3" intended for working with this database. .

Tools

Pycharm and django are used to develop.The programming language used is python.

- Pycharm -PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.[5] It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django.Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes.Project and code navigation,specialized project views, file structure views and quick jumping between files, classes, methods and usages,including rename, extract method, introduce variable, introduce constant, pull up, push down and others,Support for web frameworks like Django, web2py and Flask.

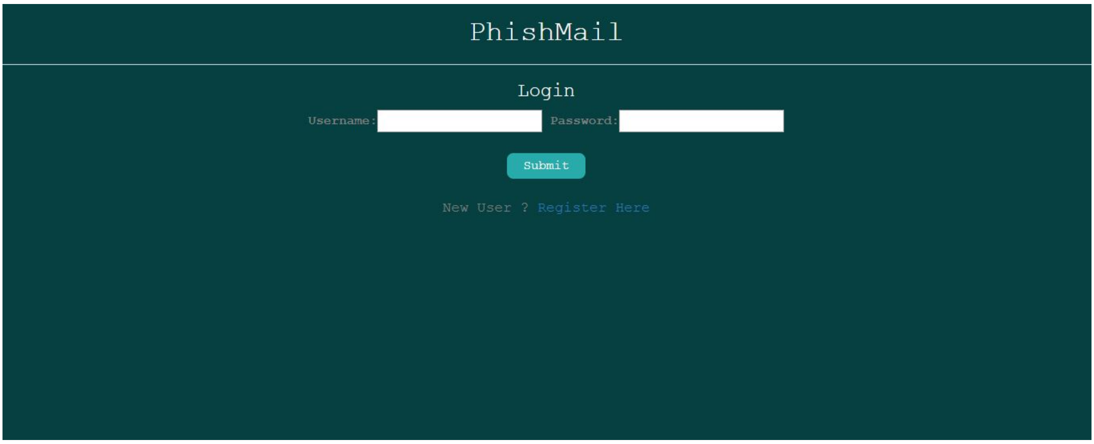
- Django-Django is a Python-based free and open-source web framework, which follows the model-view-template (MVT) architectural pattern. It is maintained by the Django Software Foundation (DSF). Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

- SQLite3- SQLite3 is a very easy to use database engine. It is self-contained, serverless, zero-configuration and transactional. It is very fast and lightweight, and the entire database is stored in a single disk file. It is used in a lot of applications as internal data storage. The Python Standard Library includes a module called ”sqlite3” intended for working with this database.The main engine is written in C, so it should be faster than the gadfly implementation in Python.It’s extensible in a very easy way via Python.It doesn’t put all data in memory like gadfly does (yet you can do that if you want, just use ’:memory:’ as filename.It’s very cool for small databased application, because you do not have to start an external DBMS

- Python-Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum’s vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Screenshots

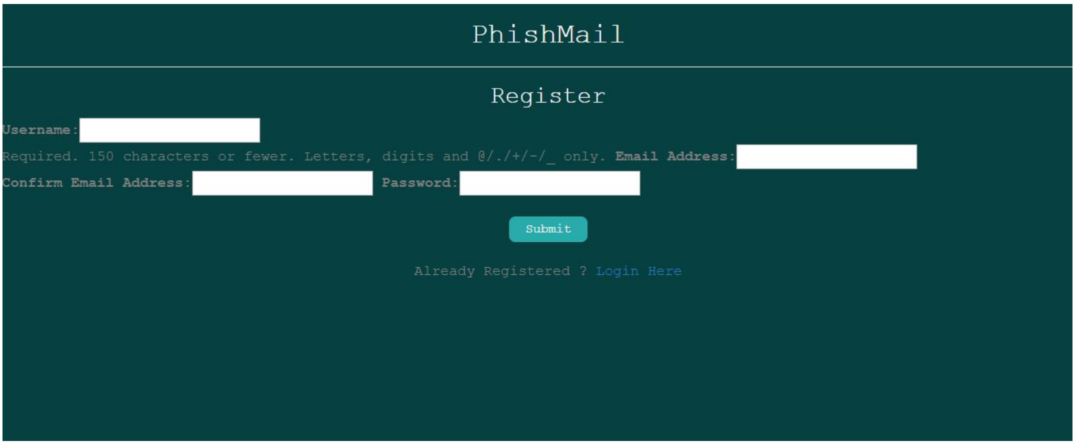
LOGIN PAGE


PhishMail

Login

Username: Password:

Figure 5.1: Screenshot of Login Page



Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Already Registered ? Login Here

Figure 5.2: Screenshot of Registration Page

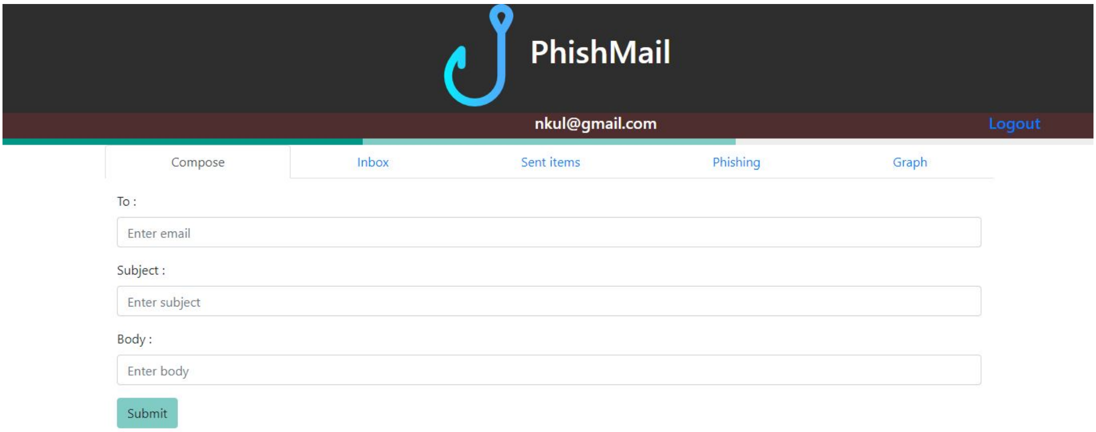
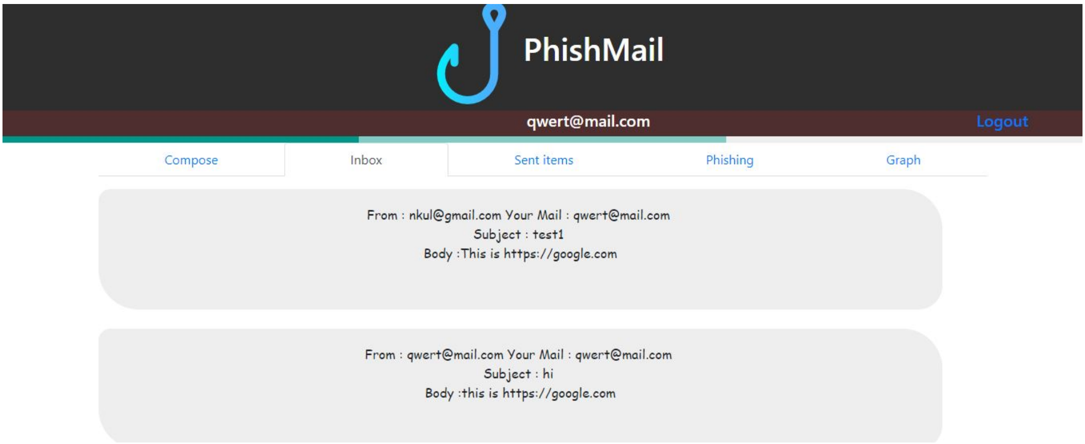


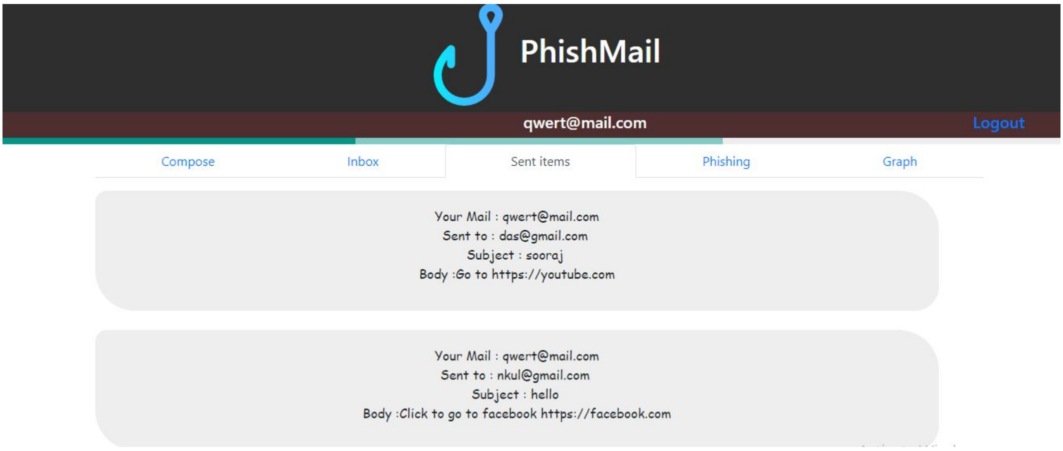
Figure 5.3: Screenshot of Homepage



From : nkul@gmail.com Your Mail : qwert@mail.com
Subject : test1
Body :This is https://google.com

From : qwert@mail.com Your Mail : qwert@mail.com
Subject : hi
Body :this is https://google.com

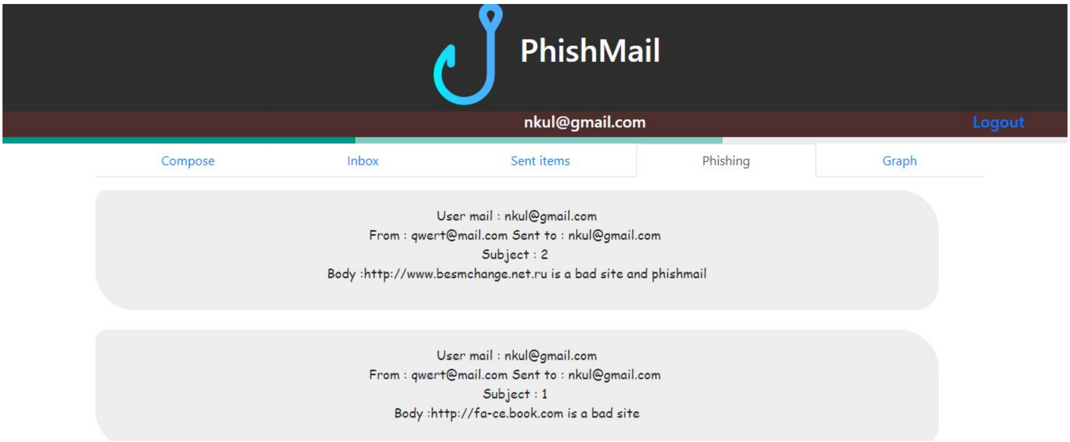
Figure 5.4: Screenshot of Inbox



Your Mail : qwert@mail.com
Sent to : das@gmail.com
Subject : sooraj
Body :Go to https://youtube.com

Your Mail : qwert@mail.com
Sent to : nkul@gmail.com
Subject : hello
Body :Click to go to facebook https://facebook.com

Figure 5.5: Screenshot of Sentitems



User mail : nkul@gmail.com
From : qwert@mail.com Sent to : nkul@gmail.com
Subject : 2
Body :http://www.besmchange.net.ru is a bad site and phishmail

User mail : nkul@gmail.com
From : qwert@mail.com Sent to : nkul@gmail.com
Subject : 1
Body :http://fa-ce.book.com is a bad site

Figure 5.6: Screenshot of Phishing Tab

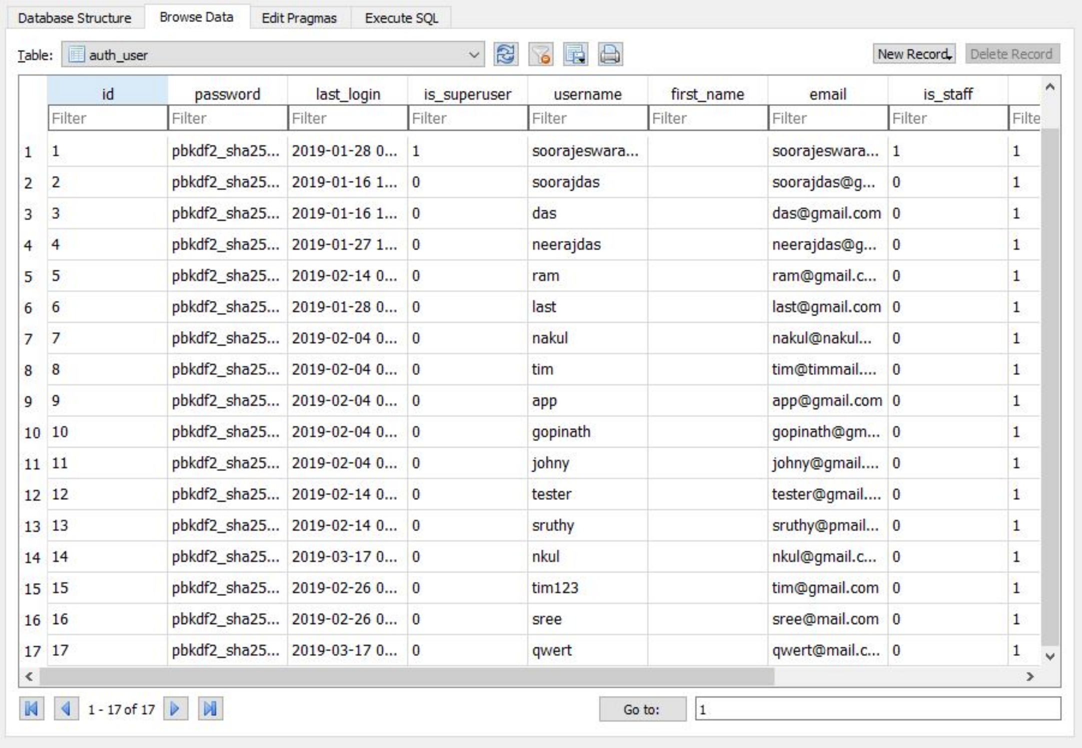


Table: auth_user

	id	password	last_login	is_superuser	username	first_name	email	is_staff	
1	1	pbkdf2_sha25...	2019-01-28 0...	1	soorajeswara...		soorajeswara...	1	1
2	2	pbkdf2_sha25...	2019-01-16 1...	0	soorajdas		soorajdas@g...	0	1
3	3	pbkdf2_sha25...	2019-01-16 1...	0	das		das@gmail.com	0	1
4	4	pbkdf2_sha25...	2019-01-27 1...	0	neerajdas		neerajdas@g...	0	1
5	5	pbkdf2_sha25...	2019-02-14 0...	0	ram		ram@gmail.c...	0	1
6	6	pbkdf2_sha25...	2019-01-28 0...	0	last		last@gmail.com	0	1
7	7	pbkdf2_sha25...	2019-02-04 0...	0	nakul		nakul@nakul...	0	1
8	8	pbkdf2_sha25...	2019-02-04 0...	0	tim		tim@timmail....	0	1
9	9	pbkdf2_sha25...	2019-02-04 0...	0	app		app@gmail.com	0	1
10	10	pbkdf2_sha25...	2019-02-04 0...	0	gopinath		gopinath@gm...	0	1
11	11	pbkdf2_sha25...	2019-02-04 0...	0	johny		johny@gmail....	0	1
12	12	pbkdf2_sha25...	2019-02-14 0...	0	tester		tester@gmail....	0	1
13	13	pbkdf2_sha25...	2019-02-14 0...	0	sruthy		sruthy@pmail...	0	1
14	14	pbkdf2_sha25...	2019-03-17 0...	0	nkul		nkul@gmail.c...	0	1
15	15	pbkdf2_sha25...	2019-02-26 0...	0	tim123		tim@gmail.com	0	1
16	16	pbkdf2_sha25...	2019-02-26 0...	0	sree		sree@mail.com	0	1
17	17	pbkdf2_sha25...	2019-03-17 0...	0	qwert		qwert@mail.c...	0	1

Go to: 1

Figure 5.7: Screenshot of User Database

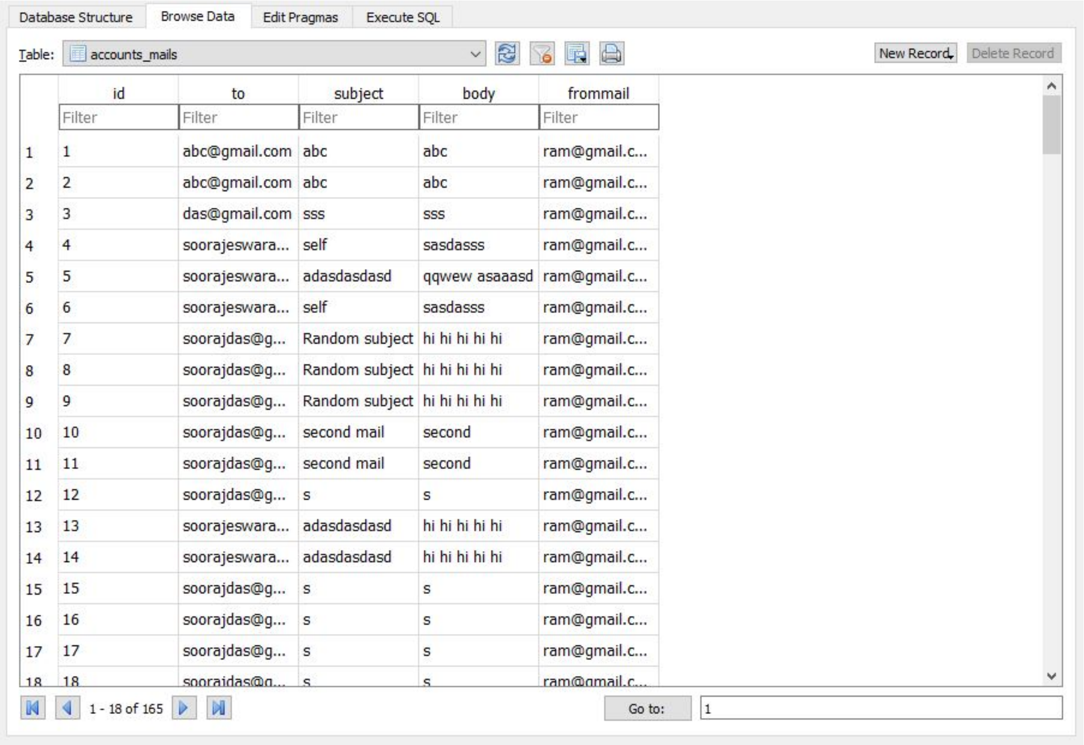


Table: accounts_mails

	id	to	subject	body	frommail
	Filter	Filter	Filter	Filter	Filter
1	1	abc@gmail.com	abc	abc	ram@gmail.c...
2	2	abc@gmail.com	abc	abc	ram@gmail.c...
3	3	das@gmail.com	sss	sss	ram@gmail.c...
4	4	soorajeswara...	self	sasdasss	ram@gmail.c...
5	5	soorajeswara...	adasdasdasd	qqwew asaaasd	ram@gmail.c...
6	6	soorajeswara...	self	sasdasss	ram@gmail.c...
7	7	soorajdas@g...	Random subject	hi hi hi hi hi	ram@gmail.c...
8	8	soorajdas@g...	Random subject	hi hi hi hi hi	ram@gmail.c...
9	9	soorajdas@g...	Random subject	hi hi hi hi hi	ram@gmail.c...
10	10	soorajdas@g...	second mail	second	ram@gmail.c...
11	11	soorajdas@g...	second mail	second	ram@gmail.c...
12	12	soorajdas@g...	s	s	ram@gmail.c...
13	13	soorajeswara...	adasdasdasd	hi hi hi hi hi	ram@gmail.c...
14	14	soorajeswara...	adasdasdasd	hi hi hi hi hi	ram@gmail.c...
15	15	soorajdas@g...	s	s	ram@gmail.c...
16	16	soorajdas@g...	s	s	ram@gmail.c...
17	17	soorajdas@g...	s	s	ram@gmail.c...
18	18	soorajdas@g...	s	s	ram@gmail.c...

1 - 18 of 165

Go to: 1

Figure 5.8: Screenshot of Mail Database

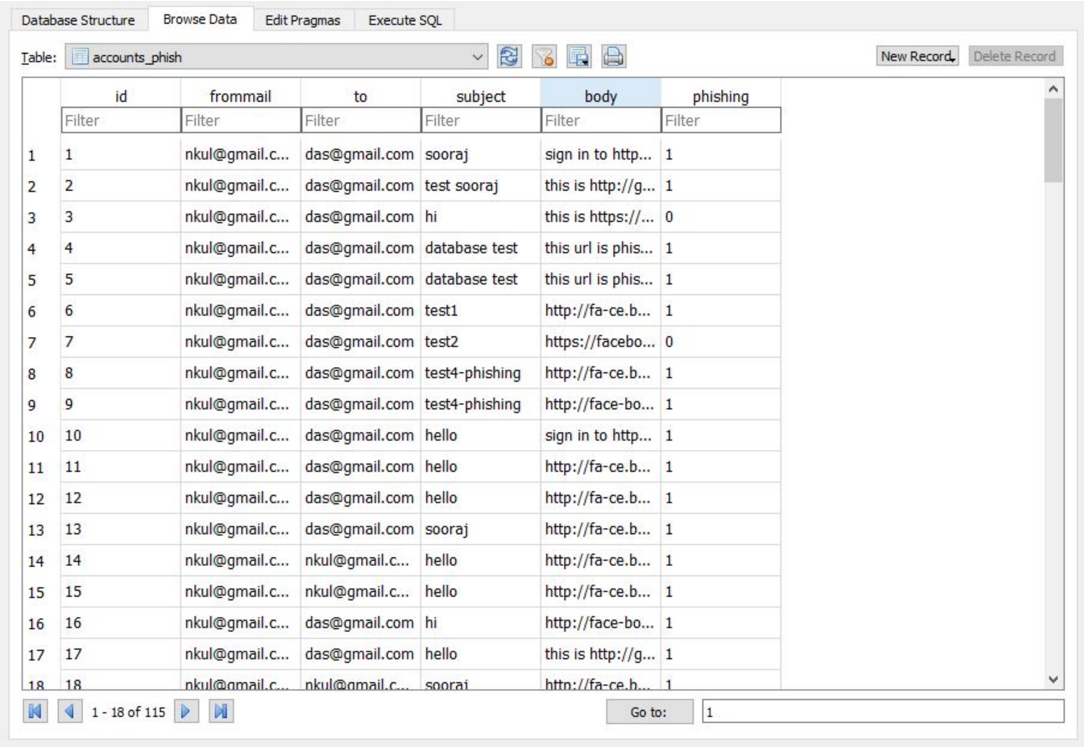


Table: accounts_phish

	id	frommail	to	subject	body	phishing
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	nkul@gmail.c...	das@gmail.com	sooraj	sign in to http...	1
2	2	nkul@gmail.c...	das@gmail.com	test sooraj	this is http://g...	1
3	3	nkul@gmail.c...	das@gmail.com	hi	this is https://...	0
4	4	nkul@gmail.c...	das@gmail.com	database test	this url is phis...	1
5	5	nkul@gmail.c...	das@gmail.com	database test	this url is phis...	1
6	6	nkul@gmail.c...	das@gmail.com	test1	http://fa-ce.b...	1
7	7	nkul@gmail.c...	das@gmail.com	test2	https://facebo...	0
8	8	nkul@gmail.c...	das@gmail.com	test4-phishing	http://fa-ce.b...	1
9	9	nkul@gmail.c...	das@gmail.com	test4-phishing	http://face-bo...	1
10	10	nkul@gmail.c...	das@gmail.com	hello	sign in to http...	1
11	11	nkul@gmail.c...	das@gmail.com	hello	http://fa-ce.b...	1
12	12	nkul@gmail.c...	das@gmail.com	hello	http://fa-ce.b...	1
13	13	nkul@gmail.c...	das@gmail.com	sooraj	http://fa-ce.b...	1
14	14	nkul@gmail.c...	nkul@gmail.c...	hello	http://fa-ce.b...	1
15	15	nkul@gmail.c...	nkul@gmail.c...	hello	http://fa-ce.b...	1
16	16	nkul@gmail.c...	das@gmail.com	hi	http://face-bo...	1
17	17	nkul@gmail.c...	das@gmail.com	hello	this is http://g...	1
18	18	nkul@gmail.c...	nkul@gmail.c...	sooraj	http://fa-ce.b...	1

1 - 18 of 115

Go to: 1

Figure 5.9: Screenshot of Phishing Mail Database

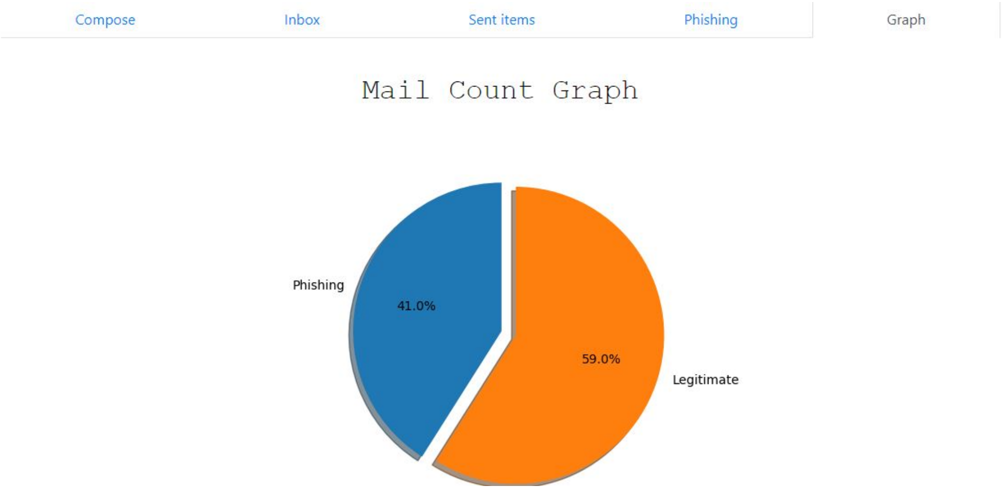


Figure 5.10: Screenshot of Mail Count Graph

Chapter 6

System Testing

The aim of the system testing process was to determine all defects in our project .The program was subjected to a set of test inputs and various observations were made and based on these observations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

Unit testing

Integration testing

6.1 Unit Testing

Unit testing is undertaken when a module has been created and successfully reviewed .In order to test a single module we need to provide a complete environment ie besides the module we would require 1.The procedures belonging to other modules that the module under test calls 2.Non local data structures that module accesses. A procedure to call the functions of the module under test with appropriate parameters Unit testing was done on each and every module.

6.2 Integration Testing

In this type of testing we test various integration of the project module by providing the input.The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module. Once the individual modules were done with testing the various modules were incorporated to a single unit and further testing were carried out till we made sure there wasnt any bugs present

Chapter 7

Result

We have described an approach for classifying URLs automatically as either malicious or benign based on supervised learning across both lexical and host-based features. This approach is complementary to both blacklisting which cannot predict the status of previously unseen URLs. The project aims at providing the user a mail server which detects a mail with a phishing URL .If a recieved mail contains a phishing URL it is oved to the phishing database automatically thus saving the user from further attack. Thus the user will be able to identify the malicious mails.So the project prevents any possible phishing attacks to the user.

Chapter 8

Future Scope

Our classifier was able to utilise only 7 characteristics of a URL to identify whether a mail is genuine or phishy. There are around 30 properties for a URL which can be used to identify the genuinity of a URL. So future endeavours can bring in the use of all 30 features of the URL and thus increase the accuracy and reduce false positive rates.

Beside URL-Based Features, different kinds of features can be used in machine learning algorithms in the detection process such as:

- URL-Based Features
- Domain-Based Features
- Page-Based Features
- Content-Based Features

There is a future scope for this project by adding a ranking wise listing of phishing mails.Like the phishing e-mails can be ranked like highly phishy or less phishy.Highly phishy emails can be automatically deleted also.

Chapter 9

Conclusion

Phishing website is a recent problem nevertheless due to its huge impact on the financial and online retailing sectors and since preventing such attacks is an important step towards defending against e-banking phishing website attacks . Latest studies shows that 1 in 61 mails anyone recieve today is a phishing mail which shows how important it is to prevent a user from falling prey to a phishing attack. One approach is stop phishing at email level. Since most of the current phishing attacks use spam to lure victims to a phishing website.This project detects phishing URLs present in received e mails and thus safeguard user from possible phishing attacks. Blacklisting alone won't help curb the menace of phishing URLs. The use of data mining techniques can help increase the accuracy and false positive rates.

Chapter 10

Reference

[1]Javaria Khalid,Rabiya Jalil ,Myda Khalid,Maliha Maryam Muhammad Aatif ShafiqueWajid Rasheed,Anti-phishing Models for Mobile Application Development: A Review Paper,2018.
[2] Abdulghani Ali Ahmed,Nurul Amirah Abdullah,Realtime detection of Phishing Websites,2016.
[3]Django for begineers by William Vincent,2018.
[4] Fundamentals of software engineering, Rajib Mall, Pearson Education, 2011.