VULNERABILITIES DETECTION USING MACHINE LEARNING AND CYBER SECURITY

Dr. Deepak Sukheja, B. Hema Sri, A. Karthika, K. Kavya, Ch. Poojitha

VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad Department of Computer Science and Engineering

Abstract: Xss attacks, SQl Injection and Online Phishing are three of the most common Internet crime techniques. Checking URLs against blacklists of known phishing websites, which are generally built based on manual verification, is a fre- quent counter measure for Phishing website Detection and is inefficient. A typical Internet User will not be able to distinguish between XSS Attack and SQL Injection Attack. As a result, as the Internet's scale expands, technology and its components advance to give timely protection to end users. We present in this thesis an effective and versatile and malicious URL detection System with a rich collection of features that reflect many aspects of phishing webpages and their hosting platforms, as well as the detection of other vulnerabilities such as XSS Attack and SQL Injection using web scraping and a dummy code injection mechanism are used in an injection at- tack. Our system has high detection power and low error rates by Random Forest Algorithm. This is the first study that we are aware if that users effective feature col- lecting to conduct a Vulnerability Detection process of large-scale websites/URLs. The results of the experiments show that our system is capable of efficiently finding dangerous websites/URLs by the end user**.**

Keywords: XSS attack, SQL Injection, Online Phishing.

1 INTRODUCTION

Over the past years, we have seen enormous growth in internet connectivity and usage. The internet is everywhere, and it has become the main aspect for all types of users ranging from building communication between two common people to sharing highly confidential information between two countries. The Internet has become just like the five elements which are essential for human survival. Even in the financial sectors internet plays a vital role. The best way to maximize the online presence on the internet is through websites. The website had become the main expansion marketing strategy for business. Even for small-scale business, there is a website. The reason for this is that building a basic static website with- out any advanced security features takes less time and cost. Cross-site scripting is a vulnerability in web security that allows an attacker to jeopardize user engage- ment with a vulnerable application. It usually allows an attacker to disguise as a victim, perform any actions that a user out, and view any data that is available to the user. It also allows an attacker to bypass the same strategy of roots, which is intended to separate different websites one from the other.2

2 LITERATURE REVIEW

Some similar applications made their way into the research industry, the applications that address rescue were mainly focussed for these Unmanned Aerial Vehicle imple- mentations.

In [1], Waleed Ali and Sharaf Malebary discussed that, In the proposed PSO-based feature weighting, the website features were weighted with the ideal weights by using PSO to enhance the detection of phishing websites. But analysis and weighting of features using the ML algorithm takes longer.

In [2], Yazan Ahmad Alsariera, Victor Elijah Adeyemo, Balogun and Ammar Kareem Alazzawi, they identified that, To build an interpretable model, hybridization can be done using alternative decision tree techniques. It included four meta-learner models developed using the extra-tree base classifier outperform existing machine learning-based models in phishing attack detection which provides effective perfor-mance.

In [3], one of the top issues identified was We may create a model to detect various types of code and server-side injection flaws, such as SQL and cross-site request for- gery. The authors proved Crawler XSS performed better than other web vulnerability scanners in terms of accuracy and false-positive rate and also they included a clear and concise explanation of the potential risks associated with various DOM-baseddata and elements and how to mitigate them.
In [4], authors discussed on SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN. They detected by presenting a method of SQL injection detec- tion based on Elastic-Pooling CNN (EP-CNN), which can output a fixed two- dimensional feature map and improve the detection accuracy. And compared pro- posed method with traditional detection methods and showed that it outperformed them. The proposed method can detect SQL injection attacks in web applications, which are harmful, universal, and severe. But it failed to identify new attacks.
In [5] authors builded classifiers using machine learning algorithms to detect the se- verity of the XSS attack. Authors concluded that the Random Forest Classifier is the most effective algorithm for detecting XSS in web applications. And authors ad- dressed that there is a 0.34 false positive rate
In [6], a very much innovative approach using KMP string matching algorithm for detecting threats. But Un-stored patterns cannot be recognized by this algorithm. The focus of Arun Prasath, Akshay Mathur, Javaid.
In[7],is detecting a plugin that alerts whenever a malicious code in detection but it can only be used in html5 based websites.. The research done by Mahdieh. and col- leagues.
In [8] implemented Fuzzy rough set feature selection which can help to reduce the number of features that need to be analyzed and helps to make phishing attack detec- tion more robust to changes in the data. But Fuzzy rough set feature selection can sometimes result in overfitting and also it's a complex theory to understand.

In [9] Chen peng is discussed Instruction Set Randomization Which is used to detect second-order SQL injection attacks but it requires ongoing maintenance and updates.

In[10] Authors used Random Forest Classifier to detect the fault, and properties are retrieved from the URL and supplied to a random forest classifier. It employs only one trained model and necessitates the use of unsupervised learning.
All these applications are very much convergent towards the goal, but these are con- centrated in a particular field and they are narrowed down to a single purpose. But the current product/project is more of a feature centered implementation rather than an application oriented.

3 METHODOLOGY

In practical, four modules are involved in this project:
1. Detection of SQL Injection (Server side)
2. Detection of XXS vulnerability (Client side)
3. Detection of Phishing vulnerability (Social engineering)
4. Building a Graphical Interface (GUI). .

3.1 Detection of SQL Injection (Server side):

When a malicious user input is accepted by an application, it is subsequently used as part of a SQL statement to query a backend database.
To modify the query structure, an attacker can inject SQL control characters and command keywords
When these control characters are combined with typical SQL commands (e.g., SE- LECT, FROM, DELETE, etc.), data components from a backend database server can be accessed or retrieved.
We have installed packages for extracting URL forms before starting this module. The packages BeautifulSoup, URL, Request and pprint have been used.
We used the BeautifulSoup library to extract all form tags from HTML. This extrac- tion takes place in the function get allforms(). A single form tag object argument is added in the other function getting formdetails() and only analyses applicable infor- mation.

3.2 Detection of XSS vulnerability (Client side):

An XSS vulnerability occurs if web applications take user data and dynamically add this data to websites without evaluating the information properly before.
The packages BeautifulSoup, URL, Request and pprint have been used.
The function get_formdetails collects all the details of forms (text, search, etc.) fromthe website.
The injection of JavaScript code is done by using the above get_formdetails() func- tion and submits the JavaScript code. It is considered vulnerable if the inserted JavaS-cript code is successfully executed.

3.3 Detection of Phishing vulnerability (Social Engineering):

In this module, we used a dataset of approximately 2000 web URLs. We split the URL in half for training and testing, with 80 percent going to training and 20 percent going to testing. A few features must be considered while detecting a phishing web- site. The features are

1. Having a long URL that hides the suspicious part of the URL
2. URL having "@" symbol
3. Having a "//" symbol in the URL, which redirects the page to a malicious one.
4. Adding Prefix or Suffix Separated by "-" to the Domain
5. Sub-Domain and Multi Sub-Domains

3.4 Building Graphical User Interface:

GUI was developed using python and PyQt5 package. We title on the top using qt widgets module and we also incorporated Qpixmap. For user input, we made an input box in the center using Qlabel. A button is made to toggle the running of all the func- tions. Finally, an output box is created on the bottom using Qlabel and for cursor ele- ments we used Qcursor.

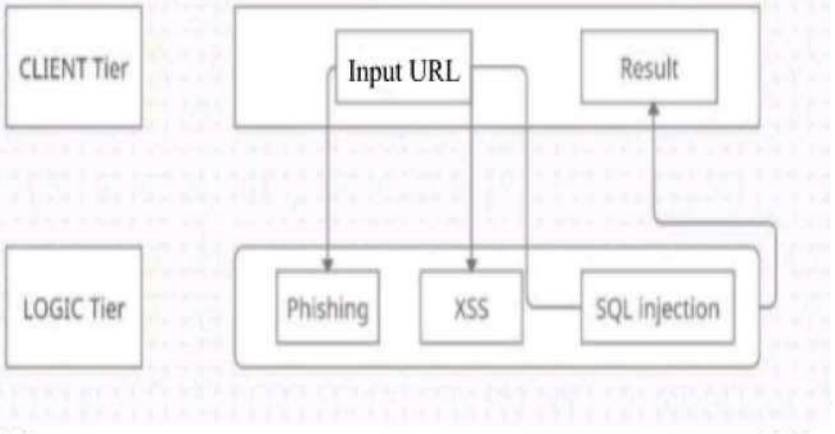

Fig1: User Interface

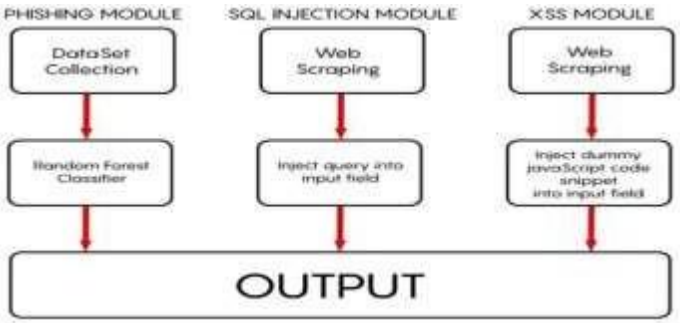

Fig2: Architecture

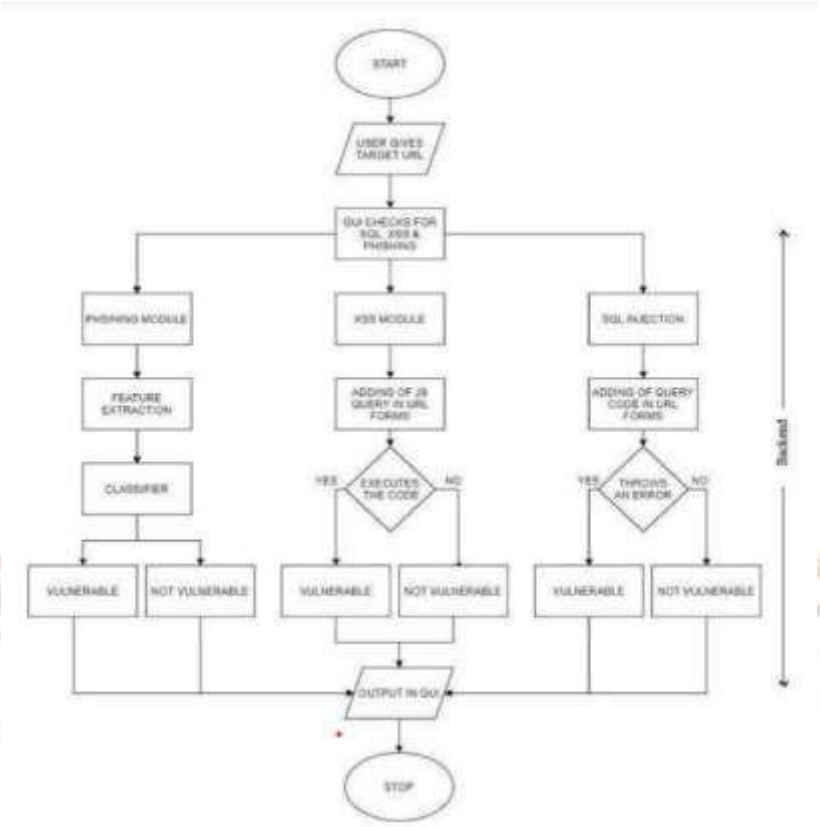

Fig3: Workflow

4 RESULTS

Designed GUI can detect, given url is free from attacker or not. This is a User friendly application where end user need not learn any crime techniques

Fig4: GUI

5 CONCLUSION

With continuously evolving methods of hacking the detection of vulnerabilities be- comes crucial. In this project we have created an GUI that can detect Phishing, Pay- load SQL injection and Reflected XSS vulnerability. The GUI can be used by anyone, even if they have no prior coding experience. It allows users to check any link just by copying it into the GUI and gives instant results. Random forest algorithms are used to detect phishing. Injecting SQL payloads and malicious scripts into the input fields of the given URL is the test of and detect the vulnerability of SQL Injections and Reflected XSS. When we get a particular output on load or script injection, the URL is classified as vulnerable to SQL injection or Reflected XSS. The GUI application can detect the given vulnerabilities in under 30 seconds with high accuracy and reliability.

6 REFERENCES

1. Waleed Ali and Sharaf Malebary, ‘‘Particle Swarm Optimization-Based Feature Weighting for Improving Intelligent Phishing Website Detec-tion,’’ https://ieeexplore.ieee.org/document/91 21227

2. Yazan Ahmad Alsariera; Victor Elijah Adeyemo; Abdullateef Olu- wagbemiga Balogun ‘‘AI Meta-Learners and Extra-Trees Algorithm for the Detection of Phishing Websites,’’ https://ieeexplore.ieee.org/document/91 54378

3. Raima Banerjee and Aritra Baksi, ‘‘Detection of XSS attacks in a web application usingMachine LearningClassifiers,’’https://ieeexplore.ieee.org/document/92 70052

4.

5. Arun Prasath Sivanesan, Akshay Mathur ‘‘A google chromium browser extension for detecting XSS attack in HTML5 based websites,’’ https://ieeexplore.ieee.org/document/85 00284

6. Mahdieh Zabihimayvan and Derek Doran, ‘‘Fuzzy rough set feature se- lection to enhance Phishing attack Detection,’’ https://ieeexplore.ieee.org/document/88 58884

7. Chen Ping, ‘‘A second order SQL injection detection method,’’ https://ieeexplore.ieee.org/document/8285104

8. Xin Xie and Chunhui Ren, ‘‘ SQL Injection detection for web applica- tions on Elastic- pooling CNN,” https://ieeexplore.ieee.org/document/88 77739