

**MULTIMEDIA UNIVERSITY OF KENYA**

FACULTY OF COMPUTING & INFORMATION TECHNOLOGY

**PROJECT PROPOSAL**

**PHISHING URL DETECTION USING MACHINE LEARNING**

BY

**NAME: SIMON NZESI MUTUA**

**REG. No: CIT-227-010/2017**

**SUPERVISOR: MR. MUTURI**

May, 2021

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Submitted in partial fulfillment of the requirements of Bachelor of Science in Software Engineering.**

# ABSTRACT

The Internet has become an indispensable part of our life, However, it also has provided opportunities to anonymously perform malicious activities like Phishing. Phishers try to deceive their victims by social engineering or creating mock up websites to steal information such as account ID, username, password from individuals and organizations. Although many methods have been proposed to detect phishing websites, Phishers have evolved their methods to escape from these detection methods. Mostly, these methods rely on blacklisting and whitelisting techniques giving the attackers a chance to easily bypass the implemented security methods. Also these methods are very specific to certain phishing instances only and hence do not detect zero day attacks.

One of the most successful methods for detecting these malicious activities is Machine Learning. This is because most phishing attacks have some common characteristics which can be identified by machine learning methods. This paper focuses on the use of machine learning methodology in conjunction with a browser extension that runs on the client side (browser) to effectively check all the URLs that the user visits and determine if those URLs are meant for phishing attacks. The browser extension then flags such URLs and warns the user of the possible phishing attack. This browser extension is expected to run independently of the browser platform in use.

# List Of Abbreviations

ICT – Information and Communication Technology

WWW -World Wide Web

SQL – Structured Query Language

URL – Uniform Resource Locator

HR – Human Resources

COVID-19 – Coronavirus Disease

SEA – Social Engineering Attacks

API – Application Programming Interface

IP – Internet Protocol

DNS -Domain Name System

XSS – Cross-site Scripting Attacks

IDN – International Domain Name

**Table of Contents**

[ABSTRACT ii](#_Toc79633185)

[List Of Abbreviations iii](#_Toc79633186)

[List Of Tables v](#_Toc79633187)

[List Of Figures vi](#_Toc79633188)

[CHAPTER 1: INTRODUCTION 1](#_Toc79633189)

[1.1 Background of Study 1](#_Toc79633190)

[1.2 Problem Statement 2](#_Toc79633191)

[1.2.1 Proposed solution 3](#_Toc79633192)

[1.3 Aim of the study 3](#_Toc79633193)

[1.3.1 Research Objectives 3](#_Toc79633194)

[1.4 Justification of the Study 4](#_Toc79633195)

[1.5 Scope 5](#_Toc79633196)

[1.6 Assumptions 5](#_Toc79633197)

[1.7 Limitations 5](#_Toc79633198)

[CHAPTER 2: LITERATURE REVIEW 6](#_Toc79633199)

[2.1 Introduction 6](#_Toc79633200)

[2.2 Related Systems 6](#_Toc79633201)

[2.2.1 User Education and Training about Phishing 6](#_Toc79633202)

[2.2.2 Server-side software solutions 7](#_Toc79633203)

[2.2.3 Client-side software solutions 9](#_Toc79633204)

[2.3 Limitations/Weaknesses of these systems 13](#_Toc79633205)

[2.4 How the proposed solution Handles these weaknesses 14](#_Toc79633206)

[CHAPTER 3: METHODOLOGY 15](#_Toc79633207)

[3.1 Introduction 15](#_Toc79633208)

[3.2 Software Development Methodology 15](#_Toc79633209)

[3.2.1 Evolutionary Model 15](#_Toc79633210)

[3.2.2 Evolutionary Model Steps 15](#_Toc79633211)

[3.3 Project Resources 17](#_Toc79633212)

[3.4 Project Schedule 18](#_Toc79633213)

[References 19](#_Toc79633214)

# List Of Tables

[Table 1: Project schedule 18](#_Toc79632600)

[Table 2: Project Budget 18](#_Toc79632601)

# List Of Figures

[Figure 1: Evolutionary Model 16](#_Toc79632592)

# CHAPTER 1: INTRODUCTION

## 1.1 Background of Study

The ever-growing and fast expanding reach of the internet is coupled with a rapid spread of information and communication technology (ICT). This advent of new communication technologies has had tremendous impact in the growth and promotion of businesses spanning across many applications including online-banking, e-commerce, and social networking. In fact, in today’s age it is almost mandatory to have an online presence to run a successful venture.

As a result, the importance of the World Wide Web (WWW) has continuously been increasing. Unfortunately, the technological advancements come coupled with new sophisticated techniques to attack and scam users. Such attacks include rogue websites that sell counterfeit goods, financial fraud by tricking users into revealing sensitive information which eventually lead to theft of money or identity, or even installing malware in the user’s system. There are a wide variety of techniques to implement such attacks, such as explicit hacking attempts, drive-by download, social engineering, phishing, man-in-the middle, SQL injections, loss/theft of devices, denial of service, distributed denial of service, and many others.

Most of these attacking techniques are realized through spreading compromised URLs (or the spreading of such URLs forms a critical part of the attacking operation). URL is the abbreviation of Uniform Resource Locator, which is the global address of documents and other resources on the WWW. A URL has two main components : protocol identifier (indicates what protocol to use also known as scheme) and a resource name. The resource name component includes the subdomain, domain name, top level domain, port, path, query, parameters, and fragment segments.

Compromised URLs that are used for cyber attacks are termed as malicious URLs. The most popular type of attack using malicious URLs is the phishing attacks.

Phishing is a social engineering attack which aims at exploiting weaknesses found in system end users. In phishing, an attacker attempts to steal user’s private, sensitive data and tries to manipulate the data by using a malicious fake link which exactly looks like a legitimate link (i.e. the attacker masquerades as a trusted entity/reputable source). When a user clicks on that link, he/she is redirected to the hacker's webpage instead of the legitimate one. User thinking that it’s legitimate, provides them the sensitive data such as login credentials, credit card numbers, email addresses and phone numbers and hence user’s data gets compromised.

With the outbreak of the COVID-19 pandemic, a significant rise of phishing attacks has been observed. The global impacts of coronavirus, as well as its implications, such as quarantine measures, a remote workforce, and the length of the pandemic are the core reasons for the increased cases. According to WHO, hackers and cyber scammers are taking advantage of the pandemic by sending fraudulent email and WhatsApp messages that attempt to trick you into clicking on malicious links or opening attachments aimed at carrying out a phishing attack.

For instance, a spam and phishing report by Kaspersky has revealed that they detected 2,023,501 phishing attacks in South Africa, Kenya, Egypt, Nigeria, Rwanda and Ethiopia in the second quarter of 2020. According to the report, South Africa had the biggest number of attacks at 616,666 within three months. This was followed by Kenya with 514, 361 attacks, Egypt at 492,532 attacks, Nigeria at 299,426, Rwanda at 68,931 and Ethiopia at 31,585. The communication to unsuspecting users was disguised as delivery services, postal services, financial services and HR services triggered by embracement of remote workforce across the globe.

## 1.2 Problem Statement

Phishing has been around and it continues to present a substantial risk to businesses as well as individuals and is often cited as a top security concern. The concern is driven by increasingly sophisticated attacks; the move from email to alternative attack vectors, such as social media and messaging; and the simple fact that phishing targets the weakest link in the security chain: people.

The global situation that has come into existence due to the COVID 19 pandemic has changed the equation in the space of Social Engineering attacks. The increase in Work-from-home situations, online education, and entertainment via online platforms has created a sharp uptick in the number of Internet Users worldwide and also consequently increasing the phishing attacks.

Phishing, one of the staples in the SEA arsenal has seen a huge increase, with technology companies such as Google and Microsoft recording trends where the attackers masquerade as officials from organizations working on COVID-19 such as the World Health Organization. As a result, phishing attacks that use social media and messaging to launch and amplify an attack are on the increase.

Often, internet users make typing errors while entering the website URL which is exploited by attackers. Besides these, the attacker may also choose to manipulate the URL by altering the sub-domain names, query lengths, adding redirect requests or making the URL excessively long. Since phishing data is easily available in phishing databases such as Phishtank, once a website is suspected/flagged of being related to phishing, the attacker can easily modify the website URL by altering the sub-domain names to make a new website. Therefore, there is a need for an intelligent method for identifying phishing URLs and reduce phishing attacks. Machine learning techniques can help in classification and prediction of website URLs into phishing and legitimate URLs

### 1.2.1 Proposed solution

The proposed solution is a browser extension framework that works at the client side (browser) of the user accessing web pages from the internet. Once installed, this extension detects phishing URLs by a series of 3 stages which are organized into 3 layers as follows:

* Whitelist check – using Alexa Top Sites API
* Blacklisting websites – using phishTank API
* Feature extraction and classifiers – by using machine learning

## 1.3 Aim of the study

The main goal of this study is to develop an efficient phishing detection system to allow a user who visits a URL on the Internet determine whether a URL is a phishing or legitimate one by ensuring high accuracy levels while reducing the average computational cost.

### 1.3.1 Research Objectives

1. To develop a phishing detection system whose assessment methodology should be fast enough to provide efficient results to the user as fast as possible.
2. Design a browser extension that is capable to handle real-time scenarios and zero-day attacks.
3. To train and test a machine learning model for feature extraction and classification in real time to cater for new websites that causes zero-day phishing attacks.
4. To promptly alert the user in case of a phishing instance through the cross-browser extension that interacts with the server component of the system.

## 1.4 Justification of the Study

Most of the anti-phishing solutions in literature claim high accuracy as 98% for phishing detection but most of these measures fail to handle real-time zero-day attacks. There is a huge gap between the high accuracy that has been reported in articles but when it comes to real-time scenario implementation, most of the existing solutions have very low effectiveness.

A few major reasons for low-effectiveness of most of the existing solutions are:

1. The design and ideology behind the solution is influenced by high detection accuracy obtained by training the classifiers with datasets having limited features or spatial correlation.
2. The models used for testing the evaluation of anti-phishing solutions are not capable of representing the real time scenarios effectively.

The major requirements for an effective anti-phishing solution is to have these characteristics:-

* The detection performance should be evaluated in real time scenarios after considering all the use cases and deployment cases.
* The temporal resilience of the data set is a must.
* The evaluation or assessment methodology should be fast enough to provide efficient results in fractions of seconds.

The proposed system sees into it that these requirements are attained by combining the advantages of profile matching techniques and machine learning classifiers in the process of phishing detection.

## 1.5 Scope

The proposed system covers all the URLs that the user visits using their web browser. This means both those that the user opens by clicking from other sources such as email clients or manually typing URL in the browser address bar.

The browser extension for the system is platform independent.

## 1.6 Assumptions

The system does not block a web page after flagging it as a phishing URL. Assumption is hereby made that it is the user’s role to avoid continued browsing to such a website or web page after the system alerts the user of the possible phishing scenario.

## 1.7 Limitations

The system is only fully advantageous to the user when using a web browser in a PC, a laptop or a tablet. This is because of the default behavior of most of the mobile-based web browsers not able to install and run browser extensions.

# CHAPTER 2: LITERATURE REVIEW

## 2.1 Introduction

Automatically detecting phishing web pages has attracted much attention from security and software providers, financial institutions, to academic researchers. This section analyzes and discusses the several approaches and comprehensive strategies that have been suggested to tackle phishing attacks through each level of attack flow. This section also outlines the weaknesses of these methodologies and how the proposed system handles these issues to bridge the gaps.

## 2.2 Related Systems

The solutions that have been proposed in literature and in industry can be grouped into three main categories:

User education and training

Server-side software solutions

Client-side software solutions

### 2.2.1 User Education and Training about Phishing

Phishing attacks usually target the users who are not much aware of the defense mechanisms for phishing. One of the effective solutions for phishing is thus training and describing to users the fact that not to follow links blindly to any web sites where they have to enter sensitive information such as passwords (Cranor, 2007). In one of the 2010 studies, the author has found that training users is useful and help people to identify the fake website provided the users read the phishing material seriously and understand this crime (Sheng, 2010).

Educating users and company employees and warning them about phishing attacks have an impact on preventing phishing attacks. Multiple methods have been proposed for training users. Many researches concluded that the most impactful approach to help the users to distinguish between phishing and legitimate websites is interactive teaching (Dodge Jr, 2007).

Although user training is an effective method, however, humans errors still exist and people are prone to forget their training. It has also been noted that providing anti-phishing material and training users are sometime ineffective since the users are more familiarized of receiving such warnings and thought they already knew how to protect themselves. Training also requires a significant amount of time and it is not much appreciated by non-technical users.

### 2.2.2 Server-side software solutions

Server-side solutions are server-based applications that attempt to mitigate the phishing problem. The idea behind server-side anti-phishing solutions is to protect a user from being a victim of a phishing attack by filtering incoming emails, taking action against fraudulent websites, or applying authentication protocols at the recipient’s mail server. These solutions make use of email-content analysis, notice-and-take-down, or protocol-based authentication methods.

#### 2.2.2.1 Email-content analysis method

The email-content analysis method focuses on examining incoming emails to find specific features of fake emails to prevent such emails from reaching the user’s inbox. To determine these features, a number of known fake emails are analyzed. The model-based machine learning and property-structure based techniques are the examples of the various techniques associated with this method.

Bergholz et al. (2008) proposed a model-based machine learning technique. In this technique a new email’s features are compared to features of known phishing emails. Then a judgment on the new email is made as to whether this email is fake or normal. This technique uses 27 basic features and different advanced features. The basic features can be grouped into five categories: structural, link, element, spam filter-based and word-based. The structural features are: the total number of body parts, the number of discrete and composite body parts, and the number of alternative body parts. The link features are: the total number of links, the number of internal and external links, the number of links with IP-numbers, the number of deceptive links, the number of links behind an image, the maximum number of dots in a link, and a Boolean, indicating whether there is a link whose text contains one of the following words: click, here, login, update. The element features are four Boolean features with regard to whether or not HTML, scripting, JavaScript, and forms are used. The spam filter-based features are: the filter test score and a Boolean of whether or not an email is considered to be spam. The word-based features are Boolean features of whether or not the words “account, update, confirm, verify, secur, notif, log, click and inconvenien” occur in the email. The advanced features are proposed by the authors. They adaptively trained Dynamic Markov Chains and novel latent Class-Topic Models to generate these features . To compare the new email’s features to the proposed features, the technique uses a classifier. Typically, this classifier has two inputs: the values of the phishing emails’ features (the training set of the classifier), and the values of the new emails’ features (the test set of the classifier).

Another technique is proposed by Chandrasekaran et al. (2006). This technique makes use of the structural properties of phishing emails to distinguish between legitimate and fake emails. To achieve their target the authors have identified 25 features. These features can be grouped into two categories: style markers-based (I.e Total number of characters, Total number of unique words, Word count, Total number of function words, Function word frequency distribution, Account, Log, Access, Bank, Credit, Click, Identity, Inconvenience, Information, Limited Minutes, Password, Recently, Risk, Social, Security, Service, Suspended, Total number of words) and structural attributes-based features ( Structure of email subject line, Structure of the greeting provided in the email body). The authors used 100 phishing and 100 legitimate emails as input to the simulated annealing algorithm, to identify the useful features. From the relevance between such features, information gain (IG) has been used to rank these features. Based on the candidate features, the authors used the Support Vector Machine (SVM) classifier to classify phishing emails.

#### 2.2.2.2 Notice-and-take-down method

Another method to combat phishers is to attack their websites before they can start harming any individuals. This can be done by finding these websites’ URLs from reported phishing emails, for example, then try to remove these websites from the Internet. Typically, specialist companies play this role as a service to financial organizations.

#### 2.2.2.3 Authentication protocol method

This method tries to solve the phishing problem by adopting authentication schemas. These schemas can be applied on the email protocol (SMTP), which is designed without security requirements. Using this method, sender’s identity can be examined. This can mitigate phishing risks. A number of techniques, that adopt this method, have been proposed such as senderID by Microsoft and DomainKeys Identified Mail (DKIM) by Yahoo.

Sender ID is used to detect spoofing. Sender ID uses the RECEIVED SMTP header and a query to the DNS records for the sender's domain to determine if the sender's email address is spoofed. Sender ID in Exchange Server is provided by the Sender ID agent, and is basically unchanged from Exchange Server 2010. When the Exchange server receives an inbound message, the Sender ID agent verifies the sender's IP address by querying the DNS records for the sender's domain. This check confirms that the message was received from an authorized IP address for the sender's domain. The IP address of the authorized sending server is referred to as the purported responsible address (PRA).

### 2.2.3 Client-side software solutions

Client-based solutions are designed to work on the Internet users’ machines. That is, using plug-ins or browser helper objects (BHOs – for Microsoft Internet Explorer) which a user can install to monitor visited web pages, and to warn the users if they have entered a fraudulent page. These solutions are different in terms of how to determine if a visited page is fraudulent or not. They can be classified into four groups: blacklist-based or white-list-based, visual-clue-based, webpage-feature-based and information-flow-based solutions.

#### 2.2.3.1 Blacklist-based method

The majority of anti-phishing methods rely on a blacklist, a list of known phishing domains. This method combats the phishing attempts by preventing user from accessing web pages that appear in the blacklist. To build this list, the method requires retrieving recent URLs of phishing webpages from specialist websites such as Anti-phishing Working Group (APWG) or PhishTank, or alternatively may receive these URLs from the users directly. The technique of Microsoft SmartScreen component make use of blacklists method.

Microsoft SmartScreen is integrated with the Internet Explorer and their modern web browser, Microsoft Edge. This tool uses two methods to determine the nature of a page: blacklist checking and heuristics analyses. Basically, when a user visits a site using IE9/Edge, the SmartScreen Filter will compare a page’s contents against heuristics characteristics, which are updated periodically using machine learning techniques developed by Microsoft. If suspicious properties are found, the tool will warn the user to avoid providing any confidential data by causing a yellow shield to appear. However, if the page passes the heuristics test, the tool will check its URL against a frequently updated online blacklist. If the URL is found in the blacklist, the page’s contents will be blocked, and a red shield will appear in the address bar. The user then has the choice whether to proceed or to close the page. The tool also checks downloaded files against the same blacklist, and the later processes will be applied. SmartScreen Filter provides its user with a reporting feature to notify Microsoft about new fraudulent URLs. In addition, to decrease the false positive detection rate, this tool depends only on verified unsafe URLs provided by reviewers at Microsoft or by employees from third parties.

#### 2.2.3.2 Visual-clue-based method

Visual-clue-based method applies the idea of using images as a base for the solution to combating phishing attacks. This method relies on the fact that phishing attackers try to lure users by imitating visual features of target websites. This method tends to use images as authentication evidences that the server should present. Visible Watermarking is an example of a visual-clue-based techniques.

Visible watermarking (ViWiD) is an integrity check technique in which the user needs to verify a watermark within the company webpage’s logo to authenticate this webpage (M. Topkara, 2005) . This watermark consists of two parts: a shared secret, which the user selects at the registration stage in a secure manner, between the user and the company’s sever, and the current date and time of the user’s time zone determined by the IP address of the user’s machine. This watermark is designed to be unique for every user in order to combat a “one size fits all” attack. The company’s logo can appear to the user in two ways: after the user login into his or her account, or by using a cookie. The last choice is preferred since the user need not to enter his or her confidential data on the login webpage to avoid revealing this data on a forged webpage. The user can trust the server’s webpage since its logo includes the shared secret. The process of adding the watermark to the company’s logo is done on the company web server, and the user need not install any tool or store any data on his or her local machine.

#### 2.2.3.3 Webpage-feature-based method

Another method depends on analyzing the web-page's contents to find fraud symptoms, and then warning the user of a potential phishing attack. A number of techniques that adopt this method have been proposed, for example SpoofGuard.

SpoofGuard is used in mitigating simple phishing attacks (Chou, 2004). The plug-in monitors a user’s Internet activity, computes a spoof index, and warns the user if the index exceeds a level selected by the user. SpoofGuard uses domain name, URL, link, and image checks to evaluate the likelihood that a given page is part of a spoof attack. For ex-ample,a page with a suspicious URL such *https://asetrade-maintenance.suspicious.orgorwww.etrade.com@129.170.213.101maintainance.asp* and an E\*Trade logo will have a higher spoof index than a page with neither of these characteristics. SpoofGuard also uses history, such as whether the user has visited this domain before and whether the referring page was from an email site such as Hotmail or Yahoo!Mail. Most importantly, spoofGuard intercepts and evaluates user posts in light of relevant history and the spoof index of a form page. SpoofGuard examines post data user name and password fields and compares posted data to previously entered passwords from different domains. This mechanism warns a user against sending her E\*Trade password to a site with an E\*Trade logo but outside the *etrade.com* domain*,* for example. Password comparisons are done using a cryptographically secure hash, so that plain text passwords are never stored by SpoofGuard

Therefore, some of these evaluations are done after downloading the webpage: URL, link, image and domain checks. In addition, some evaluations are conducted when the user interacts with such a page: password, outgoing password, referring page, outgoing post data checks.

#### 2.2.3.4 Information-flow-based method

Information-flow-based method tries to protect users from being victims of phish attacks by tracking their sensitive information to make sure that they provide this information on trusted websites. A user will be warned, if he/she is about giving away her confidential data on fake websites. One technique that follows this method is AntiPhish (Kirda, 2005). This technique detects phishing by examining the current web-page's domain when a user starts to enter sensitive data.

The AntiPhish technique’s main purpose is to protect users’ confidential data. This can be done by monitoring where the users’ confidential data is been entered and informing the user in the case of a phishing attack. Typically, when a user enters confidential data in a web page’s form for the first time, he/she may ask AntiPhish to capture this data and stores it in an encrypted form. AntiPhish uses the DES encryption algorithm to encrypt users’ confidential data by a master password. AntiPhish also stores a web page’s domain to be mapped with the user data. AntiPhish uses a domain rather than a web page’ address because some websites are hosted in more than one server. However, if AntiPhish uses the address, false attack detection may be triggered. The user needs to provide the master password the next time in order to automatically fill in the previous web page’s form. To monitor the users’ confidential data, AntiPhish examines text field elements of any form in a web page and interrupts any user event. If the user interacts with a text element, AntiPhish will compare the element value against a list of previous stored user’s confidential data. If it finds a match, domains comparison will start. If there is no match, AntiPhish will consider the current webpage as phishing. AntiPhish runs same test if the user generates evens on test elements: press a key, load new page, click or focus. JavaScript gives an attacker the ability of accessing form’s text elements before a user submits inputs. To combat this problem, AntiPhish deactivates JavaScript if the focus is on a text element and reactivates it when the focus is lost.

## 2.3 Limitations/Weaknesses of these systems

#### 2.3.1 User Education

People are prone to forget their training.

Training also requires a significant amount of time and it is not much appreciated by non-technical users.

#### 2.3.2 Server-side software solutions

The email-content analysis method may not identify some browser vulnerabilities-based attacks , such as International Domain Name (IDN) spoofing and pop-up hijacking attacks. This is because the proposed technique focuses only on email-based attacks.

The notice-and-take down methods require Server Administrator interaction to remove phishing pages thus affecting the performance of the solution.

The authentication protocol methods, sometimes the forwarding services need to modify a message’s content. In many instances this modification will result in authentication fail.

#### 2.3.3 Client-side software solutions

As with any blacklist-based solution, users are still exposed to new phishing attacks. That is, the URLs of newly established phishing sites may not yet be included in the blacklist.

The Visual-clue-based technique requires the user to have some knowledge of phishing attacks and how to identify spoofed pages in order to distinguish between an authentic and a spoof webpage. As a result of the leak knowledge, more than 20% of users ignore web-page’s visual clues and even professional users may be victims of visual-based attacks.

For the Visible-watermarking, the users have to be trained to expect what information should appear in the company’s logo in order to distinguish between real and fake web pages.

SpoffGuard can be fooled using sophisticated hybrid phishing attacks such as Cross-Site Scripting (XSS) because it is developed to address simple phishing attacks. Also, it it platform dependent as it only works with Microsoft Internet Explorer.

In Information-flow-based method, the user needs to inform AntiPhish to capture their confidential data. I.e the user participates in the verification process.

## 2.4 How the proposed solution Handles these weaknesses

Various anti-phishing solutions proposed by different authors have been given in the previous section. However, no single solution is a “full proof” solution for combating phishing attack. The Limitations of the existing anti-phishing solutions outlined above emphasize the need for innovative solutions.

The proposed anti phishing solution will be implemented at the client-side in the form of a browser extension and should be capable to handle real-time scenarios and zero day attacks. The proposed approach works efficiently for any phishing link carrier mode as the execution on clicking on any link or manually entering URL in the browser automatically launches the evaluation process to determine the legitimacy of the specified URL. This ensures that the user needs not to manually be involved in verifying if a URL is genuine or not as long as the browser is turned on.

Since client-side solutions in the form of browser extensions are easy to install and use, this ensures that the computational costs are kept at minimum thus increasing the efficiency of the proposed solution.

The proposed solution makes use of machine learning which makes it easy to detect zero-day phishing attacks. This is because most phishing attacks have some common characteristics which can be identified by machine learning methods. As a result, the system can make super fast decisions in future of determining the legitimacy of a URL based on the previously identified instances of such URLs.

Also, the proposed framework is capable to handle DNS cache poisoning attacks even if the system’s DNS cache is somehow infected.

# CHAPTER 3: METHODOLOGY

## 3.1 Introduction

Software development methodology is the process of dividing software development work into smaller, parallel or sequential steps or sub-processes to improve design, product management and project management.

## 3.2 Software Development Methodology

This project will focus on the use of the evolutionary model of software development.

### 3.2.1 Evolutionary Model

Evolutionary model suggests breaking down of work into smaller chunks, prioritizing them and then delivering those chunks to the customer one by one. The number of chunks is huge and is the number of deliveries made to the customer. The main advantage is that the customer’s confidence increases as he constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements. The model allows for changing requirements as well as all work in broken down into maintainable work chunks.

The choice of this methodology comes handy due to the time factor involved; the project is estimated to run for the next ten weeks which is a major constraint in the development.

Due to the reason stated, the use of iterative model will allow for instant modification upon and during development and in addition the application can be altered easily. Evolutionary model will allow growth of this project in a steady way.

### 3.2.2 Evolutionary Model Steps

The following are the steps to be taken to complete the development of the project:

1. Drafting of the Project proposal
2. Analysis of requirements and software specification
3. Design of different Module and interface
4. Implementation through Coding
5. Testing the application
6. Verification and validation
7. Documentation

The following section illustrates and discusses these steps in details:

Drafting Project Proposal

Requirements and software specifications

Analysis and Design

Implementation

Testing

Verification and Validation

Documentation

YES

NO

Figure 1: Evolutionary Model

*Requirements and software specification*

This is the first stage of development. It will involve researching and analyzing the requirements for the project to develop a list of requirements for the project.

*Design of different modules and interface*

This is the second phase of the development. After the requirement phase is complete, the requirements are analyzed and an initial design is created. This will be a simple design.

*Implementation*

This stage will include coding activities and use of python environment for code generation, guided by the initial design created in the second phase. Every specification will be applied into the program structure. If there are any errors with the design, the project will return to the design stage and a reviewed design will be produced.

*Testing*

This stage will check for the working of the functionalities implemented in the coding phase to test whether or not they work as expected and correct any errors arising from the coding activity itself.

*Verification and validation*

This stage will examine whether the prototype developed meets all the requirements of the project. If it does not meet the requirements, the process will have to take an entire iteration if the prototype meets all the requirements of the project, then the process moves to the next stage. This will also ensure the project is in due course with the referenced design from the earlier stages.

*Deployment*

The application will then be rolled out to the students in multimedia for their trial once the development time is reached. The students will then be able to give feedback and support in future development of the project.

Hopefully I will be able to integrate a simple feedback module so as to hear what people think of the project

## 3.3 Project Resources

*Hardware Requirements*

A computer/laptop

*Software Requirements*

Python 3.8

Visual Studio Code

PhishTank API

Alexa Top Sites API

Web Browser

JavaScript Framework

## 3.4 Project Schedule

This project will use an evolutionary prototyping model to realize the objectives. The estimated development time of the system is ten weeks. The duration of the activities of the model is shown on the gantt chart below

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PHASES** | **ACTIVITY** | **WEEKS** | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| PHASE ONE | Requirement Specification |  |  |  |  |  |  |  |  |  |  |
| Analysis and design |  |  |  |  |  |  |  |  |  |  |
| PHASE TWO | Implementation |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |
| Evaluation |  |  |  |  |  |  |  |  |  |  |
| PHASE 3 | Documentation |  |  |  |  |  |  |  |  |  |  |
| System review |  |  |  |  |  |  |  |  |  |  |

Table 1: Project schedule

**3.5** Project Budget

|  |  |
| --- | --- |
| **Resource** | **Cost** |
| Computer/Laptop | Available |
| Python 3.8 | Free |
| Visual Studio Code | Free |
| PhishTank API | Free |
| Alexa top sites API | Free |
| Internet Connection | 1000 |
| Google Chrome/ Mozilla Firefox | Free |
| Printing charges | 700 |
| **Total Cost** | **1700** |

Table 2: Project Budget

# References

1. V. B. et al, “study on phishing attacks,” *International Journal of Computer Applications*, 2018
2. Dodge Jr, R. C., Carver, C., & Ferguson, A. J. (2007). Phishing for user security awareness. *computers & security*, *26*(1), 73-80.
3. Kumaraguru, P., Sheng, S., Acquisti, A., Cranor, L.F., and Hong, J.I. (2010) *Teaching Johnny not to fall for phish. ACM Transactions on Internet Technology* 10, 2 (2010), 1 31.
4. N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, “*Client-side defense against web-based identity theft*,” in NDSS. The Internet Society, 2004.
5. <https://www.who.int/about/communications/cyber-security> (online)
6. <https://securelist.com/spam-and-phishing-in-q2-2020/97987/> (online)
7. P. Kumaraguru, J. Cranshaw, A. Acquisti, L. Cranor, J. Hong, M. A.  
   Blair, and T. Pham, “*School of phish: a real-world evaluation of anti-  
   phishing training*,” in Proceedings of the *5th Symposium on Usable  
   Privacy and Security*, pp. 1–12, 2009.
8. Topkara, M., Kamra, A., Atallah, M. J., & Nita-Rotaru, C. (2005, September). Viwid: Visible watermarking based defense against phishing. In *International Workshop on Digital Watermarking* (pp. 470-483). Springer, Berlin, Heidelberg.
9. Chandrasekaran, M., Narayanan, K., & Upadhyaya, S. (2006, June). Phishing email detection based on structural properties. In *NYS cyber security conference* (Vol. 3).
10. Bergholz, A., Chang, J. H., Paass, G., Reichartz, F., & Strobel, S. (2008, August). Improved Phishing Detection using Model-Based Features. In *CEAS*.
11. Kirda, E., & Kruegel, C. (2005, July). Protecting users against phishing attacks with antiphish. In *29th Annual International Computer Software and Applications Conference (COMPSAC'05)* (Vol. 1, pp. 517-524). IEEE.