##### PHISHING WEBSITE DETECTOR APP USING ANDROID STUDIO (ADRESS BAR BASED FEATURES)

**A PROJECT REPORT**

###### ***Submitted by***

**Yogendra Bijapur (20BCY10065)**

**R.P. Niranjan Surya Prasad (20BCY10129)**

**Joel Franko (20BCY10104)**

**Rohan Kolhatkar (20BCY10177)**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**CYBER SECURITY AND DIGITAL FORENSICS**

****

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

DECEMBER 2021

**VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“PHISHING WEBSITE DETECTOR APP USING ANDROID STUDIO (ADDRESS BAR BASED FEATURES)”** is the bonafide work of “**YOGENDRA BIJAPUR (20BCY10065), R.P. NIRANJAN SURYA PRASAD (20BCY10129), JOEL FRANKO (20BCY10104), ROHAN KOLHATKAR (20BCY10177)**” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

Dr. R. Rakesh, Assistant Professor Dr. Ganeshan R, Assistant professor

School of Computer Science and Engineering School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on 16/12/2021

**ACKNOWLEDGEMENT**

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr. Shishir Kumar Shandilya, Head of the Department, School of Computer Science and Engineering for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide Dr. Ganeshan R for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **FULL FORM** |
| **IP** | **Internet Protocol** |
| **PHP** | **Hypertext Preprocessor** |
| **HTTP** | **HyperText Transfer Protocol** |
| **URL** | **Uniform Resource Locator** |
| **HTTPS** | **Hypertext Transfer Protocol Secure** |
| **WWW** | **World Wide Web** |
| **XML** | **Extensible Markup Language** |
| **HTML** | **Hyper-Text Markup Language** |
| **.com** | **Dot Commercial** |
| **App** | **Application** |

**LIST OF FIGURES AND GRAPHS**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| **1.** | **Methodology** | **7** |
| **2.** | **Prototype** | **13** |
| **3.** | **Performance Analysis (Graph)** | **14** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO.** | **TITLE** | **PAGE NO.** |
| **1.** | **Result Table** | **14** |

**ABSTRACT**

One of the challenges faced by our research was the unavailability of reliable training datasets. In fact, this challenge faces any researcher in the field. However, although plenty of articles about predicting phishing websites using data mining techniques have been disseminated these days, no reliable training dataset has been published publicly, maybe because there is no agreement in the literature on the definitive features that characterize phishing websites, hence it is difficult to shape a dataset that covers all possible features.

In this report, we shed light on the important features that have proved to be sound and effective in predicting phishing websites. In addition, we proposed some new features, experimentally assign new rules to some well-known features, and update some other features using some research papers as a reference

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | List of Abbreviations  List of Figures and Graphs  List of Tables  Abstract | iii  iv  v  vi |
| 1 | **CHAPTER-1:**  **PROJECT DESCRIPTION AND OUTLINE**  Introduction 1.2 Motivation for the work  1.3 [About Introduction to the project  including techniques]  1.4 Problem Statement  1.5 Objective of the work  1.6 Organization of the project  1.7 Summary | 1  1  1  1-2  2  2  2  3 |
| 2 | **CHAPTER-2:**  **RELATED WORK INVESTIGATION**  2.1 Introduction  2.2 <Core area of the project>  2.3 Existing Approaches/Methods  2.4 <Pros and cons of the stated Approaches/Methods>  2.5 Issues/observations from investigation  2.6 Summary | 4  4  4  4  4-5  5  5 |
| 3 | **CHAPTER-3:**  **REQUIREMENT ARTIFACTS**  3.1 Introduction  3.2 Hardware and Software requirements  3.3 Specific Project requirements  3.3.1 Data requirement  3.3.2 Functions requirement  3.4 Summary | 6  6  6  6  6 |
| 4 | **CHAPTER-4:**  **DESIGN METHODOLOGY AND ITS NOVELTY**  4.1 Methodology and goal  4.2 Functional modules design and analysis  4.3 Software Architectural designs  4.4 Subsystem services  4.5 User Interface designs  4.6 Summary | 7  7  7  7-10  11  11-12  12 |
| 5 | **CHAPTER-5:**  **TECHNICAL IMPLEMENTATION & ANALYSIS**  5.1Outline  5.2 Technical coding and code solutions  5.3 Working Layout of Forms  5.4 Prototype submission  5.5 Test and validation  5.6 Performance Analysis (Graphs/Charts)  5.7 Summary | 13  13  13  13  13  14  14  15 |
| 6 | **CHAPTER-6:**  **PROJECT OUTCOME AND APPLICABILITY**  6.1Outline  6.2 key implementations outline of the System  6.3 Significant project outcomes  6.4 Project applicability on Real-world applications  6.4 Inference | 16  16  16  16  16  16 |
| 7 | **CHAPTER-7:**  **CONCLUSIONS AND RECOMMENDATION**  7.1Outline  7.2 Limitation/Constraints of the System  7.3 Future Enhancements  7.4 Inference | 17  17  17  17  17 |
|  | RELATED WORK INVESTIGATION    References | 18  19 |

**CHAPTER-1**

**PROJECT DESCRIPTION AND OUTLINE**

# 1.1 Introduction:

# Phishing is a very common social engineering attack that is done for the main purpose of stealing important information from the victim. People are tricked by sending malicious links that steal their personal information. Phishing traditionally functions by sending forged e-mail, mimicking an online bank, auction, or payment sites, guiding users to a bogus web page that is carefully designed to look like the login to the genuine site. The project is completely based upon the address bar features for the detection of Phishing links.

1.2 Motivation for the work:

With the increasing reach of the internet, phishing URLs are also on the rise many websites are available that detect phishing links but we want to make an Android app, as most of the internet surfing takes place through smartphones. So, building an Android application for this is the main motivation for the work.

1.3 Introduction to the project including techniques:

The app detects phishing links based upon the conditions of address bar-based features. The app doesn’t even require an internet facility to detect the links. The various address bar features are: -

1. Long URL to Hide the Suspicious Part: To ensure the accuracy of our study, we calculated the length of URLs in the dataset and produced an average URL length. The results showed that if the length of the URL is greater than 75 characters then the URL is classified as phishing.

2. Using the IP Address: If an IP address is used as an alternative to the domain name in the URL, such as “[*http://125.98.3.123/fake.html*](http://125.98.3.123/fake.html)”, users can be sure that someone is trying to steal their personal information.

3. Dots: A secure web-page link contains at most 4 dots. If perhaps, there are more than 4 dots on a web page then it may be recognized as a phishing link. For example [*http://www.website1.com.my/www.phish.com/index.php*](http://www.website1.com.my/www.phish.com/index.php)

4. URLs having “@” Symbol: The phishing URL may include the “@” symbol because the web browser, ignores everything to the left of the @ symbol, therefore, the address [ebay.com@fake-auction.com](mailto:ebay.com@fake-auction.com) would be “fake-auction.com.”

5. Redirecting using “//”: The existence of “//” within the URL path means that the user will be redirected to another website. URL should not contain it more than 2 times.

6. Adding Prefix or Suffix Separated by (-) to the Domain: The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example [*http://www.Confirme-paypal.com/*](http://www.Confirme-paypal.com/)*.*

7. URL using the word ‘tiny’: URL must not contain the word tiny in it. If it contains it might be phishing.

8. The Existence of “HTTPS” Token in the Domain Part of the URL: The phishers may add the “HTTPS” token to the domain part of a URL to trick users. For example <http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/>.

1.4 Problem Statement:

Phishing is one of the most dangerous threats to our important data. According to a recent survey by ESET Israel, 83% of participants said that they had been targeted by phishing messages at least once. Over 57% also said that they had clicked on promotional links without first checking for red flags.

Many such surveys show the severity of the phishing attack. So, it is important to be aware and should have detection for such links.

1.5 Objective of the work:

Our goal in this project is to create a phishing detecting android application which can detect whether the website accessed by a user is a phishing website or not using its URL. Our application is user-friendly and can be used by the user to be protected from ransomware, spam, and other fraudulent activities.

1.6 Organization of the project:

Our Project is organized by our team according to some split-ups of information and data. We gathered information about phishing websites and some ways to identify phishing websites through some research papers. By using the methods to identify the phishing websites we created an android app using android studio and java.

1.7 Summary:

Phishing is a social engineering attack that is used to steal personal information. Of all types of attacks like denial of service, ransomware, malware, etc phishing always stands ahead of all. Also, cybersecurity breaches that occurred in the organization in recent past years are done through phishing which accounts for more than 53% of all. Most phishing attacks are caused due to malicious links which are mainly spammed through emails. So, it is necessary to have a preventive measure against such links by which we can help ourselves to keep our valuable information safe.

**CHAPTER-2**

**RELATED WORK INVESTIGATION**

2.1 Introduction:

In this chapter, we will see about the work-related investigation we have done to develop our project. This chapter will clearly explain the core area of our project, the methods used in our project, and the pros and cons of our project.

2.2 Core area of the project:

The core area of our project is based on phishing websites detected in cybersecurity. cybersecurity or information technology security is the protection of computer systems and networks from information disclosure, theft of, or damage to their hardware, software, or electronic data, as well as from the disruption or misdirection of the services they provide. Phishing websites are created to dupe unsuspecting users into thinking they are on a legitimate site. The criminals will spend a lot of time making the site seem as credible as possible and many sites will appear almost indistinguishable from the real thing. We are trying to prevent this cyber-crime phishing using our project phishing website detector application in android.

2.3 Existing Methods:

At first, we have researched the methods to identify a phishing website using its URL. Then we have got some data to identify a phishing website through address bar features then we implemented it to a code that has an incrementation method if the URL seems a URL of a phishing website, it gives a higher number thus the code considers the website may be phishing. Then we implemented the code to the Android studio and created an android app with a better user interface.

2.4 Pros and cons of the stated Approaches/Methods:

Pros: - There is no android application on the feature that works without the internet. Most phishing detecting websites work online for detection but this app works offline.

Cons: - The application cannot work over other applications, we always have to paste the links there for checking. Also, it only checks for address bar features, there may be a possibility that links don’t have address bar issues but maybe phishing on account of other features like domain-based and JavaScript issues.

2.5 Observations from investigation:

We have observed various online phishing websites detecting applications in windows and many research papers on how we can detect a phishing URL in which the simple method we found to detect a fishing website is to use their URL's so we implemented it to our project.

2.6 Summary:

In this chapter, we have seen about the core area of our project is cyber-crime phishing websites. Then we the methods existing in our project explain the pros and cons of the project which has some cons. Then finally this chapter discusses the observations we had to develop this project.

**CHAPTER-3**

**REQUIREMENT ARTIFACTS**

3.1 Introduction:

We have previously discussed the main idea of the project. Here we can discuss how features need to be implemented in java for making the application.

3.2 Hardware and Software requirements:

The project requires a basic understanding of JAVA for the code implementation of the features and Android studio for making an Android application. Moreover, an Android smartphone is required for the working of the application.

3.3 Specific Project requirements:

Some specific requirements needed for this project are java coding, XML coding, basic android studio, and project algorithm developing skills, and also we need an android device to execute our project.

3.3.1 Data requirement:

We require a set of phishing and non- phishing sites for the detection and finding the output safe or unsafe. Based upon the check conditions it will give us the appropriate outputs.

3.4 Summary:

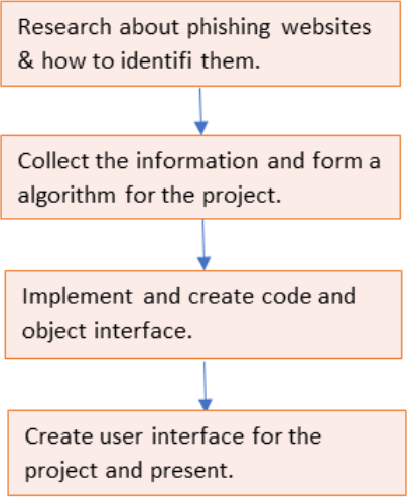
We can summarize as the requirement for the implementations of the code are java and android studio. The application will check for all conditions one by one for each of the inputs and will tell whether it is safe or not.

**CHAPTER-4**

**DESIGN METHODOLOGY AND ITS NOVELTY**

4.1 Methodology and goal:

Methodology:



Goal:

Our main goal is to create a threat-free friendly cyber network using our project as our first step.

4.2 Functional modules design and analysis:

The main functional module used in our project is the value increment system and condition analysis system. They both play an important role in our project to detect the given URL is a phishing URL or not using conditions and give the result using an increment of values.

4.3 Software Architectural designs:

For the development of the app java is used.

package gocy.com;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import java.util.regex.\*;

import android.content.DialogInterface;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

EditText url;

Button button;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

url = findViewById(R.id.url);

button = findViewById(R.id.button);

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String title = "STATUS";

String msg = url.getText()+" is SAFE";

String msg1 = url.getText()+" is PHISHING";

int pc = 0;

int x = 0;

String c1 = "//";

String c2 = "http";

for (int i = 0; i < url.length(); i++) {

if (url.getText().charAt(i) == '.') {

x++;

}

} // For counting number of dots

if (!url.getText().toString().contains("www")) {

pc = pc + 1;

} // To check URL is starting with IP address or not

if (url.length() > 75) {

pc = pc + 1;

} // To check URL length

if (url.getText().toString().contains("-")) {

pc = pc + 1;

} // To check URL is containing "-" or not

if (url.getText().toString().contains("tiny")) {

pc = pc + 1;

} // To check URL is containing "tiny" or not

if (url.getText().toString().contains("@")) {

pc = pc + 1;

} // To check URL is containing "@" or not

if (x >= 4) {

pc = pc + 1;

} // To check URL is not having more then 4 dots

Pattern p = Pattern.compile(c1);

Matcher m = p.matcher(url.getText());

int c=0;

while (m.find())

c++;

if (c>=2){

pc=pc+1;

} // To check URL is not having more then 2 "//"

Pattern p1 = Pattern.compile(c2);

Matcher m1 = p1.matcher(url.getText());

int d=0;

while (m1.find())

d++;

if (d>1){

pc++;

} // To check URL is not having http token as domain

if (pc<3) {

new AlertDialog.Builder(MainActivity.this)

.setTitle(title)

.setMessage(msg)

.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

}

}).create().show();

} // this is used to show the result

if (pc>=3) {

new AlertDialog.Builder(MainActivity.this)

.setTitle(title)

.setMessage(msg1)

.setPositiveButton("Ok", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

dialog.dismiss();

}

}).create().show();

} // this is used to show the result

} });

}

}

4.4 Subsystem services:

The Android Studio is the subsystem service used in our project. It helps us to create a simple user interface android app without much complications or messy codes.

4.5 User Interface designs

For user interface XML is used:

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/b"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="403dp"

android:layout\_height="582dp"

android:layout\_margin="20dp"

android:layout\_marginTop="80dp"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0">

<TextView

android:id="@+id/textView"

style="@style/Widget.MaterialComponents.Snackbar.TextView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:fontFamily="@font/aclonica"

android:gravity="center"

android:shadowColor="#130C0C"

android:text="@string/check\_whether\_the\_url\_is\_safe\_or\_not"

android:textColor="#F1EEEE"

android:textSize="30sp" />

<EditText

android:id="@+id/url"

android:layout\_width="390dp"

android:layout\_height="48dp"

android:layout\_margin="10dp"

android:background="@color/white"

android:ems="10"

android:hint="@string/enter\_url"

android:textColor="#000" />

<Button

android:id="@+id/button"

android:layout\_width="177dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="10dp"

android:layout\_marginTop="40dp"

android:background="@android:drawable/button\_onoff\_indicator\_on"

android:fontFamily="@font/alfa\_slab\_one"

android:gravity="center|center\_horizontal|center\_vertical"

android:text="@string/check"

android:textColor="#121111" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

4.6 Summary:

In this chapter, we have very elaborately discussed about the methodology and goal of our project. Then we have explained the functional modules, we have then discussed the software architecture of our project using JAVA and network design using XML

**CHAPTER-5**

**TECHNICAL IMPLEMENTATION & ANALYSIS**

5.1Outline:

In this chapter will look into the application’s performance; success rates, tests, prototypes, etc.

5.2 Technical coding and code solutions:

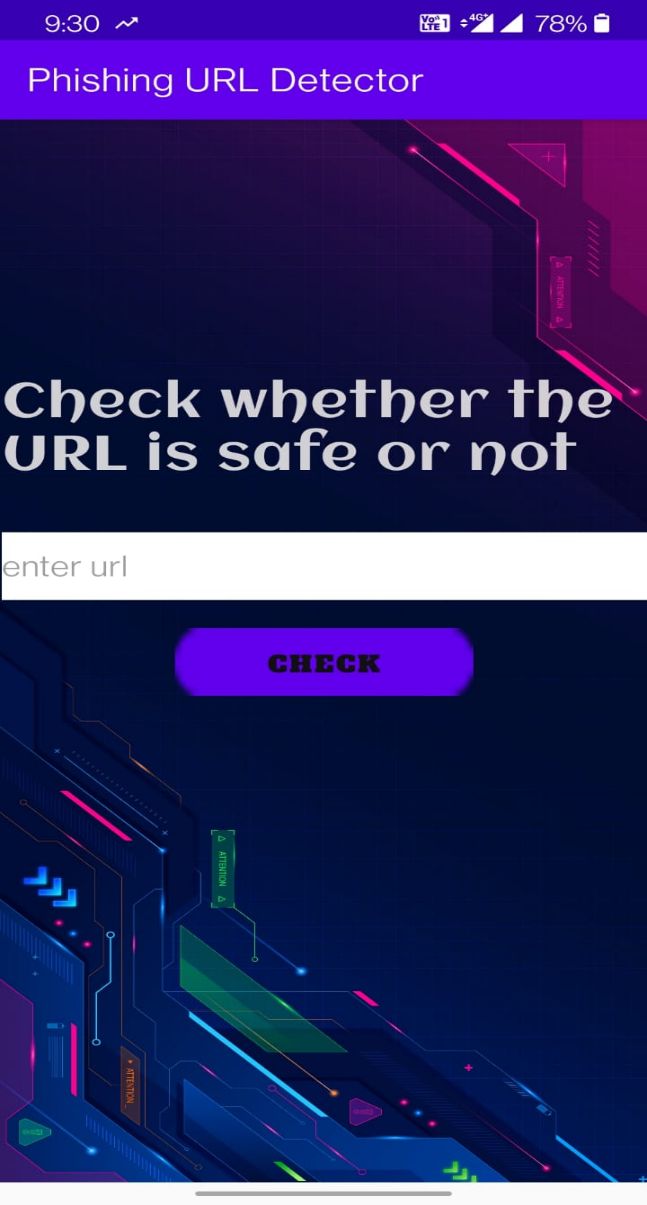
Java and XML coding are used in our project;

5.3 Working Layout of Forms:

We have first found various ways of detecting phishing sites then implemented the most appropriate one in the application. So, the working layout is somewhat like

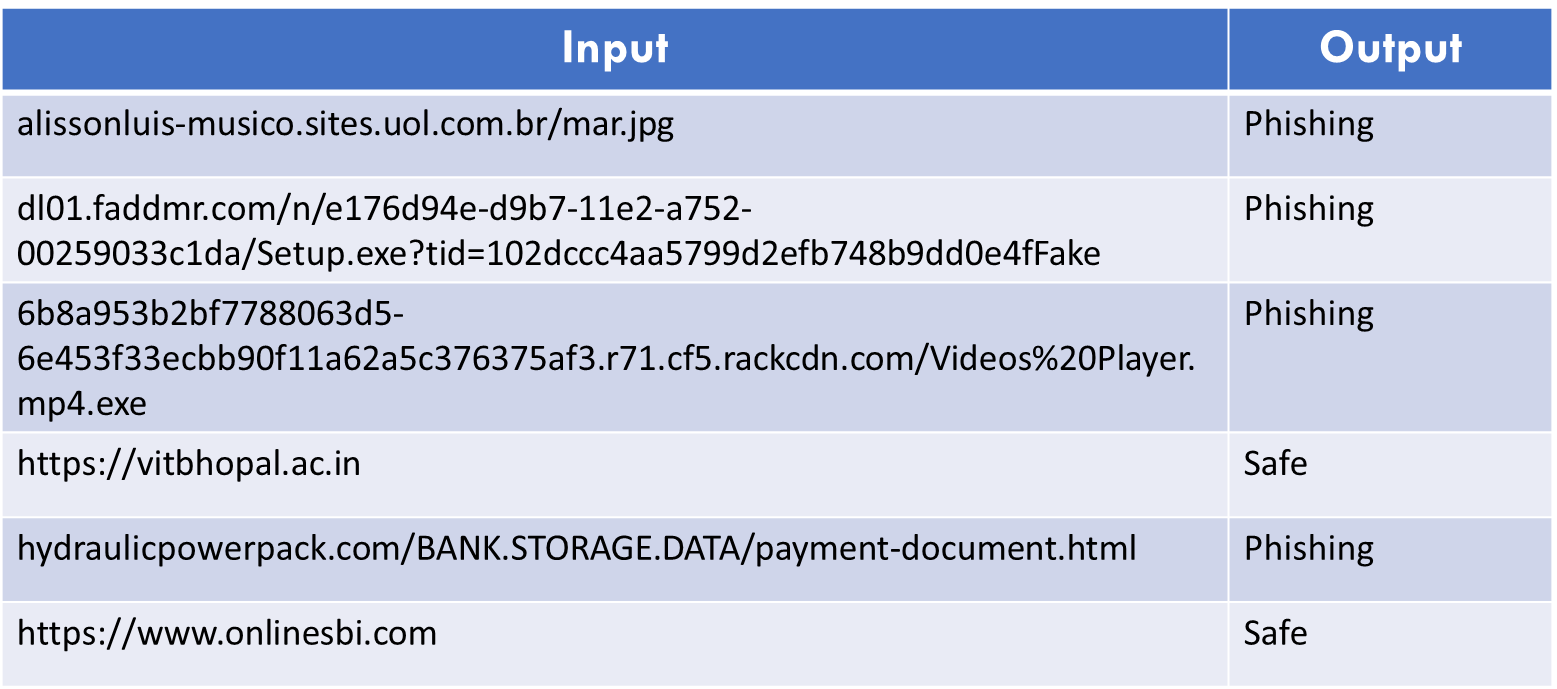
Address bar features -------->implementation in java --------> application

5.4 Prototype submission:



5.5 Test and validation:

We have done various tests by applying various URLs 98% of the time our application detects phishing websites. 2% of phishing websites are missed due to exact copy URLs used by phishing websites. Eg: [www./ping.tapion//japtecon.com](http://www./ping.tapion//japtecon.com).

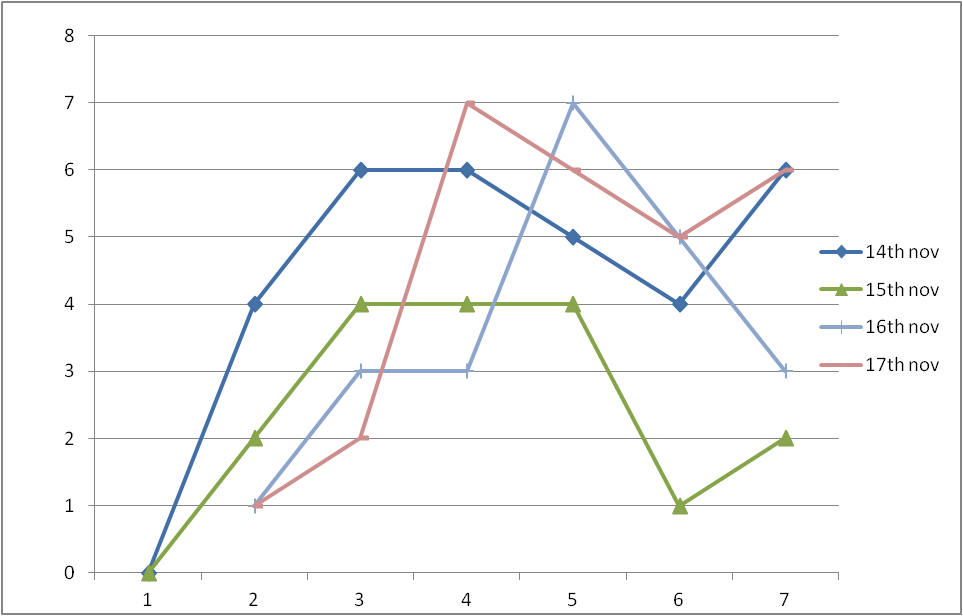


5.6 Performance Analysis (Graphs):

This graph shows the phishing websites detected by our application for 4 days trial.

X-Axis: number of phishing websites detected.

Y-Axis: number of URLs applied for phishing detection.



5.7 Summary:

In this chapter we have explained technical coding, working layout of our project, prototype of our project, then we have shown the test analysis and performance analysis of our project.

**CHAPTER-6**

**PROJECT OUTCOME AND APPLICABILITY**

6.1Outline:

The project is successfully implemented by us. It works well smooth and fine. The outcome is the same as we expected, it successfully detects the links, and tells they are safe or not.

6.2 Key implementations outline of the System:

The main part of the project is the implementation of the code, we found a way we first made a code in python then transformed it into a java code for use in Android studio.

6.3 Significant project outcomes:

The project is successfully implemented it works for all types of links. It successfully detects phishing links based upon the input.

6.4 Project applicability on Real-world applications:

This project is just singly purposed that is it can only detect phishing links. It can be merged into multi-function applications like mobile antivirus programs. It can also be made to detect suspicious links automatically as most desktop antivirus programs do.

6.4 Inference:

Here in the chapter, we discussed how we implemented and also tried for what purpose it can be used in the real world. It can work well as a single-purpose app or can be further enhanced and clubbed with other advanced security applications.

**CHAPTER-7**

**CONCLUSIONS AND RECOMMENDATION**

7.1Outline:

We can conclude that phishing links are not the same as ordinary links, they can be judged and we can find they are phishing or not.

7.2 Limitation/Constraints of the System:

The limitation of phishing detection is that most detectors are not accurate they make sometimes give false results. Also, some may even ask for signing up for detection and the limitation of the project is it works only on address bar features.

7.3 Future Enhancements:

The application is purely based upon some address bar features for the detection of phishing links. We can also move on using many more complex features like domain-based features, Html/JavaScript-based features, and abnormal features which can link more potentially phishing ones. It can be further made more advanced by adding more features like we don’t have to put links for detections it automatically detects the link and shows them in green if they are safe or in red if they are doubtful ones.

7.4 Inference:

As discussed, phishing can be dangerous, so we must have prevention against them. Most phishing links have Address bar-based issues they are not like ordinary links. So, we can infer from this that phishing links are not like ordinary links, they can be judged on various parameters to know whether they are phishing or not.

**RELATED WORK INVESTIGATION**

Anderson, J. R., A. T. Corbett, K. Koedinger, and R. Pelletier in 1995 they have identified Phishing attacks, in which criminals lure Internet users to websites that impersonate legitimate sites, are occurring with increasing frequency and are causing considerable harm to victims. they have described in a paper that the design and evaluation of an embedded training email system that teaches people about phishing during their normal use of email. They have conducted lab experiments contrasting the effectiveness of standard security notices about phishing with two embedded training designs they developed. They found that embedded training works better than the current practice of sending security notices which is used before 1995. They also derived sound design principles for embedded training systems.

**REFERENCES**

1. M. Mohammad School of Computing and Engineering University of Huddersfield Huddersfield, UK.rami.mohammad@hud.ac.uk
2. Fadi Thabtah E-Business Department Canadian University of Dubai Dubai, UAE. fadi@cud.ac.ae
3. Lee McCluskey School of Computing and Engineering University of Huddersfield Huddersfield, UK. t.l.mccluskey@hud.ac.uk
4. <https://www.sciencedirect.com/topics/computer-science/phishing-detection>
5. <https://www.israeldefense.co.il/en/node/52989>