VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgavi-590018, Karnataka, INDIA

PROJECT PHASE-I REPORT

On

"Analysis of Malicious URLs using Machine Learning"

Submitted in partial fulfillment of the requirements for the VII Semester

Subject Code: 17CSP78

Bachelor of Engineering

In

COMPUTER SCIENCE AND ENGINEERING

For the Academic year

2020-2021

BY

Ankita Aditya	**1PE17CS023**
Atul K Uchil	**1PE17CS029**
Mitali Singh	**1PE17CS182**
Prescilla Angel	**1PE18CS419**

Under The Guidance Of

Prof. Sudeepa Roy Dey

Assistant Professor

Dept of Computer Science & Engineering, PESIT-BSC.

Department of Computer Science and Engineering

PESIT Bangalore South Campus

Hosur Road, Bengaluru-560100

PESIT Bangalore South Campus

Hosur Road, Bengaluru-560100

Department of Computer Science and Engineering

CERTIFICATE – PROJECT PHASE-I

This is to certify that the Project Phase-I Subject Code: 17CSP78 work entitled ***"Analysis of Malicious URLs using Machine Learning"*** *is a bona fide work carried out by* ***Ankita Aditya, Atul K Uchil, Mitali Singh and Prescilla Angel*** *and bearing USNs* ***1PE17CS023, 1PE17CS029, 1PE17CS182 and 1PE18CS419*** *respectively in partial fulfillment for the award of Degree of Bachelors (Bachelors of Engineering) in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during seventh semester for the academic year 2020-21.*

It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the Report. The Project Phase-I report has been approved as it satisfies the academic requirements in respect of Project Phase –I work prescribed for said degree.

Signatures:

Project Guide	Head Dept. of CSE
Prof. Sudeepa Roy Dey	**Dr. Annapurna D**
Assistant Professor	PESIT-BSC, Bengaluru
Dept. of CSE	
PESIT-BSC, Bengaluru	

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our effort with success.

We are indebted to our Guide, Prof. Sudeepa Roy Dey, Assistant Professor, Department of Computer Science and Engineering. PESIT - Bangalore South Campus, who has not only coordinated our work but also given suggestions from time to time.

We are greatly thankful to Dr. Annapurna D, Professor and HOD, Department of Computer Science and Engineering, PESIT Bangalore South Campus, for her able guidance, regular source of encouragement, and assistance throughout this project.

We would like to express our immense gratitude to Dr. Subhash Kulkarni, Principal, PESIT Bangalore South Campus, for providing excellent infrastructure.

We gratefully acknowledge the help lent out to us by all faculty members of the Department of Computer Science and Engineering, PESIT Bangalore South Campus, at all difficult times. We would also take this opportunity to thank our college management for the facilities provided. Furthermore, we acknowledge the support and feedback of our parents and friends.

Ankita Aditya
Atul K Uchil
Mitali Singh
Prescilla Angel

ABSTRACT

Nowadays, the WEB has become the highest priority issue in the field of cybersecurity, giving a platform for various online criminal activities. Massive online social networks like Facebook, Twitter, etc. with hundreds of millions of active users are increasingly being used by Cybercriminals to spread malicious URLs, which exploit vulnerabilities on the user's machine for personal gain. URLs are used as the main vehicle in this domain. To tackle this major issue, the community is mainly focused on techniques for blacklisting the malicious URLs. The detection of malicious URLs is one of the highest priority issues for cybersecurity practitioners. There are plenty of machine learning techniques to address this issue, the most used approach remains blacklisting. The main obstacle to using machine learning is the difficulties in data collection. We are building a model that can overcome the obstacles and create a proactive system for the effective and efficient detection of malicious URLs.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Malicious Web sites are a cornerstone of Internet criminal activities. As a result, there has been broad interest in developing systems to prevent the end-user from visiting such sites. The main notion of our project is to identify malicious URLs based on certain lexical features. The project model analyzes the lexical-based feature of malicious URLs. It shows that lexical analysis is effective and efficient for the proactive detection of malicious URLs. We often click some URLs and after visiting the page, we get to know that it was a malicious URL. To avoid this issue, we are incorporating the static model as a chrome extension, wherein users can add the extension to their system. Once the extension is added, and if the user clicks on the malicious URL, the extension throws an alert message that the URL is malicious. This helps the user to know prior that whether the URL, he/she is visiting is malicious or safe to proceed.

1.1.1 Purpose of the project

There are plenty of machine learning techniques to address this issue, the most used approach remains blacklisting but, has the following limitations:

1. The main obstacle to using machine learning is the difficulties in data collection.
2. At present, there is no chrome extension for the real-time detection of malicious websites on a browser.

We are building a model that can overcome the obstacles and create a proactive system for the effective and efficient detection of malicious URLs.

1.1.2 Scope

Malicious Web sites are a cornerstone of Internet criminal activities. As a result, there has been broad interest in developing systems to prevent the end-user from visiting such sites.

This project aims to provide a solution by providing a chrome extension for the detection of malicious urls and alerting the users from the damage.

1.1.3 Definitions, Acronyms and Abbreviations

- Cybersecurity is the protection of internet-connected systems such as hardware, software and data from cyber-threats. The practice is used by individuals and enterprises to protect against unauthorized access to data centers and other computerized systems.
- A malicious URL is a link created with the purpose of promoting scams, attacks and frauds. By clicking on an infected URL, you can download a malware or a Trojan that can take your devices, or you can be persuaded to provide sensitive information on a fake website.
- Lexical features such as URL length, domain length, and the number of special characters (e.g., "%", "$") are widely used to identify malicious URLs.
- Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean/average prediction of the individual trees.
- Google Chrome extensions are programs that can be installed into Chrome in order to change the browser's functionality. This includes adding new features to Chrome or modifying the existing behavior of the program itself to make it more convenient for the user.

1.2 Literature Survey

The proposed methodology, not only takes care of the syntactical nature of URL, but also the semantic and lexical meaning of dynamically changing URLs.

The malicious URLs detection is treated as a binary classification problem and performance of several well-known classifiers are tested with test data. These algorithms are used for training the dataset for classification of good and bad URLs.The detection of malicious URLs is a binary classification problem and several Machine Learning Algorithms, namely Random Forests, SVMs and Naive Bayes are implemented on training dataset. Also, it has been seen that the Random Forest classifier performs better for the particular problem than other classification algorithms.

Another paper proposed a method to overcome the problem of malicious URLs victimizing users. They propose a tool deployed as a chrome extension. The major objective of the proposed model is to aid the users to avoid becoming a victim of malicious and fraudulent activities like malicious URLs, phishing, and social engineering that favor social media as their target medium by detecting them accurately.

1.3 Existing System

The World Wide Web supports a wide range of criminal activities such as spam-advertised e-commerce, financial fraud and malware dissemination. Although the precise motivations behind these schemes may differ, the common denominator lies in the fact that unsuspecting users visit their sites. These visits can be driven by email, web search results or links from other web pages. In all cases, however, the user is required to take some action, such as clicking on a desired Uniform Resource Locator (URL). In order to identify these malicious sites, the web security community has developed blacklisting services. These blacklists are in turn constructed by an array of techniques including manual reporting, honeypots, and web crawlers combined with site analysis heuristics. Inevitably, many malicious sites are not blacklisted either because they are too recent or were never or incorrectly evaluated.

1.4 Proposed System

The main notion of our project is to identify malicious URLs based on certain lexical features. The project model analyzes the lexical-based feature of malicious URLs. It shows that lexical analysis is effective and efficient for the proactive detection of malicious URLs. We often click some URLs and after visiting the page, we get to know that it was a malicious URL. To avoid this issue, we are incorporating the static model as a chrome extension, wherein users can add the extension to their system. Once the extension is added, and if the user clicks on the malicious URL, the extension throws an alert message that the URL is malicious.

1.5 Statement of Problem

Malicious URL detection techniques do suffer low detection accuracy and high false alarm. Besides, the most common technique used, blacklist-based method is inefficient in responding to malware attacks since registering new domain has become easier, no comprehensive blacklist can ensure a perfect up-to-date database.The solution that we are providing uses real-time detection and analysis of Malicious URLs using Machine Learning Techniques and alerts users when they click on a malicious URL.

CHAPTER 2

Software Requirement Specifications

2.1 Software Requirements Specifications

A software requirements specification (SRS) is a description of a software system to be developed. The software requirements specification lays out the functional and the non-functional requirements of the system. It is modelled after business requirements specification, also known as a Stakeholder Requirements Specification (StRS). Used appropriately, software requirements specifications can help prevent software project failure.

2.2. Operating Environment

This section gives a brief overview of the operating environment in which the system will run, and the hardware and the software prerequisites of the project.

2.2.1. Hardware Requirements

- Any suitable computer with an active internet connection.
- RAM – 4 GB.
- Processor - Intel I5 (7th Gen) or higher
- Hard disk - 256 GB or higher
- Dedicated RAM - minimum 4GB

2.2.2. Software Requirements

- Browser - Firefox,Chrome
- OS - Windows 7/8/10/Linux
- Programming Language - Python 3.7 or lower (since GPU support is not available to the newer version at the time of the preparation of this report).
- Documentation Tool – Microsoft Word and Google Docs
- Google Collab

2.3. Functional Requirements

- Chrome extension should be usable to a developer as an extension from the browser.
- Training of the system has to be done with a stable Machine Learning model, for synthesising new URLs based on their Lexical features.
- Balance the trained model for accepting different types of URLs irrespective of URL length, token counts, etc.
- Extraction of lexical features of the URLs acceptable to model.
- Output should provide an option to the users to choose whether they want to continue to the malicious URL detected or want to redirect back with safety.

2.4. Non-functional requirements

- **Usability**: Degree to which a product or system can be used by specified users to achieve specific goals with effectiveness, efficiency, and satisfaction in a specified context of use.
- **Security**: Degree to which a system protects information and data, so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.
- **Performance**: The time for detecting the malicious URLs and alerting the user.
- **Availability**: Describes how likely the system is accessible for a user at a given point in time.

CHAPTER 3

HIGH LEVEL DESIGN

3.1 High Level Design

High-level design (HLD) explains the architecture that will be used for developing this project. The architecture diagram provides an overview of the entire system, identifying the main components that would be developed and their interfaces. The HLD uses non-technical to mildly technical terms that will be understandable to the user. In contrast, low-level design further exposes the logical detailed design of each of these elements for programmers or developers using our API. High-level design usually includes a high level architecture diagram depicting the major modules and interfaces in the system.

3.2 Design Considerations

This section describes the issues which need to be addressed or resolved before attempting to devise a complete design solution.

3.2.1 Assumptions and Dependencies

All the different types of unsafe URLs like defacement URLs, phishing URLs, malwares, etc are categorized under a singleton class that is malicious URLs. Similarly all the safe URLs are under the benign URL category. The only dependency for the entire model is the lexical features of the URLs which plays a deciding factor in getting the target output and categorizing it under malicious or safe category.

3.2.2 Goals and constraints

The main goal of the project is to provide users a hassle-free experience in detecting whether the URL to which they are navigating is safe or not in real-time. Users can easily add the chrome extension and this extension will provide real-time detection of malicious URLs. The main constraint here is the storage. Each time a new URL is generated, all the lexical features associated with it must get stored, as it contributes to model training and detecting the safe and unsafe links.

3.3 System Architecture

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

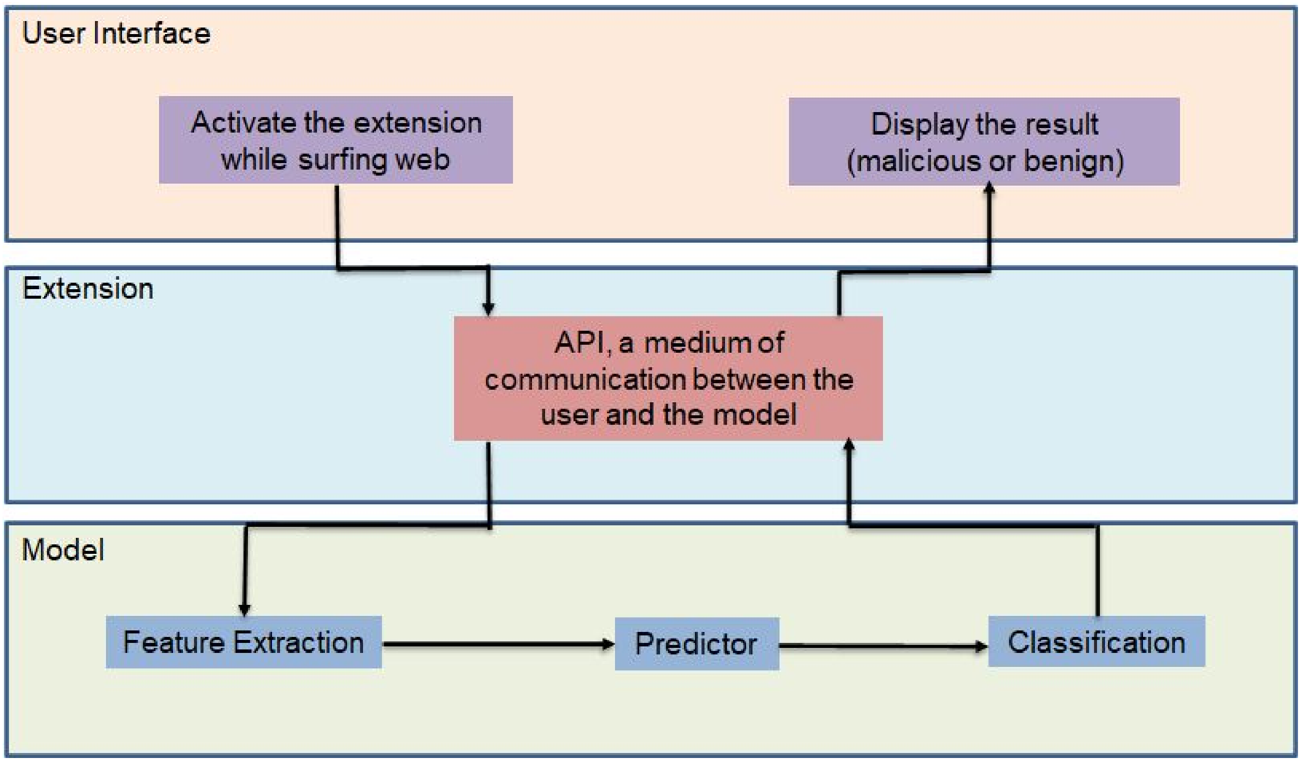
User Interface
Activate the extension while surfing web
Display the result (malicious or benign)
Extension
API, a medium of communication between the user and the model
Model
Feature Extraction
Predictor
Classification

Figure 3.1: System Architecture diagram

3.4 Data Flow Diagram

A data-flow diagram is a way of representing a flow of data through a process or a system. The Data Flow Diagram also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

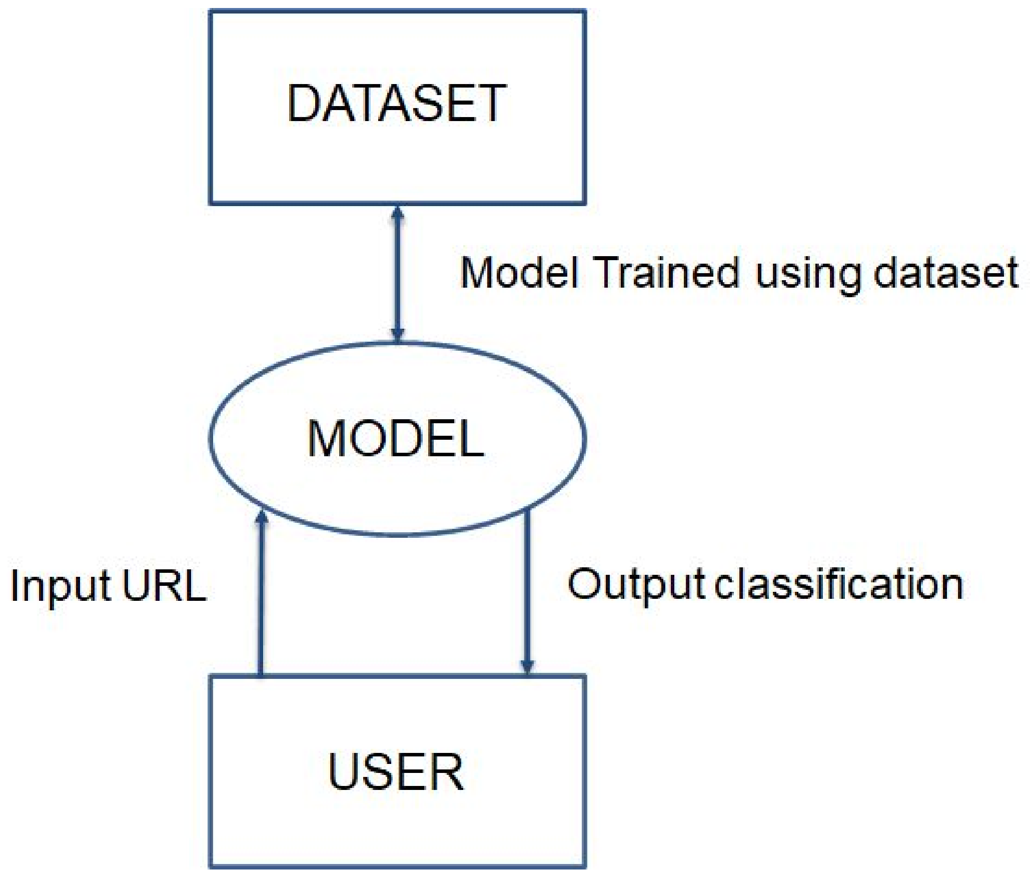

Figure 3.2: Data Flow Diagram

3.5 Activity Diagram

Activity diagram is defined as a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

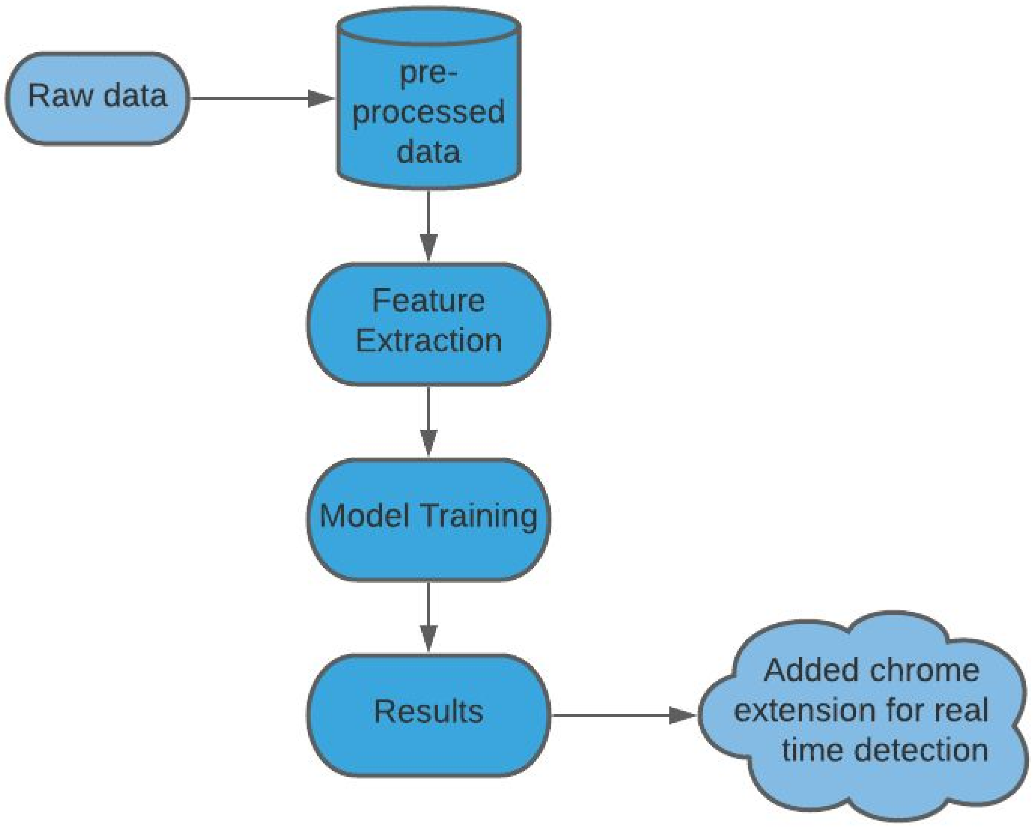

Figure 3.3: Activity Diagram

3.6 Use case diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

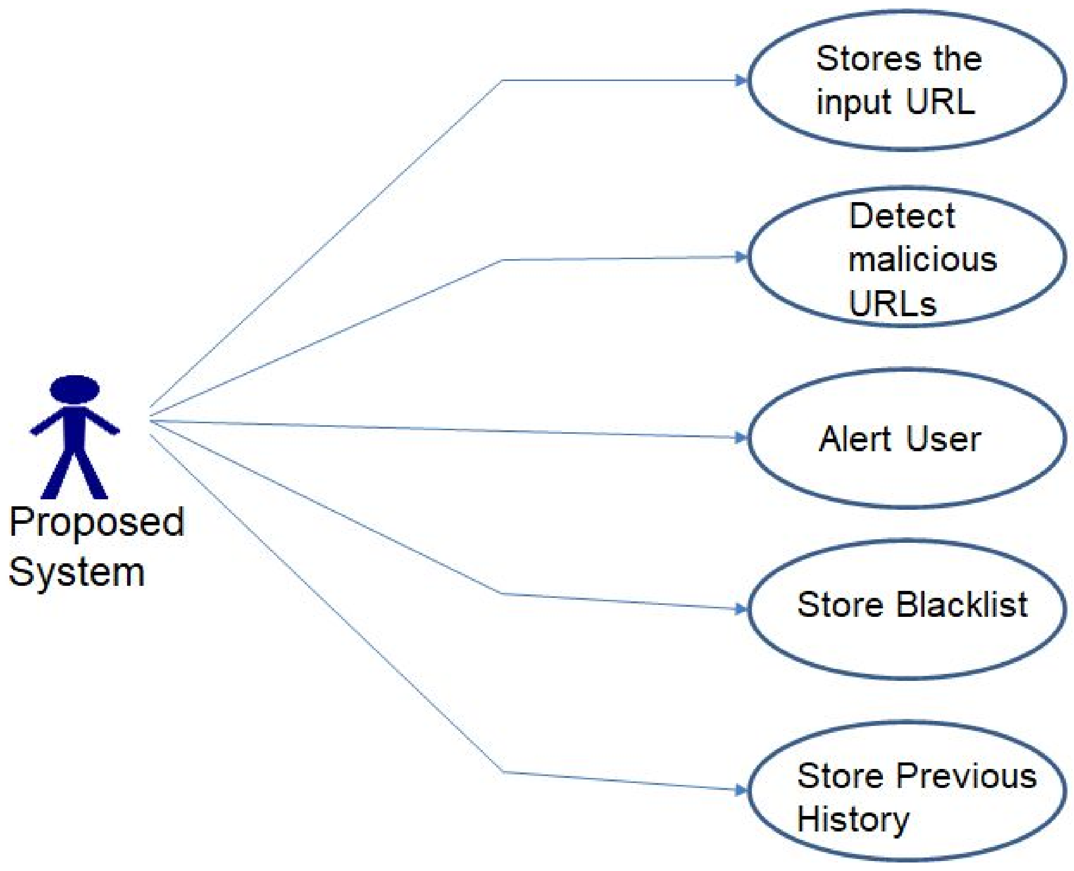

Figure 3.4: Use case diagram

3.7 Sequence diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function.

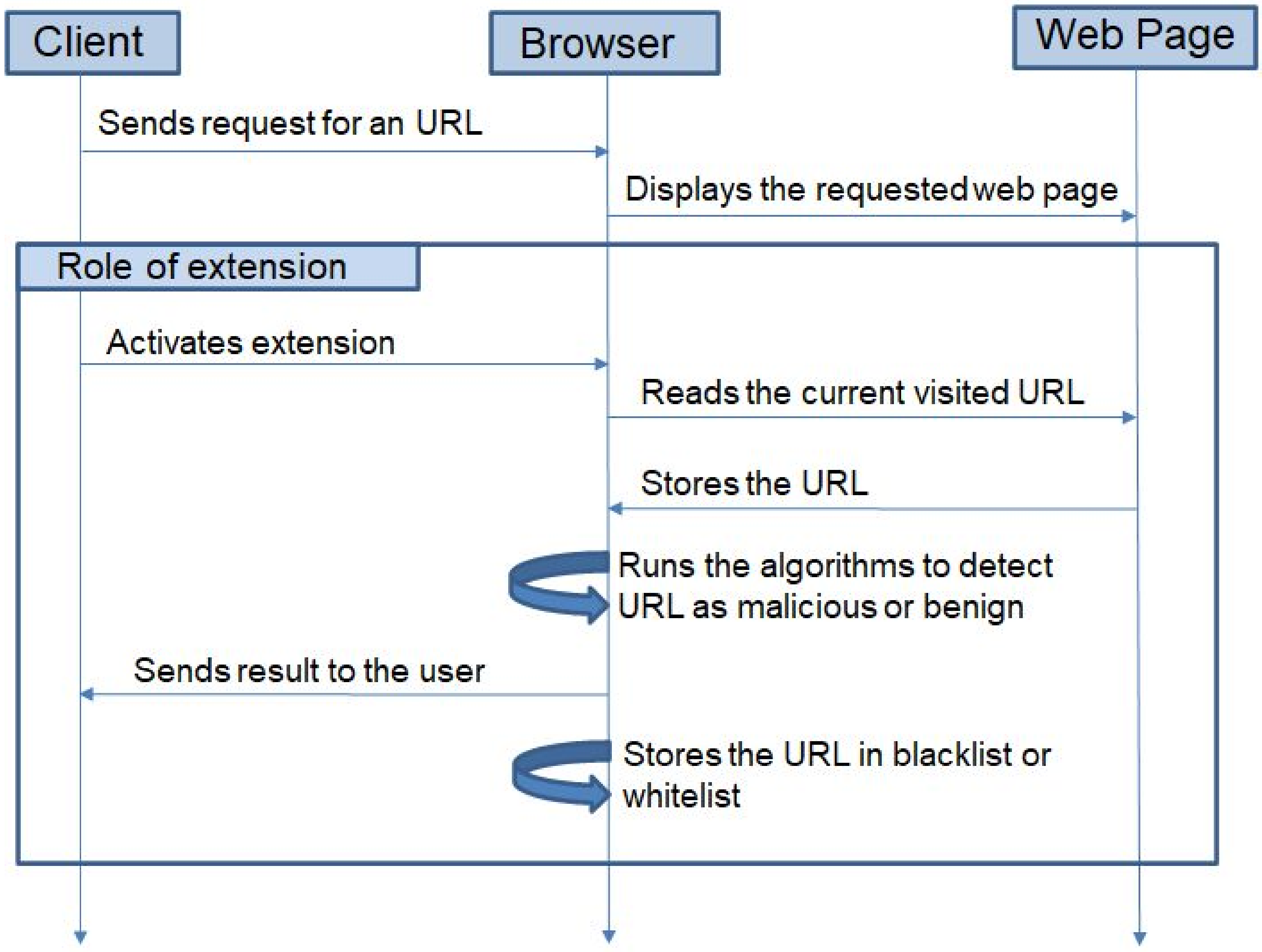
Client
Browser
Web Page
Sends request for an URL
Displays the requested web page
Role of extension
Activates extension
Reads the current visited URL
Stores the URL
Runs the algorithms to detect URL as malicious or benign
Sends result to the user
Stores the URL in blacklist or whitelist

Figure 3.5: Sequence diagram

3.8 State transition diagram

State-transition diagrams describe all of the states that an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (guards), and the activities undertaken during the life of an object (actions).

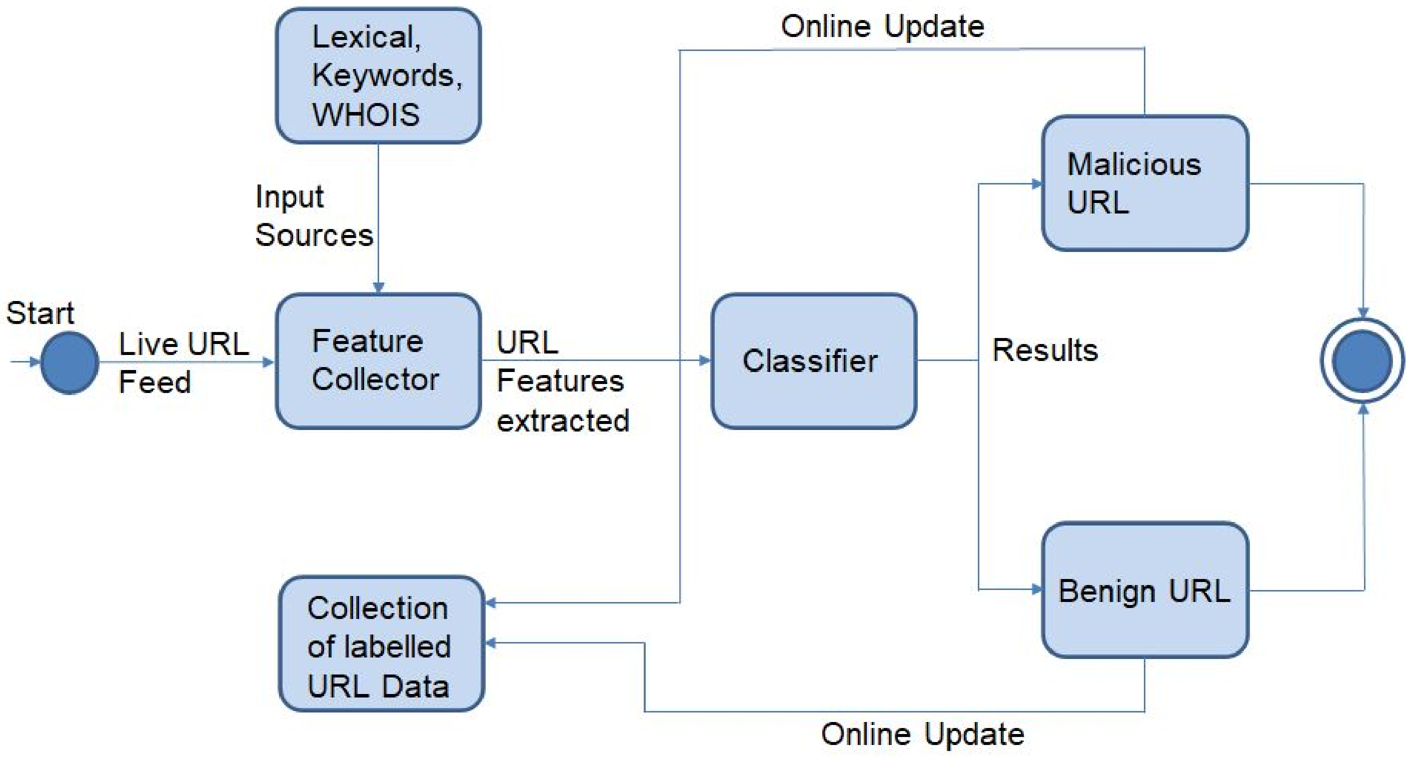

Figure 3.6: State transition diagram

CHAPTER 4

CONCLUSION

Malicious URL detection plays a critical role for many cybersecurity applications, and clearly machine learning approaches are a promising direction. In this project work, we have described how machine learning techniques are able to judge the URLs based upon the lexical feature set. Specifically, we described the lexical feature sets and an approach for detecting malicious URLs based on the extracted lexical feature set. When traditional methods fall short in detecting the new malicious URLs on its own, our proposed method can be augmented with it and is expected to provide improved results. Here in this work, we proposed the lexical feature extraction which is able to classify the URLs as malicious or benign. The Future work is to fine tune the machine learning algorithm that will produce the better result by utilizing the extracted lexical feature set. Adding to that the open question is how we can handle the huge number of URLs whose features set will evolve over time. Certain efforts have to be made in that direction so as to come up with the more robust feature set which can change with respect to the evolving changes in a real-time basis.

REFERENCES

[1] Detection of Malicious URLs using Machine Learning Techniques – Immadisetti Naga Venkata Durga Naveen, Manamohana K, Rohit Verma – IJITEE – March 2019

[2] Empirical Study on Malicious URL Detection using Machine Learning – Ripon Patgiri(B), Hemnath Katari(B), Ronit Kumar(B), Dheeraj Sharma – ICDCIT – January 2019

[3] Chrome Extension For Malicious URLs detection in Social Media Applications Using Artificial Neural Networks And Long Short Term Memory Networks – Shivangi S, Pratyush Debnath, Sajeevan K, D. Annapurna – IEEE – September 2018

[4] Detecting Malicious URLs Using Lexical Analysis – Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelkar – NSS – September 2016