A Method for Detecting Phishing URLs Using Machine Learning

[1]K.U.V.Padma, [2]D.Padma , [3]Ch.S.K.V.R.Naidu
Assistant Professor, Dadi Institute of Engineering & Technology,
kuvpadma@diet.edu.in , dpadma@diet.edu.in,
chskvrnaidu@diet.edu.in

Abstract
One of the most frequent and serious cybercrime assaults is abstract phishing. The information utilised by people and organisations to execute transactions is the target of these attacks. Phishing websites use a variety of indicators in both their text and information that is based on web browsers. The goal of this work is to classify 30 variables, including data from phishing websites, using Extreme Learning Machines (ELM) on a database at UC Irvine. ELM had the highest accuracy of 95.34% when compared to other machine learning techniques for results evaluation, including Support Vector Machine (SVM) and Naive Bayes (NB).

Existing System
The main goal of the phisher is to trick the user by creating an exact replica of a trustworthy website so that the user would not suspect the phishing site. Therefore, the anti-phishing techniques compare an image from a suspicious website with an authentic image database to determine the similarity ratio, which is then used to categorize questionable websites. When the similarity score exceeds a predetermined level, the website is labeled as phishing; otherwise, it is recognized as authentic.
More space is needed to store the legitimate image database. A web page with animations compared to a phishing website results in a low percentage of similarity, which results in a high false negative rate. This approach is unsuccessful.

Proposed System
*To reduce the false positives while identifying new phishing sites, combine new heuristic features with machine learning techniques.
*Made an attempt to find the best machine learning algorithm to detect phishing sites with higher accuracy than the existing techniques.

*Classified the websites as authentic and phishing using the machine learning methods Support Vector Machine (SVM) and Naive Bayes (NB).

*The recent literature's use of classifiers has influenced the decision to take into account these machine learning methods.

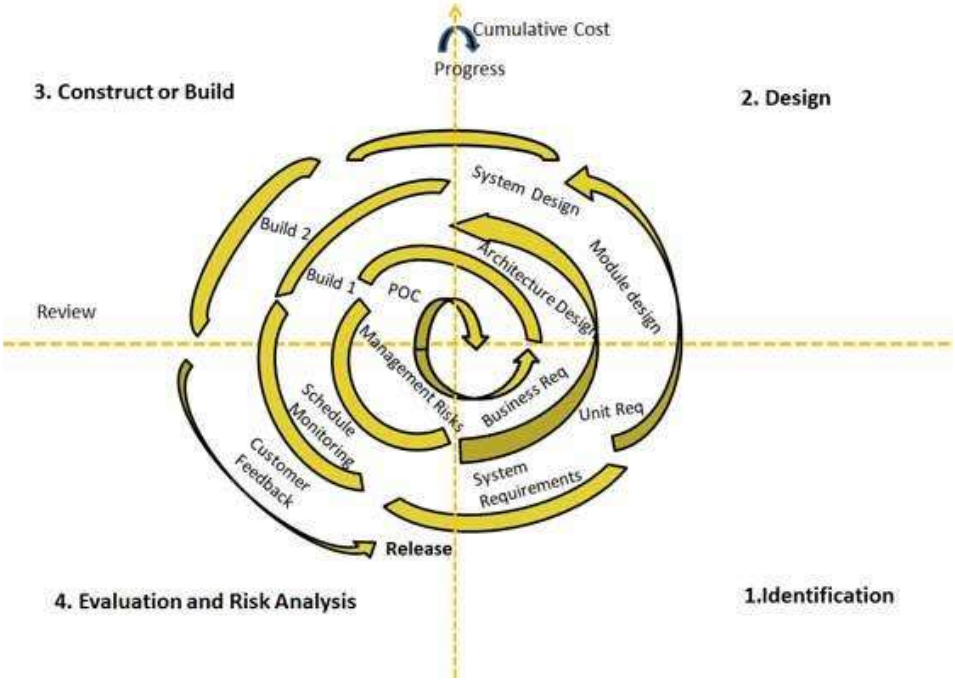
Phishing Attack Overview

Introduction

Because technology is developing so quickly, using the internet has become a necessary component of our daily lives. Data security of these systems has become extremely important as a result of the technology's rapid advancement and extensive use. Assuring that the required safeguards are taken against threats and dangers that users are likely to encounter while using these technologies is the basic goal of maintaining security in information technologies [1].Phishing is described as the impersonation of trustworthy websites in order to obtain the confidential information, such as usernames, passwords, and citizenship numbers, entered into websites on a daily basis for a variety of objectives. Phishing websites include a variety of indicators in their text and browser-based data [2-4]. The perpetrator(s) of the fraud transmit the bogus email or website.

WITH JUSTIFICATION, PROCESS MODEL IS USED Software Development Life Cycle is exactly what it sounds like. In order to create high-quality software, the software industry uses this standard.

The SDLC (Spiral Model)

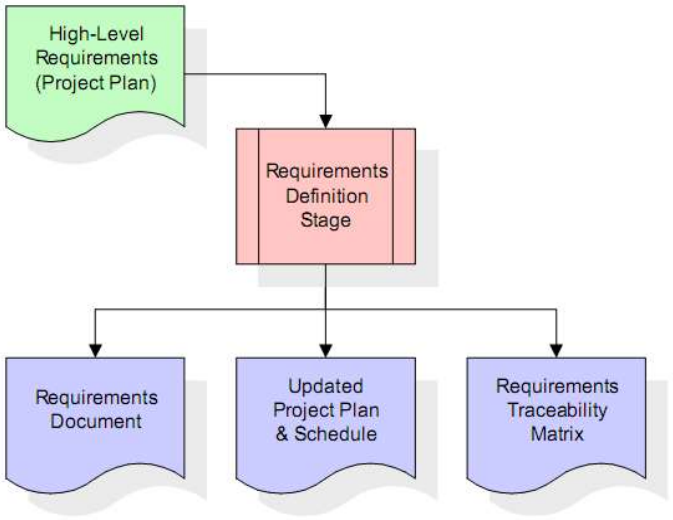


The gathering and analysis of requirements are stages of the SDLC.

- Designing
- Coding
- Testing
- Deployment

Requirements Definition phase and analysis

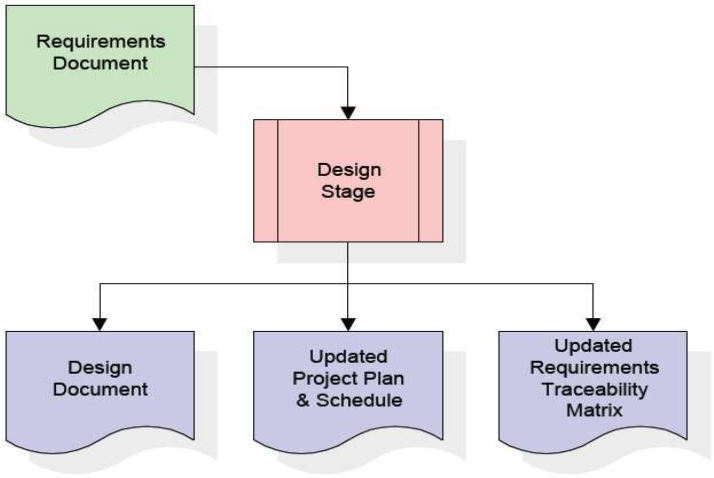
The objectives listed in the high-level requirements portion of the project plan are used as input during the requirements gathering process. Every objective will be further defined into a collection of one or more conditions. In addition to defining operational data areas and reference data areas as well as the initial data entities, these requirements also specify the main functions of the envisaged application. Crucial operations that must be managed as well as inputs, outputs, and reports that are essential to the purpose are considered major functions. With regard to these important functions, data spaces, and data entities, a user class hierarchy has been created. An individual requirement is each of these definitions. Requirements are recognised by distinctive requirement IDs, which must at least include.



The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary; however, detailed listings of database tables and fields are not included in the requirements document. The title of each requirement is also placed into the first version of the Requirements Traceability Matrix (RTM), which is the main deliverable for this stage.

Design Stage

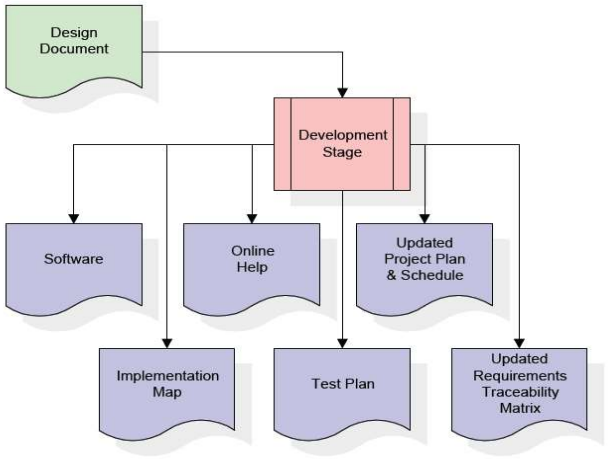
The requirements listed in the approved requirements document serve as the first input for the design stage. As a consequence of interviews, workshops, and/or prototype efforts, a collection of one or more design elements will be created for each requirement. A complete entity-relationship diagram with a full data dictionary, functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, and pseudo code are examples of design elements that explain the required software features in depth. The purpose of these design elements is to sufficiently define the software so that competent programmers may create it with little more help.



The RTM is updated to reflect that each design element is now formally linked to a particular requirement once the design document has been completed and approved. The project plan, an updated RTM, and the design document is the outputs of the design stage.

Development Stage

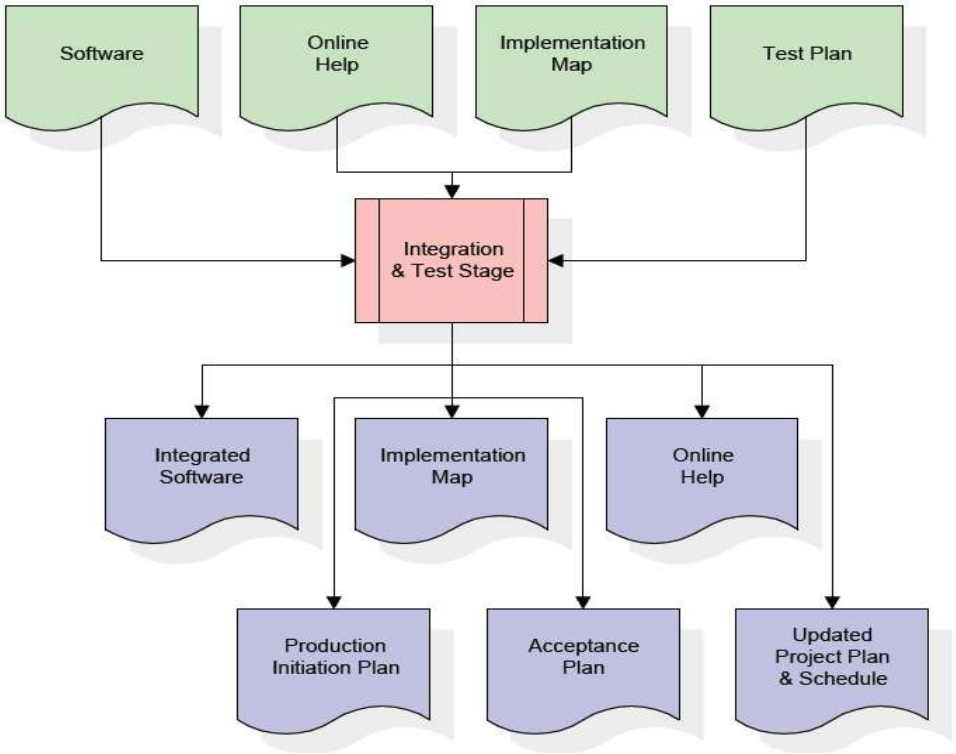
The design components listed in the approved design document serve as the development stage's main contribution. A set of one or more software artefacts will be created for each design element. Menus, dialogues, data management forms, data reporting formats, and specialized operations and functionalities are only a few examples of software artefacts. For each group of functionally related software artifacts, appropriate test cases will be created, and an online help system will be created to direct users as they engage with the product.



The RTM will be modified to reflect that every developed artifact has a corresponding test case item and is connected to a particular design element. The RTM is currently set up to its final state. A fully functional set of software that complies with the requirements and design elements previously documented, an online help system that explains how to use the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that outlines the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project are the outputs of the development stage.

Integration & Test Stage

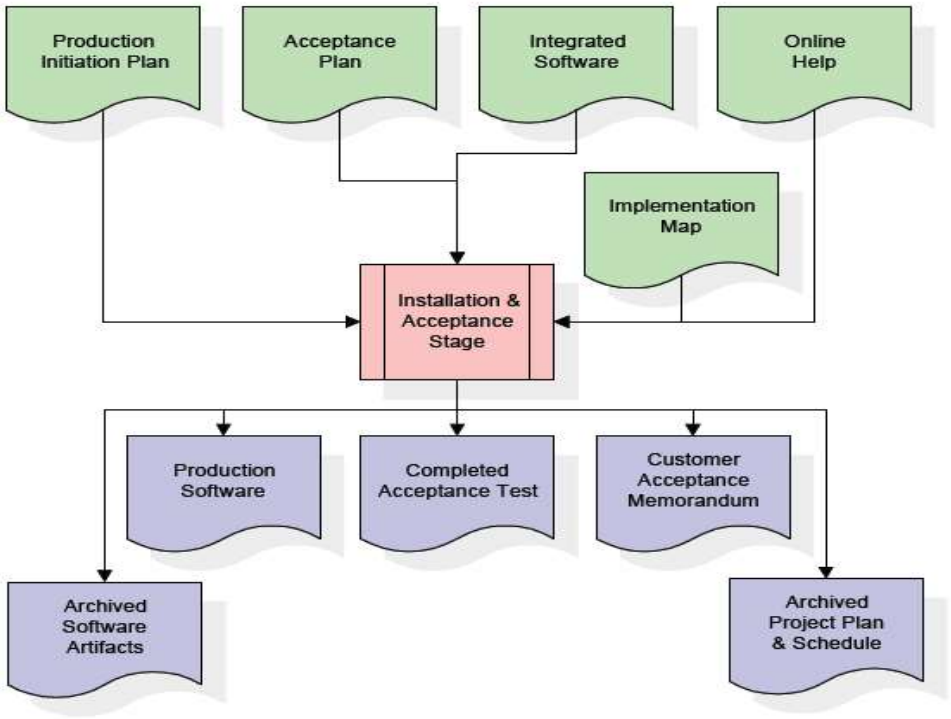
The migration of test data, online help, and software artifacts from the development environment to a separate test environment occurs during the integration and test stage. To confirm the accuracy and comprehensiveness of the software, all test cases are now executed. Executing the test suite successfully verifies a reliable and comprehensive migration capability. In this phase, production users are identified and matched to the correct roles while reference data is finalized for use in production. The Production Initiation Plan includes the production user list and the final reference data (or links to the source files for the final reference data).



An integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan that contains the final set of test cases, and an updated project plan are among the outputs of the integration and test stage.

Installation & Acceptance Stage

Software artifacts, online help, and the first batch of production data are all loaded onto the production server during the installation and acceptance phase. To confirm the accuracy and comprehensiveness of the software, all test cases are now executed. Acceptance of the software by the customer is contingent upon the test suite being executed successfully. The customer formally accepts the delivery of the software once customer personnel confirm that the initial production data load is accurate and the test suite has been completed with positive results.



A production application, a suite of completed acceptance tests, and a statement from the client confirming their acceptance of the software are the main outputs of the installation and acceptance stage. The project is then locked as a permanent project record and the PD R inputs the final of the real labour data into the project schedule. The PDR "locks" the project at this point by saving all

software components, the implementation map, the source code, and the documentation for later use.

Conclusion

In this research, we outlined the characteristics of phishing assaults and provided a categorization model to categorise phishing attacks. This technique uses a categorization section and feature extraction from websites. We have outlined the phishing feature extraction criteria in detail, and these rules were applied to extract features. SVM, NB, and ELM were employed to classify these features. Six alternative activation functions were applied to the ELM, which received the greatest accuracy rating.

References

[1] G. Canbek and . Sa ro lu, "A Review on Information, Information Security and Security Processes," Politek. Derg., vol. 9, no. 3, pp. 165–174, 2006.

[2] L. McCluskey, F. Thabtah, and R. M. Mohammad, "Intelligent rule based phishing websites classification," IET Inf. Secur., vol. 8, no. 3, pp.153–160, 2014.

[3] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Predicting phishing websites based on self-structuring neural network," Neural Comput.Appl., vol. 25, no. 2, pp. 443–458, 2014.

[4] R. M. Mohammad, F. Thabtah, and L. McCluskey, "An assessment of features related to phishing websites using an automated technique,"Internet Technol. ..., pp. 492–497, 2012.

[5] W. Hadi, F. Aburub, and S. Alhawari, "A new fast associative classification algorithm for detecting phishing websites," Appl. SoftComput.J., vol. 48, pp. 729–734, 2016.

[6] N. Abdelhamid, "Multi-label rules for phishing classification," Appl.Comput. Informatics, vol. 11, no. 1, pp. 29–46, 2015.

[7] N. Sanglerdsinlapachai and A. Rungsawang, "Using domain top-page similarity feature in machine learning-based web phishing detection," in3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010, 2010, pp. 187–190.

[8] W. D. Yu, S. Nargundkar, and N. Tiruthani, "A phishing vulnerability analysis of web based systems," IEEE Symp. Comput.Commun.(ISCC2008), pp. 326–331, 2008.

[9] P. Ying and D. Xuhua, "Anomaly based web phishing page detection,"
in Proceedings - Annual Computer Security Applications Conference,ACSAC, 2006, pp. 381–390.

[10] M. Moghimi and A. Y. Varjani, "New rule-based phishing detection method," Expert Syst. Appl., vol. 53, pp. 231–242, 2016.