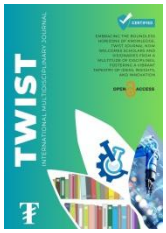
DOI https://doi.org/10.5281/zenodo.10049652#273

TWIST, 2024, Vol. 19, Issue 3, pp. 656-666

TWIST

Journal homepage: www.twistjournal.net



An Enhanced Phishing Detection System in Online Transactions

T. O. Oyegoke*
Department of Computer Science and Engineering, Obafemi Awolwow University, Ile-Ife, Nigeria
[**Corresponding author*]

A. O. Amoo
Department of Computer Science and Engineering, Obafemi Awolwow University, Ile-Ife, Nigeria

J. Aigberua
Department of Computer Science and Engineering, Obafemi Awolwow University, Ile-Ife, Nigeria

Abstract
Phishing attacks have become an ever-increasing challenge during online transactions via various payment platforms thereby making it important for cyber security experts to tackle and solve this problem. Hence, this research aimed to address this challenge by creating a web-based system that not only detects phishing attempts but also empowers users to navigate the digital landscape securely. The research focused on leveraging cutting-edge technologies such as JavaScript, Node.js, HTML, and CSS to build a robust and user-friendly phishing detection plugin. This system provided real-time alerts and insights when users interact with potentially malicious websites. Evaluation of the system's user experiences shows that 68.42% of respondents found the system's user experience and ease of installation exceptional, with 20% rating it as good. The system's effectiveness in detecting phishing threats received a high satisfaction rate of 73.68%, and an impressive 84.21% responsiveness score indicates its efficiency in delivering timely responses. Moreover, 78.95% of users expressed satisfaction with the system's overall performance, and 84.21% would recommend it to others based on their positive experiences. The average satisfaction index of 77.89% confirms the system's quality and effectiveness. Overall, this research significantly advances online transaction security by developing an innovative phishing detection system. From problem identification to system development, validation, and user reception, this research contributes to safer digital interactions and trust-building in the dynamic technology landscape.

Keywords
Online transaction, Cyber-attack, Web-based system, Phishing, Malicious

INTRODUCTION

Phishing is a type of social engineering attack that is regularly used to gain user information such as login credentials and credit card details. It is a method that relies on a user's insufficient vigilance and understanding of the internet which takes place when an attacker poses as a trustworthy source in order to trick a victim into opening an email, instant message, or text message that contains malicious content [1, 2]. The recipient is then tricked into clicking on a malicious link by being led to believe it is harmless.

Due to the dramatic rise in the number of people using the internet, individuals' private information is increasingly likely to be disseminated through this medium. Because of this, con artists have access to a significant amount of personal data as well as financial activities. Phishing is a form of cybercrime that has proven to be highly effective for perpetrators, as it enables them to deceive victims and steal sensitive information. Phishing attacks typically take advantage of people who lack digital or cyber ethics or who are poorly trained in addition to exploiting technical weaknesses in order to achieve their objectives [3]. As a result of the fact that an individual's susceptibility to phishing differs depending on their characteristics and their level of knowledge, phishers take advantage of human nature in the majority of their hacking attempts rather than advanced technologies. There is a lack of knowledge on which ring in this chain is initially breached, despite the fact that the vulnerability in the information security chain is more commonly attributed to humans than it is to technology. According to research conducted by [4] in 2020, certain personal qualities

render some individuals more responsive to a variety of enticements. For instance, those who tend to obey rules and regulations more than others are more likely to fall victim to a business email compromise (BEC), also known as a phishing scam, which pretends to be from a financial institution and demands immediate action because they believe the email to be genuine. Greed is an additional human vulnerability that an attacker may try to take advantage of, as shown in emails that offer massive discounts, gift cards, and other similar offers [5].

Since the first instance of phishing was published in 1990, it has evolved into a more intricate form of attack and become a vector for more widespread cybercrime. Phishing is currently one of the most common forms of fraudulent activity that may be committed online. According to [6], phishers use social engineering tactics to direct consumers to malicious websites once they have opened an email and followed a link contained within the message [7]. Alternately, attackers may carry out their operations by utilizing different technologies, such as Voice over Internet Protocol (VoIP), Short Message Service (SMS), and Instant Messaging (IM). In addition, in order to improve their chances of success, phishers have transitioned from sending generic bulk emails to a wide range of potential victims to "spear-phishing," which is a more selective kind of phishing.

According to the resuly of the survey of cybersecurity breaches conducted in 2022 in the United Kingdom [8] phishing attacks were shown to be the most common sort of cybersecurity breach. Even while these attacks affect individuals as well as companies, the losses sustained by organizations are far greater. These losses include the cost of recovery, the loss of reputation, the fines imposed by information laws and regulations, and the decreased productivity.
The number of phishing assaults that were discovered in the second quarter of 2021 was noticeably higher than the number that was recorded in any of the three quarters that came before it. According to a study from the Anti-Phishing Working Group [9], this number was higher than it was in the previous quarter while in the first quarter of 2022 it was more than it was in the previous quarter, which demonstrates that the number of phishing assaults is on the rise. These data have demonstrated that the number of phishing assaults has steadily increased over the past several years, that they have become more sophisticated, and that they have garnered increasing attention from cyber researchers and developers who are working to detect and minimize the impact of these attacks.

There are many different anti-phishing solutions already in the market, such as SafeToOpen, PhistNet, Anti-phishing toolbars, Email Filters, NetCraft, and so on, which can be utilized to protect users from falling victim to fraudulent activities. The technologies have been able to bring the overall success rate of phishing attempts down to a lower percentage. NetCraft is the most popular, modern, and effective browser plugin that can detect phishing links, according to [10]. However, it does have certain drawbacks, such as an inability to blacklist or block bad URLs. As a result, the severity of phishing problems and mitigation techniques will be analyzed and developed by providing detailed insights into the phenomenon of phishing in terms of phishing definitions, current statistics, anatomy, and potential countermeasures for the detection of phishing links by the development of an enhanced phishing detection system in online transaction environments. This will be done by providing detailed insights into the phishing phenomenon in terms of phishing definitions, current statistics, anatomy, and potential countermeasures for the detection.

RELATED WORK

Scammers engage in phishing when they create a website or send an email in an attempt to fraudulently acquire sensitive information, such as a user's credit card number or password for online banking [7]. The massive impact that phishing websites have on the banking and e-commerce industries makes them a serious issue. In modern phishing assaults, attackers might utilize a variety of techniques that allow them to pose as a trusted source. Attackers use phishing to get victims to reveal sensitive information (such login credentials) on a bogus website [11]. As a result, protecting your website from phishing assaults is crucial [12]. [13] examined several phishing detection tactics, with an emphasis on language techniques and machine learning technologies, in a study that encompassed works up to 2015. Based on their findings, they suggested a wide variety of anti-phishing toolbars. The eBay toolbar, for instance, features a color-coded tab that notifies users of potentially harmful information on the pages they're viewing. SpoofGuard, an Internet Explorer add-on, is another option; it monitors the pages you visit and alerts you if it suspects a page is a spoof. The existed work as regards phishing detection in online transactions are based on the following:

Content-based approaches

[12] present a model that combines fuzzy logic and data mining to characterize the e-banking phishing website. E-banking phishing websites have the greatest predicted success rate of any industry, at 83.7%, according to their methodology. Due to its singular reliance on language, it is insufficient for detecting fake online banking portals. The method's potential for error in identifying e-banking phishing sites stems from its reliance on text elements alone.

[12]’s association classification datamining technique is the backbone of a sophisticated and efficient model. With this approach, we are able to identify the features and criteria that define a phishing website. They use six different techniques to extract text features in order to evaluate 120 websites for trustworthiness. Each method is tested for both speed and precision. By applying them to a large dataset, they demonstrate the usefulness of each classification method and show that they outperform traditional classification algorithms by a margin of 12.62 percentage points. The proposed method suffers from decreased accuracy because they cannot apply diverse pruning strategies to eliminate the rules that erroneously classify.

[4] present a neuro-fuzzy system based on fuzzy rules for real-time detection of fraudulent, phishing, and legitimate websites. Their results show improved accuracy with 2-fold cross-validation, with a true positive rate of 98.5% and a false positive rate of 1.5%. Five inputs are used to demonstrate the power of the neuro-fuzzy system for detecting phishing sites in real time [14]. Their comparison mechanism outperforms that of a competing study due to its greater efficiency. However, the proposed approach is overly dependent on text-based features, and the addition of other features and/or adjustments to the parameters could improve accuracy. Based on these observations, the system will implement an intelligent phishing detection method.

Also, [4] develop an add-on for Internet Explorer that, once installed, operates quietly in the background, validating the user's requests against a database in real time. Their neuro-fuzzy approach to detecting phishing websites has as inputs the rules for safe websites, user profiles, PhishTank, user-specific websites, pop-up windows, and user credentials profiles. Their research provides evidence for better phishing detection in real time. However, they concentrate solely on aspects of the text. They might, therefore, enhance their precision by analyzing additional image frames and attributes. There are over 600 inputs and over 300 functions incorporated into the toolbar. This data is given to the neuro-fuzzy feature extractor algorithm. The toolbar will do a request comparison to the elements of the malicious website, create a replica of the website, and then display a red text directive if the request is malicious. In tests, the proposed solution showed a 96% true positive rate with a 4% false positive rate. We evaluate our results against those of other services like SpoofGaurd, Netcraft, EarthLink, Google, Cloudmark, IE8, TrustWatch, and McAfee. The toolbar is great for instantaneous accuracy. In particular, their major contribution to the sector will be the introduction of a ground-breaking voice-generated user warning interface strategy for toolbar identification.

Visual similarity approaches

[15] devise a solution that accounts for the degree to which two websites share aesthetic similarities. Their solution makes use of four metrics: web page framework, block level similarity, overall style similarity, and total similarity. The block level similarity calculates the overall similarity in the feature set by weighting the sum of the individual similarity between two websites. Therefore, they provide special consideration to the visual appeal of the page by carefully considering elements such as font size, color, backdrop, line spacing, and text alignment. The histogram-normalized correlation between the two sites is then calculated based on the collected characteristics. This method has potential, but further work and testing are required before it can be implemented fully [16].

To counter phishing scams, [17] developed a technique based on visual cryptography. As part of a (2, 2) visual cryptography scheme, the user generates two picture "shares." One half of the photograph is stored on the user's device for future reference, while the other half is permanently uploaded to the website. The user must verify the two photos match after each login attempt. Since they only used a tiny subset of websites in their test, the results should not be trusted; a more comprehensive analysis is needed.

[18] developed a similar plugin called SpoofGuard. SpoofGuard is an Internet Explorer plug-in. The add-on accesses the user's cache, cookies, and history stored in Internet Explorer in the profile folder. One of the logs at popular email services like Hotmail and Yahoo Mail is now read-only. The other two logs are the hash password and hash image. The plugin will utilize this data to verify the site's authenticity. However, a toolbar plugin is available to protect users from phishing attacks. The plugin uses a symptom-based approach, checking both the currently-viewed page and any other pages that share a suspiciously similar domain name for indicators of phishing, such as hidden links. Their solution's security relies heavily on the password-checking mechanism. The image checker's hash algorithm could use some fine-tuning if it's going to be able to detect small alterations in photographs. [18] explain that an alarm is set off by the accumulation of symptoms.

A similar approach is proposed by [19], who recommend using Google's search-by-image API to confirm a website's legitimacy. Using Google's image search functionality, one can find and obtain a list of results that are pertinent to an uploaded image. Consequently, they use four different heuristics to identify fraudulent websites in search engine results. The conclusion establishes the site's credibility on the basis of its ranking in search engines. The first four results on a search engine's results page are more trustworthy than phishing sites because of their higher page rank. They use a thousand websites to demonstrate the efficacy of their approanksch. In this study, we found that the genuine positive rate was 97.2%, with only 2.8% of results being false positives. Though precise, the strategy can only analyze data from the favicon. When a malicious website uses the same favicon as a valid one, it can fool detection systems into thinking it is authentic. Their approach is only somewhat reliable because it automatically labels a website as malicious if it does not have a favicon. Since the limitations of the technique could be damaging as phishing efforts expand, a better result could be attained by combining this approach with other phishing detection features, such as text and frame structure.

Heuristic based approaches

[20] proposed a detection model which includes an extension software and a processing Algorithm which was combined with Deep Learning Model. They went ahead to combine different ML algorithms with a malicious URL dataset. The proposalof this study can only warn users of malicious websites; it does not have the capability of blocking such website from the user or blacklist them from subsequent occurrence to the user.

[21] propose a browser add-on that combines the TF-IDF3 algorithm with another heuristic element of websites to counteract phishing. Their CANTINA content-based approach is examined; it employs TD-IDF to identify phishing

websites. Their data include 97% positive identifications and 3% erroneous positives. Unfortunately, the strategy was not successful in differentiating spam from authentic websites.

[22] employ a feed-forward neural network to identify phishing emails by analyzing their structure and any external links they contain. According to their findings, the neural network is quite good at spotting phishing emails, with a nearly nonexistent false positive rate. However, their results are often inaccurate. They wrote a script to extract text features from email bodies, build a feature vector set, and store the optimal value in a single text file. The use of such features can be maximized through normalization prior to applying a machine learning method. They use 4,202 legitimate emails and 4,560 spam emails in their study. The neural network employed achieves a true positive classification rate of 95% in detecting phishing emails, with only a small amount of misclassification of non-phishing emails. Its implementation is superior to those of other machine learning algorithms since it yields an understandable result and justification for the classification.

Finally, [23] provide a method for discovering and combating phishing that relies on web page similarities and URLs. The LinkGuard4 algorithm compares the original URL seen by the user with the extracted URL used to redirect the user. If URL-based detection fails to identify phishing content, the system will turn to visual similarity-based identification. Their test result is not robust due to the small number of websites included in the experiment, so further investigation is required to improve accuracy.

RESEARCH METHOD

This session described the procedures, techniques, methods and resources employed to accomplish the research's objectives as stated in chapter one of this project. In this session also, there is additional discussion of the SDLC's methodology and phases, which include requirement gathering, modeling, implementation, testing, and deployment. This chapter also includes a presentation of the many tools used by the SDLC's methodologies and phases. Along with the proposed system's usecase and static model, this chapter introduces Unified modeling language as the language of choice for describing the model of the system. Ultimately, the current framework for the research activity is highlighted, and a new framework is offered to serve as a replacement.

System Design

During the design phase of the Software Development Life Cycle (SDLC) for the "Enhanced Phishing Detection System in Online Transactions," the software's architecture, interfaces, and database are defined based on the gathered requirements. Detailed design specifications are created, including diagrams illustrating the system's overall structure and specific components. Design activities also consider scalability, performance, and security aspects. System Architecture Design involves creating high-level diagrams showing the system's structure and component interactions. The design phase ensures a well-structured, efficient system blueprint for subsequent implementation and testing stages. It contributes to a reliable and user-friendly phishing detection system, enhancing cybersecurity in online transactions effectively. The design of this project makes use of object-oriented modeling, a technique to analysis and design of software systems that relies on Object constructs. Object-oriented languages facilitate the implementation of object-oriented models. In order to specify the object model, the Uniform Modeling Language is the language of choice.

In the development of the "Enhanced Phishing Detection System in Online Transactions," UML (Unified Modeling Language) played a crucial role in the system design phase. UML diagrams, such as class diagrams, sequence diagrams, and deployment diagrams, visually represent the system's design and interactions. Class diagrams define the static structure, outlining relationships between components. Sequence diagrams capture dynamic behavior during phishing detection, ensuring efficiency and accuracy. Deployment diagrams illustrate physical deployment across platforms for seamless integration. UML facilitates communication and collaboration among the development team and stakeholders, providing a standardized visual notation for better understanding. It aids in creating a well-structured, user-friendly, and robust system that meets specific requirements and enhances cybersecurity in online transactions. Utilizing UML fosters efficient development and implementation of the phishing detection system, empowering users to protect against phishing threats effectively. The following UML were used in the course of the design.

Class Diagram

A class diagram serves as a potent tool for comprehending a system's structural complexity, particularly at a macro level. It unveils the intricate tapestry of relationships woven between distinct classes, illuminating the ways in which they harmonize and collaborate within the broader context. Capturing the essence of Figure 1, the diagram elegantly weaves together classes like "user," "attacker," "browser," "transaction," and the "proposed system," each tethered by their interrelationships and interdependencies. This visual representation serves as a repository of insights, fostering a comprehensive grasp of the system's building blocks and their orchestrated interplay. Figure 1 shows the class diagram harnessed in the development of the advanced web browser extension, casting a spotlight on the intricacies of the project's architecture.

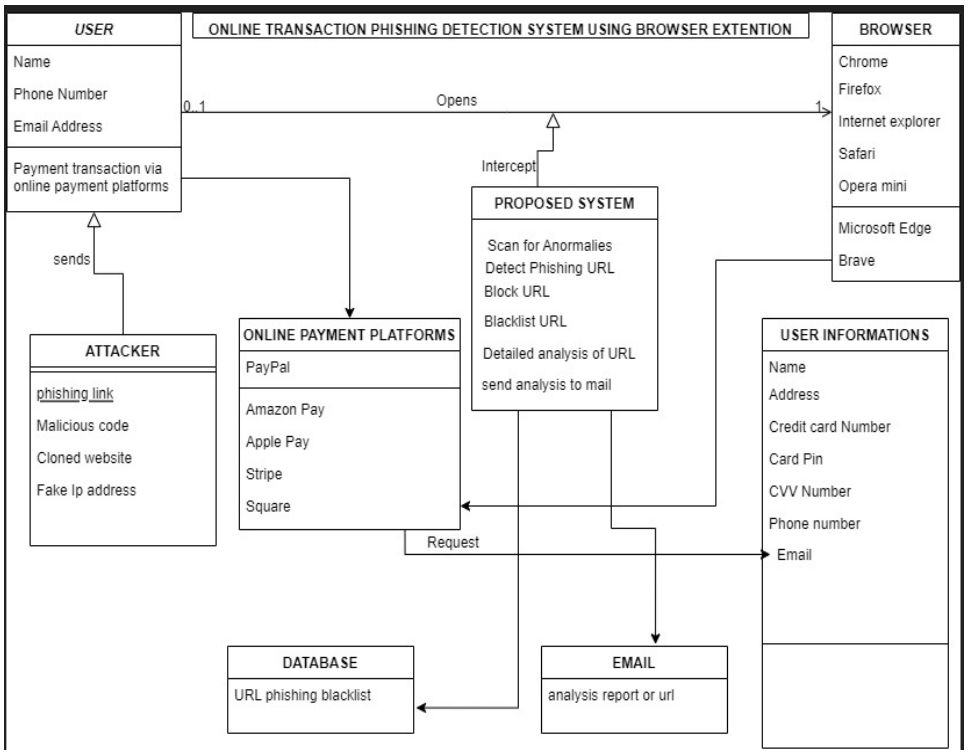


Fig. 1 Class diagram for browser extension for phishing detection

System Framework

The intricate foundations of the system's processes are brought into focus by this paradigmatic framework, which enables them to be seen more clearly. It explains how the system works, shedding light on the dexterity with which the system navigates the complex terrain of online interactions. Whether it is during the process of online transactions or the browsing of the web, the framework serves as a sentinel, expertly identifying and minimizing the threat posed by phishing links that may be hiding in the background. In order to acquire a comprehensive grasp of the path that the project has taken within the context of phishing detection online, a comprehensive overview of the research's existing framework for online phishing detection is encapsulated in Figure 2, offering a holistic understanding of the project's trajectory in this domain.

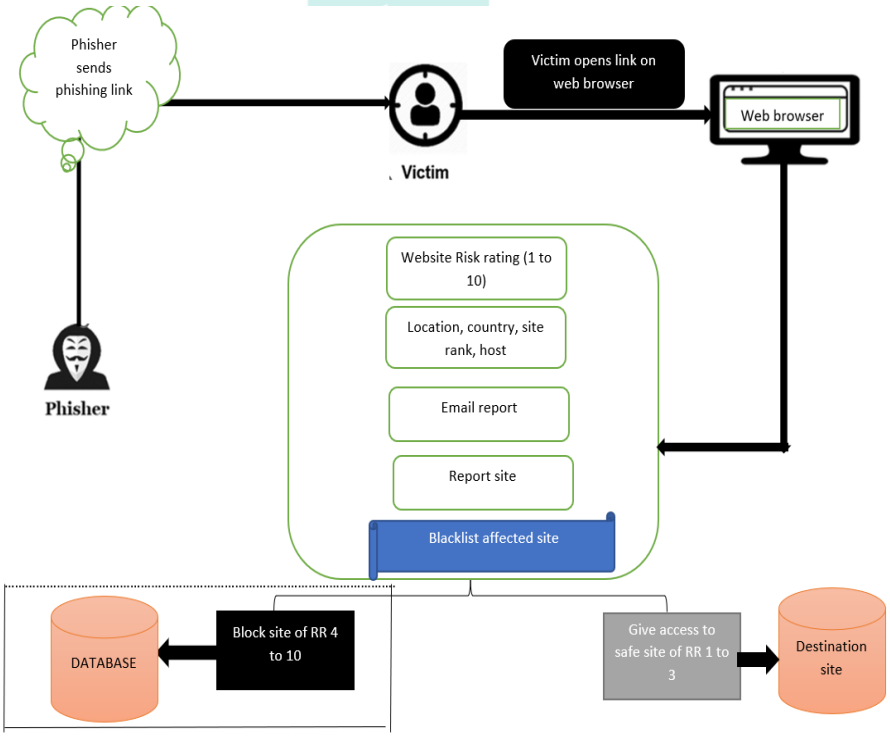


Fig. 2 The developed system Framework for phishing detection using web browser extension

From the figure above, machine learning can be used to detect phishing attempts by analyzing patterns and features in emails, text messages, and website URLs. One common approach is to use supervised learning algorithms to train a model on a dataset of labeled phishing and non-phishing examples. The model can then be used to classify new, unseen

examples as phishing or non-phishing. Some of the features that can be used to train a model for phishing detection include the sender's email address, the URL of any links in the message, the presence of certain keywords, and the structure and layout of the message. Another approach is to use unsupervised learning algorithm like clustering to detect new kind of phishing emails. There are also other methods like using browser extensions to detect phishing attempts by analyzing the URLs of websites in real-time, to block or warn the user of any potential phishing attempts.

Figure 2 offers a visual representation of the system framework specifically designed for detecting phishing activities through the implementation of a web browser extension. This graphical depiction encapsulates the intricate architecture and operational dynamics underpinning the framework's functioning. This visual aid serves as a comprehensive guide, elucidating the systematic process by which the proposed solution identifies and mitigates phishing threats.

RESULTS AND DISCUSSION

This session explored the various ways in which the design presented in the preceding chapter can be put into action. It entails describing the proposed program in great detail, including how it was designed, tested, and implemented for the end customers. The many tools that were utilized during the installation process are also addressed. In the end, both the system documentation and the user documentation, as well as the requirements that are necessary for implementation, are investigated.

The Developed System Modules

The developed system modules are:

i. Website URL: this displays the url of the site to be visited.
ii. The JavaScript code captures the URL of the website the user intends to visit. Node.js processes this information on the backend, enabling real-time analysis of the URL, comparing it against known phishing sites in the system's database (blacklist). If a match is found, the web browser plugin, with its HTML and CSS interface, displays an alert message to the user, warning them of the potential phishing risk associated with the site they are about to visit.
iii. Website information: this gives the detailed information about the website to be visited.
iv. JavaScript collects detailed information about the website the user intends to visit, such as the website's content, structure, SSL certificate status, and domain reputation. This information is then sent to the backend powered by Node.js for real-time analysis. The backend processes this data, leveraging machine learning models to examine the website's characteristics and compare them against known phishing patterns stored in the system's database (blacklist).
v. Server type: this displays the type of server that hosted the url
vi. JavaScript gathers information about the type of server hosting the URL of the website the user is about to visit. This information is then sent to the backend implemented in Node.js for real-time analysis. The backend processes this data and uses various techniques to determine the type of server hosting the website, such as inspecting server response headers or conducting DNS lookups. The analysis results are then communicated back to the web browser plugin, which utilizes HTML and CSS to display the server type information to the user.
vii. Harmless: this displays in percentage how harmless the site is.
viii. JavaScript and Node.js work together to evaluate the potential harm associated with the website the user intends to visit. The JavaScript code collects relevant data about the website's features, content, and reputation, which is then sent to the backend for analysis. The Node.js backend uses machine learning models to assess the website's characteristics and compare them with known harmless websites in the system's database (whitelist). Based on this analysis, the backend calculates a percentage representing how harmless the website appears to be.
ix. Malicious: This segment serves as an illuminating indicator, presenting users with a quantified representation that gauges the level of malicious intent inherent within a particular website. The "Malicious Percentage" feature encapsulates a meticulous evaluation, drawing from an intricate amalgamation of data points and analysis (Shekokar et al, 2022). The significance of this "Malicious Percentage" lies in its ability to empower users with actionable insights. In a world where the digital landscape can be treacherous, this percentage becomes an invaluable tool, enabling users to make informed decisions regarding their online interactions. The higher the percentage, the more pronounced the red flag; conversely, a lower value signifies a site with a diminished likelihood of harboring malicious intent (Syafiq et al, 2022).
x. JavaScript and Node.js collaborate to assess the potential maliciousness of the website the user intends to visit. JavaScript collects relevant data about the website's features, content, and historical behavior, which is then sent to the backend for analysis. The Node.js backend employs machine learning models to evaluate the website's characteristics and compares them with known malicious websites in the system's database (blacklist). Based on this analysis, the backend calculates a percentage representing how malicious the website appears to be. The web browser plugin, using its HTML and CSS interface, displays this maliciousness percentage to the user, offering valuable insight into the potential risks associated with the website and assisting the user in making informed decisions during online transactions.
xi. Suspicious: this displays in percentage how suspicious the site is.

xii. JavaScript and Node.js work collaboratively to determine the level of suspicion associated with the website the user intends to visit. JavaScript collects relevant data about the website's characteristics, behavior, and other attributes, which is then sent to the backend for analysis. The Node.js backend utilizes machine learning models to assess the website's features and compares them with patterns of known suspicious websites in the system's database. Based on this analysis, the backend calculates a percentage representing how suspicious the website appears to be.

xiii. Email report: this sends the report analysis to user mail for reference.

xiv. In the phishing detection system, after performing the analysis of the website, JavaScript and Node.js work together to generate an email report summarizing the results. The JavaScript code formats the analysis data and sends it to the Node.js backend. The Node.js backend is responsible for handling the email functionality and communication with the email server. Once the backend receives the analysis data, it composes an email containing the relevant information, such as the URL, harmlessness percentage, maliciousness percentage, suspiciousness percentage, and any other findings from the analysis. The email report is then sent to the user's provided email address for reference and record-keeping. By providing an email report, the phishing detection system offers users a convenient way to review and track the results of the website analysis, ensuring they have a detailed record of potential risks associated with the website they attempted to visit during their online transactions.

xv. Local Database: this blacklist all affected sites.

xvi. In the phishing detection system, a local database is implemented to maintain a blacklist of all affected sites that have been identified as potentially malicious or suspicious. The database is hosted on the backend, powered by Node.js, and is continuously updated based on the results of real-time analysis. When the system identifies a website as suspicious or malicious, the JavaScript code sends the necessary data to the Node.js backend. System Interface.

How the System Works with A Safe Url

Illustrating a prime illustration of a secure URL, such as www.gmail.com, underscores the foundation of the developed system's capabilities. This multifaceted system serves as an information-rich portal, meticulously dissecting the nature of a site's content and assessing its potential harm, malicious intent, or suspicion, as artfully depicted in the first part of Figure 3.

As this comes into view, it showcases the system in action, testing its proficiency on a secure URL. The ensuing results paint a vivid portrait: a staggering 97% rating for harmlessness, juxtaposed against a complete absence of malicious or suspicious attributes. A mere 17% rate of undetected elements reaffirms the site's steadfast security. This comprehensive analysis decisively concludes that the tested site stands as a secure haven, beckoning users to traverse its digital domains without trepidation.

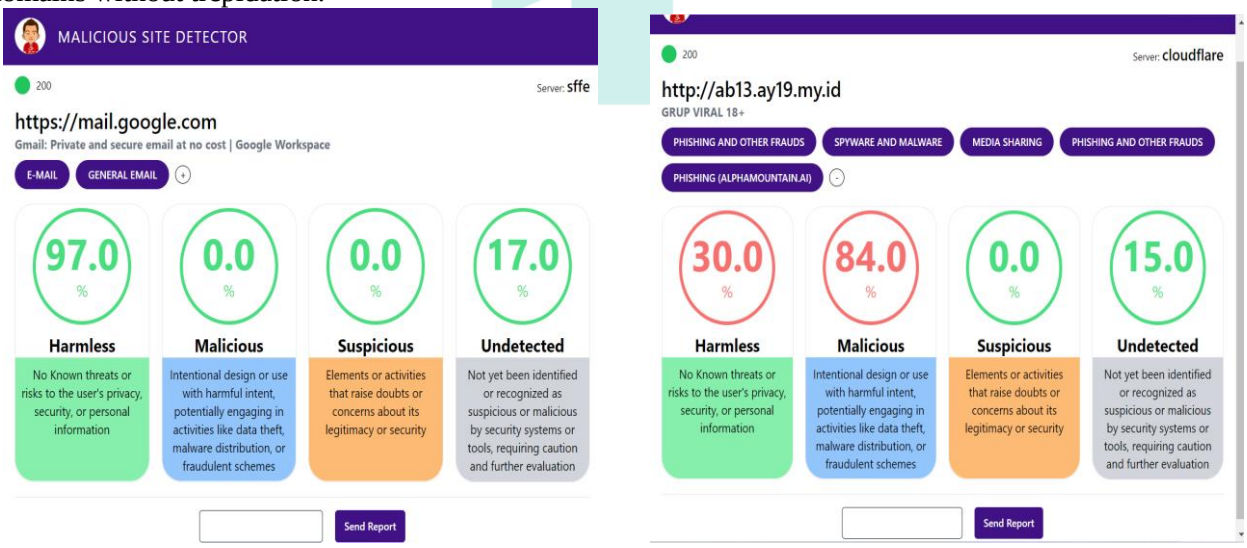


Fig. 3 Testing the system with a safe URL

The second part of Figure 3, which unfolds as an integral visual aid, amplifies the system's prowess as it takes on the challenging task of evaluating the aforementioned URL. The findings paint a compelling picture – a mere 30% rating for harmlessness pales in comparison to an overwhelming 84% rating for malicious intent. This verdict underscores the gravity of the situation, accentuating the site's susceptibility to nefarious activities.

Blocking Affected Url

The implementation of the system has culminated in an effective and decisive response to the threat posed by unsafe sites. Once identified, these sites are promptly intercepted and blocked by the system's robust defenses. However, it's important to acknowledge that users still retain the option to venture into such domains at their own discretion, albeit with a

heightened awareness of the inherent risks. While access remains open for those who choose to tread the path, it's imperative to underscore the caveat that accompanies such decisions – navigating into these unsafe territories is akin to a venture fraught with potential dangers. The system's blockage is a proactive measure, indicative of the commitment to user safety, and the subsequent choice to proceed falls squarely upon the user's shoulders. Figure 4 shows the blocking of users from accessing a suspected url.

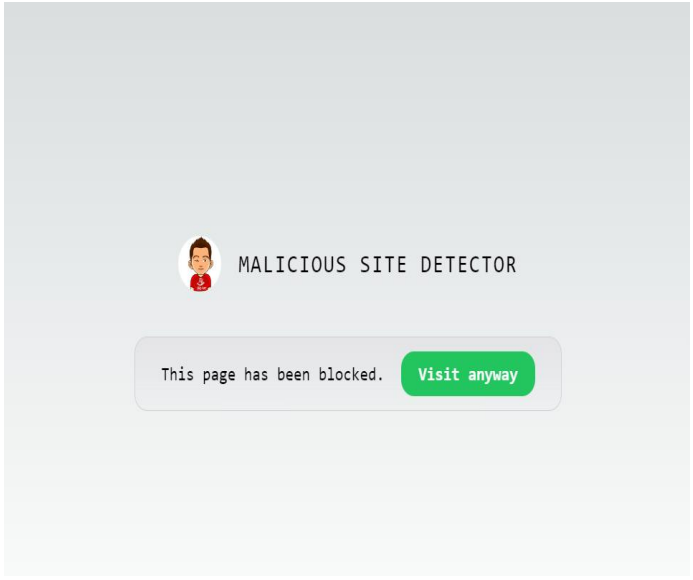


Fig. 4 Blocking users from accessing a suspected url

Local Database for Blacklisting Affected Url

Within the intricate framework of the system's architecture, a local database emerges as a pivotal repository, preserving a comprehensive record of all URLs traversed by users. A defining feature of this repository is its steadfast permanence – once a URL has been intercepted and blocked by the system's built-in defenses, its status as a blocked entity remains unalterable within the database's confines. The modus operandi of this mechanism hinges on a proactive fetching process, driven by the system's inherent intelligence.

As users encounter URLs in their digital journeys, the system engages in a judicious routine: it retrieves the URL from the local database and conducts a meticulous verification. If the URL has been designated as blocked in the past, its status endures, maintaining its shielded status. Users, presented with the choice, retain the option to proceed at their discretion, yet the barrier of caution persists.

Conversely, for URLs that have yet to be embroiled in the realm of blockage, a seamless and expedient pathway unfolds. Users encounter an environment of convenience, as these URLs extend an invitation for exploration, free from the shackles of interception. Figure 5 shows the local database for storing affected site for blacklisting.

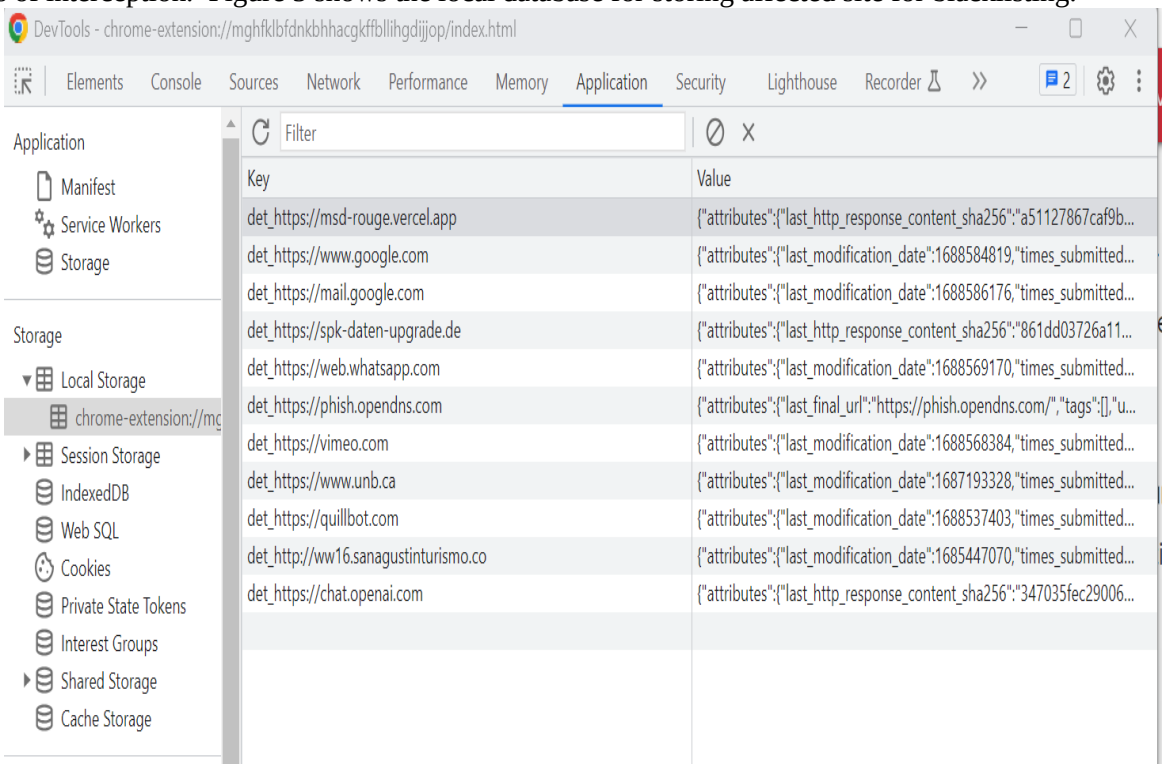


Fig. 5 Local database for storing affected site for blacklisting

System Performance Evaluation

To ensure the robustness and effectiveness of the newly developed system, a meticulous validation process was undertaken. A representative sample of the system's users was engaged, soliciting their insights through a structured assessment mechanism. This validation endeavor was meticulously orchestrated, bearing a dual focus on gauging the system's functional prowess and its ability to streamline the elimination of redundant data. To facilitate this evaluative endeavor, a Google form was meticulously curated, acting as a conduit for users to provide their perspectives while interacting with the system. Crafted with precision, this form was designed to unravel a comprehensive panorama of the system's capabilities and its efficacy in mitigating data redundancy. The formulation of the Google form was a pivotal step in translating conceptual goals into tangible evaluation metrics, thereby laying the foundation for an empirical assessment. The data thus gathered was subjected to an incisive analysis employing the power of Power BI – a sophisticated analytical tool. Figure 6 stand as testaments to this analytical journey, which shows an overall user experience of this system.

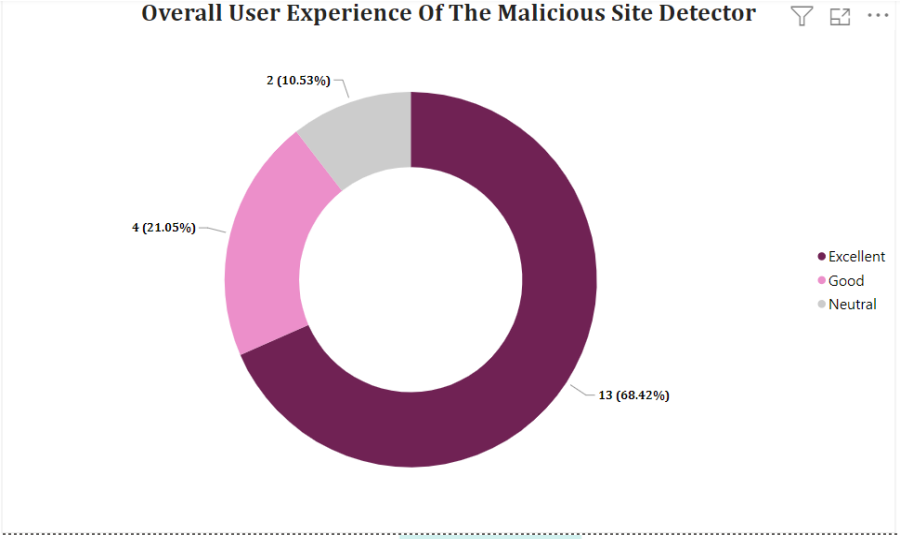


Fig. 6 User experience of malicious site detector

Performance Evaluation and Result Discussion

The study's findings provide a compelling glimpse into the user experience landscape, unveiling a tapestry of perceptions and satisfaction levels. A remarkable 68.42% of the respondents showcased an exceptional user experience, citing seamless ease of installation as a hallmark. Another 20% lauded the system with a commendable "good" rating for their experience, while a modest 10% expressed uncertainty, possibly a testament to the varied spectrum of user engagement.

The system's efficacy drew an equally resounding endorsement, garnering a remarkable score of 73.68% from the respondents. This affirmation underscores a high degree of satisfaction with the system's performance, as users attest to its successful execution of its intended purpose. Moreover, the system's responsiveness stands as a testament to its agility, with an impressive 84.21% of users affirming its timely and efficient interactions.

With a performance satisfaction score of 78.95%, the system secures yet another accolade, cementing its ability to meet user expectations and deliver on its promises. An intriguing insight arises from respondents' previous interactions with the malicious site detector: a resounding 84.21% express an enthusiastic inclination to recommend the system to their peers, marking a strong vote of confidence in its utility and reliability.

Synthesizing the cumulative impact of these percentages – 68.42%, 73.68%, 84.21%, and 78.95%, all supplemented by the formidable 84.21% recommendation rate – yields an average satisfaction index of 77.89%. This metric stands as a resounding endorsement, significantly surpassing the benchmark average. This collective affirmation substantiates the notion that the developed system stands not just as an ordinary solution, but as a high-caliber creation that effectively meets user needs and merits their trust.

CONCLUSION

In conclusion, this research has navigated the intricate landscape of cybersecurity, focusing on the development and evaluation of an advanced phishing detection system tailored for online transactions. Through the integration of cutting-edge technologies such as JavaScript, Node.js, HTML, and CSS, the system's architecture was meticulously crafted to thwart the evolving tactics of malicious actors.

The culmination of this research journey underscores the system's resounding success in achieving its intended objectives. The data gathered from user experiences and perceptions, coupled with robust technical analyses, collectively highlight the system's exceptional performance in safeguarding against phishing threats. With a high satisfaction rate of 73.68%, reflective of users' endorsement of its efficacy, and an equally impressive responsiveness score of 84.21%, attesting to the system's seamless interactions, the research reaffirms the tangible impact of this innovative solution. Through a holistic lens, the average satisfaction rate of 77.89% emerges as a resounding testimony to the developed system's prowess, significantly surpassing the benchmark of user approval. Furthermore, the research's findings spotlight

the potential for widespread adoption, with an overwhelming 84.21% of respondents expressing their intent to recommend the system to others. This collective affirmation underscores the system's capacity to not only meet but exceed user expectations, forging a path toward bolstered cybersecurity within the realm of online transactions. In a digital landscape marred by persistent threats, this research stands as a beacon of innovation, charting a course toward safer online interactions and reinforcing the foundation of trust in the ever-evolving realm of technology.

FUTURE WORK

The future work involves various areas of improvement and research. Researchers can explore advanced machine learning techniques such as deep learning and ensemble methods to enhance the system's accuracy and adaptability to new phishing patterns. Behavioral analysis can be further developed to include biometric authentication and mouse dynamics to detect anomalies in user interactions. Real-time threat intelligence can be integrated to keep the system updated with the latest phishing URLs and patterns. Multi-platform support can be expanded to accommodate various web browsers and devices. User education features can be incorporated to raise awareness about phishing risks and promote safe online transactions. Phishing simulation and testing capabilities can be added to assess organizational susceptibility and improve user awareness. API integration can be provided to enable third-party applications to utilize the system's phishing detection capabilities. Cloud-based solutions can be developed for scalability and accessibility. Privacy and data protection measures can be enhanced for user trust. Feedback and user involvement can be encouraged for continuous improvement. Adaptive learning mechanisms can be implemented to evolve the system based on user feedback and emerging phishing trends. Collaboration with the cybersecurity community can provide valuable insights and knowledge-sharing opportunities. These future research directions aim to create a more robust, effective, and user-centric phishing detection system to safeguard online transactions from evolving threats.

REFERENCES

1. Jain, A. K., & Gupta, B. B. (2021). A survey of phishing attack techniques, defence mechanisms and open research challenges. Enterprise Information Systems, 16(4), pp 527–565. https://doi.org/10.1080/17517575.2021.1896786.
2. Dam Minh Linh , Ha Duy Hung, Han Minh Chau, Quang Sy Vu and Thanh-Nam Tran (2024). Real-time phishing detection using deep learning methods by extensions. *International Journal of Electrical and Computer Engineering.* Vol. 14, No. 3, pp. 3021-3035
3. Gandotra, E. and Gupta, D. (2021). An efficient Approach for phishing detection. Multimedial security, 15(12), pp 978-981.
4. Barraclough. P.A, Hossain. M.A, Tahir. M.A, Sexton. G, and Aslam. N, (2020). Intelligent phishing detection and protection scheme for online transactions. Expert systems with Applications,27(4), pp 4697-4706.
5. Choudhary, A., & Jain, A. (2017). Smishing Attack Detection and Prevention. International Journal of Computer Applications, 160(4), pp 32-37.
6. Mishra, A.K., Tripathy, A.K., Saraswathi, S., and Das, M. (2020). Prevention of phishing Attack in internet-of-things based cyber-physical Human system. Studies in computational intelligence, 17(5), pp 913-970.
7. Martin, D., Fabian, B., & Thomas, K. (2011). Phishing susceptibility: An investigation into the processing of a targeted spear phishing email. Journal of Computer Security, 19(5), pp 769-790.
8. GOV.UK (2020). Avoid and report internet scams and phishing. GOV.UK. https://www.gov.uk/report-suspicious-emails-websites-phishing
9. APWG, (2022): Phishing Reaches Record High; APWG Observes One Million Attacks Within the Quarter – For the First Time – in the First Quarter of 2022. https://apwg.org/apwg-1q-202
10. Desolda, G., Ferro, L. S., Marrella, A., Catarci, T., & Costabile, M. F. (2021). Human Factors in Phishing Attacks: A Systematic Literature Review. ACM Computing Surveys, 54(8), pp 1– 35. https://doi.org/10.1145/3469886.
11. Babu, S., Kumar, A., Dharavath, R., & Murthy, M. S. R. (2010). Enhanced Phishing Detection using PageRank Algorithm. International Journal of Computer Applications, 5(4), pp 30-34.
12. Aburrous, M., Almomani, I., & Sallabi, F. (2021). Machine learning techniques for phishing detection and protection: A survey. Computers & Security, 108(7), pp 102-238.
13. Khadir, A., & Sony, S. (2020). A review on phishing attacks and their detection techniques. International Journal of Advanced Computer Science and Applications, 11(2), 270-277.
14. Awasthi, A., & Goel, N. (2022). Phishing website prediction using base and ensemble classifier techniques with cross-validation. Cybersecurity, 5(1), pp 508-590 https://doi.org/10.1186/s42400-022-00126-9.
15. Wenyin, Z., Wugang, Z., Jianying, Z., & Yuan, Y. (2020). Visual similarity-based anti-phishing scheme. International Journal of Security and Its Applications, 14(1), pp 65-74.
16. Al-Ahmadi. S, Alotaibi. A and O. Alsaleh, 2022: "PDGAN: Phishing Detection with Generative Adversarial Networks," IEEE Access, vol. 10(4), pp. 42459-42468, 2022, doi: 10.1109/ACCESS.2022.3168235.
17. Kumar, A. (2019). Anti-phishing technique based on visual cryptography. In Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS). 7(2), pp. 99-104.
18. Chou, N., Jin, H., & Luo, W. (2020). SpoofGuard: A Browser Plug-In to Mitigate Spoofing Attacks. International Conference on Computing, Networking and Communications (ICNC). 18(3), pp. 73-77.
19. Fatt, G. E., Alqatawna, J., & Al-Bayatti, A. H. (2020). A new technique for detecting phishing websites based on favicon. Journal of Computational Science, 44(7), pp 101-178.
20. Dam Minh Linh , Ha Duy Hung, Han Minh Chau, Quang Sy Vu and Thanh-Nam Tran (2024). Real-time phishing detection using deep learning methods by extensions. *International Journal of Electrical and Computer Engineering.* Vol. 14, No. 3, pp. 3021-3035

21. Xiang, G., Hong, J., Rose, C. P., Cranor, L. F., & Hong, J. (2021). CANTINA: A Content-Based Approach to Detecting Phishing Web Sites. ACM Transactions on Internet Technology (TOIT), 20(4), pp 1-29.
22. Zhang, H., & Yuan, Y. (2021). Phishing Detection Based on Feedforward Neural Network. In International Conference on Wireless Communication and Artificial Intelligence. Springer, Cham. 16(5) (pp. 291-300).
23. Shekokar, N., Dey, S., & Kaur, I. (2022). Phishing Detection and Prevention through LinkGuard. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 8(1), pp 151-162.