Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Awareness model for minimizing the effects of social engineering attacks in web applications

Maher Al-Khateeb[a], Mohammad Rasmi Al-Mousa[a*], Ala'a Saeb Al-Sherideh[a], Dmaithan Almajali[b], Mahmoud Asassfeh[a] and Hayel Khafajeh[a]

[a]*Zarqa university, Jordan*
[b]*Applied Science Private University, Jordan*

C H R O N I C L E

Article history:
Received: October 20, 2022
Received in revised format: October 28, 2022
Accepted: January 30, 2023
Available online: January 30, 2023

Keywords:
Social Engineering
Cybercrime
Malicious code
Attacks
SE attacks

A B S T R A C T

Social Engineering (SE) Attacks against information systems continue to pose a potentially devastating impact. Security information systems are becoming increasingly significant as the number of SE incidents rapidly increased and became more aggressive than before. The World Wide Web (WWW) has evolved for information exchange and knowledge-sharing. It enables the sharing of information in a timely, effective, and transparent manner. Identity theft and identity misuse are two sides of cybercrime in which hackers and fraudulent users collect sensitive information from current legal users in order to perform fraud or deceit for financial gain. Malicious links are used as phishing methods, in which malicious links are planted beneath legitimate-looking links. As the number of web pages grows, the number of malicious web pages and the attacks of such become more complex. In this paper, we provide a method for identifying malicious web pages using a crawling and classification approach that helps to support the automatic discovery of the malicious links. The proposed approach can successfully complete the crawling session even if the page requires partial page refreshment and authentication credentials. The evaluation of the proposed approach shows a higher accuracy compared to an existing approach with an overall accuracy of 72% in three custom applications. Moreover, the proposed approach will calculate the significance and the impact severances of each link on the website and it better differentiates malicious web pages and normal links. The results of the proposed approach will also help in providing a set of recommendations which can increase the awareness level of the end-users, website administrators on how to better deal with these types of SE attacks.

1. Introduction

Before 1999, people were only readers of Web content and the Internet was regarded as a fancy tool that only professionals, computer-savvy users, and "nerds" could play with. During that time, the number of Internet users was less than 5% of the population globally; however, cyber-attacks have grown exponentially due to recent developments in the Web. These malicious URLs are widely used to send different types of attacks, with examples including malware, spam, and phishing and the detection and identification process of these threats are very hard (He et al., 2018; Wang et al., 2017; Al-Sherideh et al., 2018; Al-Sherideh et al., 2020). The emerging Web applications and content services with smart search facilitates the job of adversaries to use the Internet as a medium to spread malicious attacks, such as spamming, malware infection, and phishing (Al-Mousa, 2021). Phishing, for example, usually entails using a fake website that appears to be from a reliable source to trick people into clicking a connection that leads to a fake website. Many researchers in academia and industries have been working

* Corresponding author.
E-mail address: mmousa@zu.edu.jo (M.R. Al-Mousa)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2023.1.010

on malicious link detection to address this challenge, typically through inspecting links to detect their malicious content (Al-Mousa et al., 2020; Rasmi & Al-Qawasmi, 2016; Al-Mousa et al., 2022; Saxe et al., 2018).

Today, many malicious links are found on the Web. And because of the high rate of joblessness worldwide, job-seeking websites are becoming popular as they are convenient and economical (He et al., 2018; J. Liu et al., 2018). One can quickly get a space with a publisher such as Jobbeman.com and place adverts for job vacancies. On the other hand, many people rely heavily on those websites to have information about available job vacancies in order to apply. This information can help hackers and attackers find web links to be low-cost and highly effective in conducting malicious activities (Wang et al., 2019; Wang et al., 2017; Wang et al., 2017).

This paper will focus on detecting malicious links found on job-seeking websites so that a publisher or administrator of such websites would be able to identify and block those URLs containing such links before any job seeker visits them and gets trapped.

2. Related Work

Social Engineering (SE) refers not to strategies but to the art of using human psychology to gain access to private data. In order to let workers, know their password instead of trying to find a flaw inside a program or technology, they may, for instance, call an employee to act as an IT support person (Singh & De, 2017). Attacks can be detected but not avoided, as verified by the authors of (Saxe et al., 2018) SE attacks take advantage of sacrifices to gain sensitive information, which later can be used to unique results. With the widespread use of large-scale data that has drastically altered the way organizations, businesses, industry's function, assailants/adversaries can take advantage of them to capitalize on useful knowledge for business purposes (Paananen et al., 2020). With smart search web apps, opponents may use the Internet as a means to spread malicious attacks, including spamming, malware infection, and phishing. The most common attacks committed by SE are phishing attacks (Adi et al., 2017; Sahoo & Gupta, 2019; Silva et al., 2020) the goal is to acquire private and sensitive data through telephone calls or emails fraudulently from intended purposes. Victims are misled by attackers in order to gain private and sensitive information. False pages, emails, advertisements, virus-free applications, scareware, etc. However, technology cannot avoid such a form of attack, and a SE with no security expertise can easily trick even a robust security system. Fig. 1 Shows SE attacks classifications; they can be classified into different categories based on several perspectives.

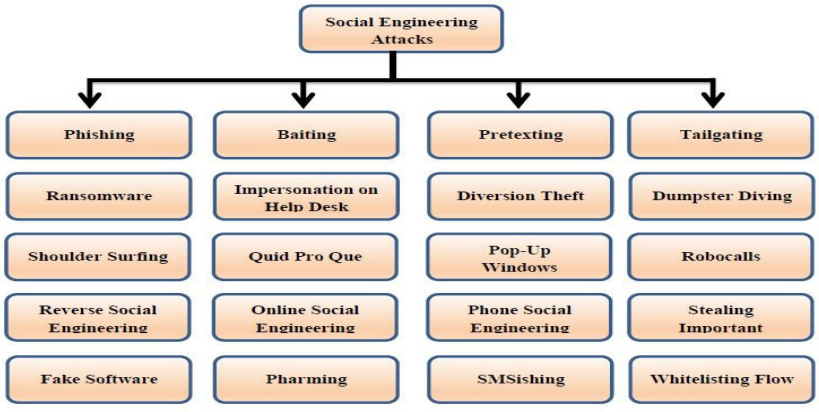

Fig. 1. SE Attacks Classifications.

As a result of the influx of information, crime has increased and offenders have made use of certain technologies and environments, both in space and in the real world. The most common form of cybercrime is Credit card fraud, with spoof websites and viruses' examples of internet crime. Hacking, network interference, and spreading malicious code that result in phishing crime is gaining momentum every day. Currently, millions of people use the Internet to connect to perform legal activities. Since web attacks are becoming more frequent, it's important to figure out what's causing them (Almasalha et al., 2018). In this regard, URL classification has piqued the interest of many privacy and security researchers. Phishing is a problem involving fake e-mails in which naive users are drawn in by malicious websites, which then gain access to the user's personal details. However, most websites today are not secure enough to prevent fishing activities that can quickly trick their visitors. Alqahtani and Alsulaiman (2020) highlight a framework that uses machine-learning to detect malicious URLs. They concentrate on optimising the use of manually defined features accuracy in detection (Mahdavinejad et al., 2018), which is also concerned with detecting malicious code. However, while the first of these is based on URLs, the second is not: the first is based on a manual feature engineering approach, while the second is based on a computer-aided feature engineering approach, which shows that learning features from raw data using a deep learning algorithm is possible. The efficiency of a neural network is improved. Hosseini et al. (2019) make use of URLs as a signal of identification, but also include other data to extract handcrafted data, such as URL referrers within web links features, which they provide as feedback to both SVM and

machine-learning and classifiers based on K-nearest neighbours. The studies of An et al. (2020) and Xu et al. (2020) provide examples of a body of work that attempts to manually removing features from malicious web content from HTML and JavaScript, and accordingly place them into one of two applications: the detection method, based on machine-learning or heuristics. Chen & Zhou (2020) suggest a method for extracting a broad range of features from a dataset of static HTML and JavaScript material on the website, and then feed machine-learning algorithms, which will use this data. They direct their efforts to include a number of different features and learning algorithms to compare and contrast their relative merits. Dai, Liu & Zhang (2020) are opposed to machine-learning and accordingly propose heuristics for detecting malicious code that are manually specified HTML, which is also built on static functionality.

Buldas et al. (2020) also make use of a heuristic-based system, but one that takes advantage of both to extract high-quality data, using a JavaScript emulator and an HTML parser characteristics Similarly, Singh et al.,2020 propose a web crawler with an integrated search engine embedded JavaScript de-obfuscation and de-compilation engine analysis with the goal of aiding in the detection of malicious content. A new approach to handling websites for phishing was proposed by Sadeghi et al.,2015 with the use of data mining and fluid logic to protect internet users when carrying out online transactions. In order to construct a model to predict websites using fuzzy data mining, it was employed to evaluate the likelihood of e-banking websites based on 27 features. In this model, there is an approach that has been multi-layered, with each layer including a collection of rules for categorising websites into five categories: very legitimate.

Another phishing detection method by Chin et al. (2017) uses a neural network, a technique of machine-learning. Since emails appear to come from a trusted source, many attacks by phishing users are unknown. In order to detect phishing attacks, a range of e-mails are analysed. The detection model integrates feed-forward neural networks through the collection and specification of a set of email-structure-related features and external connections (NNs). This model of NN has one layer of input and one layer of output. The technology extracts features to classify them during preprocessing before using a neural network. Bunder et al. (2018) suggest a mix of whitelist and visual approaches to countering phishing attacks. A SURF detector is a technique for computer vision extracting a main feature set from fraudulent and targeted websites. To start with, a whitelist of all valid URLs is created, which is then compared with the accessed URL. If they're the same, the website is thought secure; however, if the given URL is not included in the whitelist, SURF detection will kick in. The legitimate and suspicious pages are compared using a degree of similarity. If the semblance value is greater and the opposite way around the Web page is considered. Phishing and Social Engineering attacks (Yasasin et al., 2020) make users share private or delicate details by saying that they are legitimate web pages. Spam is the use of unwanted ads or phishing messages. Such attacks occur in large amounts and every year cause thousands of dollars of damage. Effective systems for timely detection of these malicious URLs can help to meet many and a variety of cyber-security threats. A content-based policy CANTINA is a program that detects phishing web links. Web spam is an issue associated with websites that have falsely generated pages. When a user clicks on a spam URL, it redirects them to dangerous pages that appear to be very real, but actually drives traffic to pages that are used for fun, benefit, or to mis-advertise people. Various methods are used in conjunction with classification algorithms to inspect certain spam pages or classify these malicious sites.

3. Proposed Awareness Model

Adversaries have used the Web as a vehicle to deliver malicious attacks such as phishing, spamming, and malware infection. For example, phishing typically involves sending an email seemingly from what appears to be a trustworthy source to trick people to click a URL (Uniform Resource Locator) contained in the email that links to a counterfeit webpage. This paper proposes a malicious links detector, which consists of crawling, detection, classification, and reporting. This component contains the basic building blocks for the proposed malicious connection detector. These components are viewed as modules, classes, objects, or a collection of similar functions during the implementation stage. The proposed malicious link detector architecture is shown in Fig. 2.

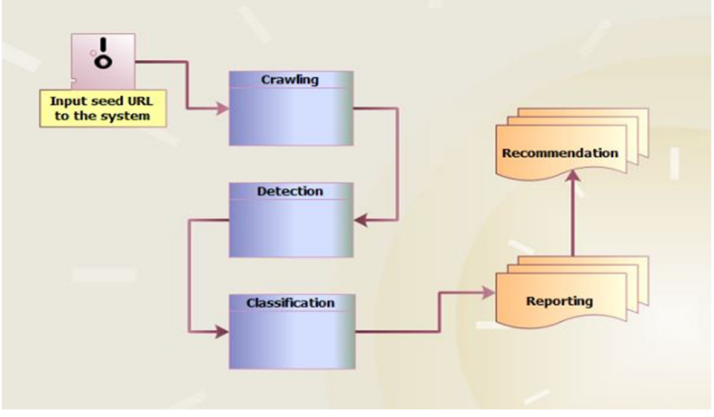

Fig. 2. The Architecture of Proposed Malicious link Detector

The components of the proposed malicious link detector can be explained as follows:

1. When a user provides a seed URL in the system, the second part, known as the crawler, takes the URL and gathers information about the target Web application's layout, as well as possible potential malicious relative links.
2. The third part, known as the detection test, is where the injection of all relative's links are gathered and stored for further assessment.
3. The fourth part is the classifier that consists of a pattern of malicious links, which is then used to compare a previously stored link in the third part. The link stored in the third part is considered malicious when it contains suspicious features identified in the classifier.
4. The last part is called reporting. In the reporting part, all potential malicious links detected are displayed.

Algorithm design is one of the basic elements of software design. It describes steps, processes, series, variables, and decisions that are required to bring built software into reality. The ability to analyse and visualise the solution in a more realistic manner is practical by having an algorithm developed. To make the proposed malicious link detector easier to implement, its components must be separated into a set of process and decision phases. The complexity of implementing the planned algorithm will be reduced as a result. The proposed malicious links detector is given in Algorithm 1.

Algorithm of Proposed Malicious Link Detector

1. Start
2. $crawling \leftarrow$ {seed url} // extract all the relative links available in a given website by using Jsoup
3. $if\ authentication\ is\ required$
4. $bypass \leftarrow \{attack\ codes\ to\ bypass\}$ //using brute-force method and a number of attacks is provided in Knowledge base dataset for login authentication bypassing
5. $else$
6. $extrac\ relative\ links \leftarrow \{seed\ ur\}$
7. $end\ if$
8. $store\ relative\ links \leftarrow \{Database\ A\}$ // store the extracted relative links from the given website
9. $store\ knowledge\ based \leftarrow \{Daabase\ B\}$ // Knowledge base dataset
10. For $A[0]\ to\ A[n]\ do$ // detection and classification process
11. Check for the pattern match in B [0] to B[n-1]
12. If match is found in B, then
13. Link is malicious. Link
14. $store$ malicious $links \leftarrow \{Daabase\ C\}$ // storage for the detected malicious links
15. Else
16. If match in B is null
17. Preform authentication bypassing
18. Repeat steps 6 -12
19. End if
20. For $C[0]\ to\ C[n]\ do$
21. Print the malicious links
22. End for
23. End for

The main goal from this paper is to improve the effectiveness of malicious links in web applications by proposing a malicious links detector to help and minimise the incidence of false positive and false negative results. Also, when authentication exists (a link in a given website needs user authentication), the malicious links detector employs a brute-force method in an attempt to discover the password of that link by systematically trying every possible combination of letters, numbers and symbols until discovering the password and to speed up the process. A number of common ways to bypass the login authentication are provided in the knowledge base dataset (i.e., dataset B).

The malicious links detector begins by crawling through the main home page of the website, it then extracts all the relative links available in that website by using Jsoup, which is a Java library that provides a powerful technique for data extraction and manipulation. The collected relative links are then stored in Dataset A. In the case that the website requires login authentication, the malicious links detector is provided with a number of attacks in Dataset B to bypass the login authentication. Dataset A serves as training set data. Dataset B refers to a test dataset that contains the list of unwanted keywords that can harm the website or its users. To identify whether or not the link is malicious, the malicious links detector begins by taking the links from Dataset A, one by one, and comparing them with the list of unwanted links in Dataset B (i.e., by searching for abnormal keywords, such as +1. 1+1, etc.). If any link from Dataset A contains similar keywords to Dataset B, that link is considered malicious and is stored in Database C. This process continues until all links in Dataset A are compared with unwanted keywords in Dataset B. A detailed description for datasets A and B, along with dataset C, is provided as follows:

Dataset A: Refers to the links the scanner has collected from the website requiring assessment. This includes all relatives' links that can take users from that home page to another page. These relatives' links can be linked to the original website or may redirect the user to another website.

Dataset B: This is the knowledge-based dataset that has been compared with what scanner has collected from the website. When the pattern from Dataset A is seen to match the pattern in Dataset B, that link is considered to be potentially malicious and is then passed to Dataset C.

Dataset C: This is the storage where previously identified malicious links are kept, which is then used to generate a report of malicious links detected by the administrator.

The malicious link detector uses a complex approach to understand how the pages in the web application are linked to one another. Knowing how pages are connected allows a malicious link detector to effectively identify every single page in an application. Many of the malicious connection detectors tested failed to detect malicious links on pages with a one-way relationship to their parent pages. Consider a web application with thirteen pages (13), as shown in Fig. 3 below, where the user page is only accessible after the login page has been accessed, and the profile page is only accessible after the user page has been accessed.

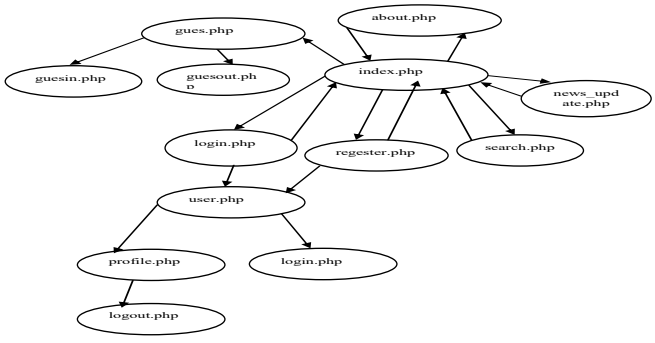

Fig. 3. Structure presentations of pages in news forum online application

The malicious link detector must examine these requests from a navigation standpoint in order to construct a graph of bidirectional communication through linked pages. Based on the navigation trace, the graph shows which page can be reached from which page. Following the malicious link detector's visit to the site, the next step is to extract relative links that are invoked by HTTP GET, POST, Request Cookies, and other types of variables that can be used to send relevant requests to the application. The Malicious Connection Detector was able to remove all of the application's relative connections. Many malicious link detectors struggle to recognise all existing links in the target application, especially when the page requires partial page refreshment of the authentication mechanism. To address this challenge, the proposed malicious link detector comes with a database of malicious link patterns that can be used to test potential malicious links and provide complete coverage of all links. Since there is no norm for defining type parameters, the proposed malicious connection detector must visit and define the right fields to field up with attacks.

3. Results and Analysis

The identification of attack types is useful since the knowledge of the nature of a potential threat allows us to properly respond, as well as to determine a pertinent and effective countermeasure against the threat, despite there being a variety of potential web applications designed to enable individuals or vendors to validate their tools against needed malicious links. Therefore, three web applications containing malicious links are used in this work. The reason for selecting these websites is down to the fact that the majority of the relevant malicious link detectors or scanners reviewed in the literature review used them to evaluate their proposed malicious links detectors in various scenarios. These websites include BuyandSell, Online Forum and JobSearch web applications.

The first Web application is the BuyandSell vulnerable target application, which contains five (5) documented malicious connections, three (3) of which are relative malicious links, and two (2) being additional malicious links that involve login authentication. The second Web application that is vulnerable is known as Online Forum, which is an online news application that contains four (4) documented malicious links.

The third vulnerable target application is Job Search, which contains thirteen malicious links that are directly connected from malicious sites designed to deceive people into entering an online gaming club. It contains two out of 13, which are underneath the login tab, with login authentication required to reach beyond those links. Two malicious links can only be accessed if the proposed detector is able to circumvent login authentication; otherwise, they will remain unreachable.

The first experiment was conducted on BuyandSell, as well as a malicious connection that was introduced by the original author of the malicious link detector. As can be seen below (see Fig. 4), the input URL of the BuyandSell application is given to the proposed malicious link detector, which crawls all pages, extracts, and indexes all links to known pages with malicious links (see Fig. 5).

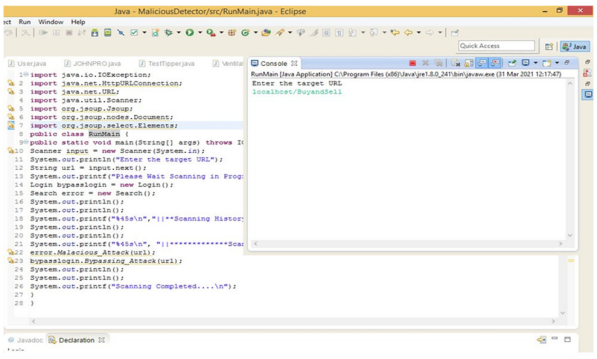

Fig. 4. Input URL of BuyandSell web application

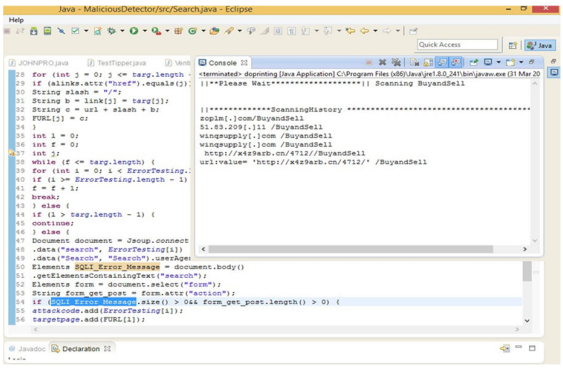

Fig. 5. Output scanning result of BuyandSell web application

Fig. 5 shows that the proposed system has successfully identified all malicious link pages with an additional false negative. This is because, in this scenario, the proposed malicious link detector is intelligent enough to bypass the login page successfully and reach underneath all pages available on the website. By using equation 1 and 2, the results are:

TPV = 4, and TKL = 5 Therefore, accuracy = 4/5 * (100%), = 80%
(TFPM = 1, and TKL = 5 Therefore, accuracy = 1/5 * (100%), = 20%

The second experiment is with the Online Forum Web application.

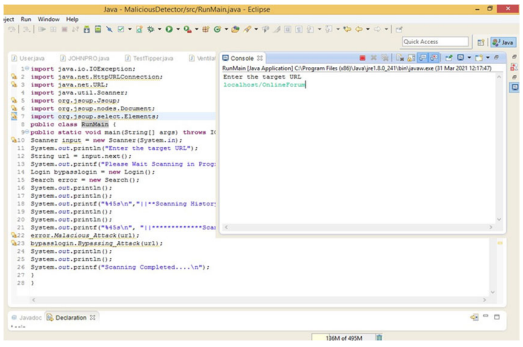

Fig. 6. Input of Online Forum application

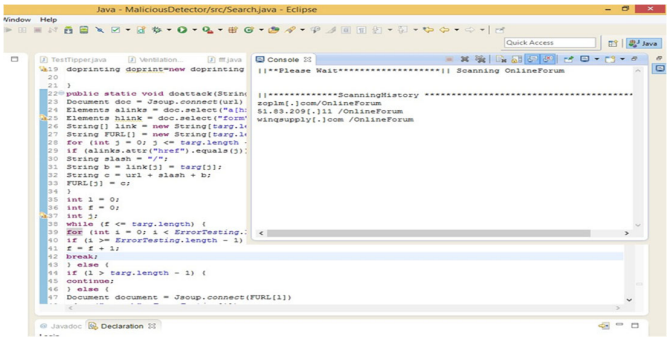

Fig. 7. Output scanning result of Online Forum application

As can be seen in Fig. 7, the proposed malicious links detector has the ability to detect 3 out of 4 malicious links, because one of these links is concealed under a legitimate link. The results, after applying Eq. (1) and Eq. (2), are as follows:

TPV = 3, and TKL = 4 Therefore, accuracy = 3/4 * (100%), = 75% (1)
TFPM = 1, and TKL = 4 Therefore, accuracy = 1/4 * (100%), = 25% (2)

The third experiment involved (13) malicious links on the JobSearch application.

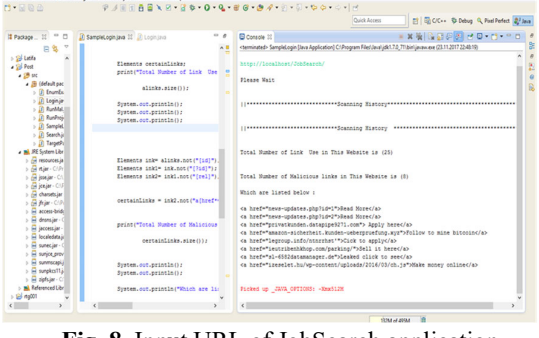

Fig. 8. Input URL of JobSearch application

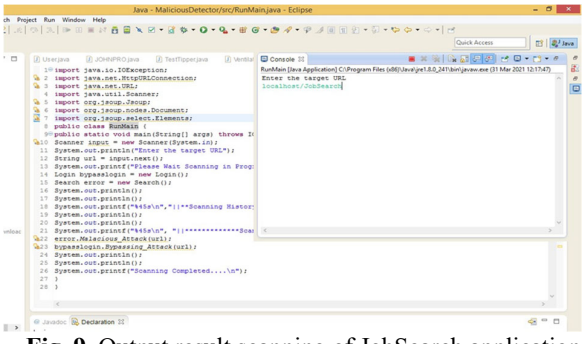

Fig. 9. Output result scanning of JobSearch application

As shown in Fig. 8, the input JobSearch application was given to the proposed malicious links detector, which successfully detected eight (8) malicious links out of 13, as shown in Fig. 9. The results of the accuracy are:

TPL = 5, and TKV = 6 Therefore, accuracy = 8/13 * (100%) = 61% (3)
TFPV = 1, and TKV) = 6 Therefore, accuracy = 5/13 * (100%) = 39% (4)

As can be seen in the first experiment, the proposed system achieved 80% accuracy and 20% false report on BuyandSell. In experiment 2, it achieved 75% accuracy and 25% false report on Online Forum and achieved 61% accuracy and 39% false report using JobSearch. Fig. 10 shows the results after applying the proposed malicious links detector on BuyandSell, Online Forum and JobSearch web applications.

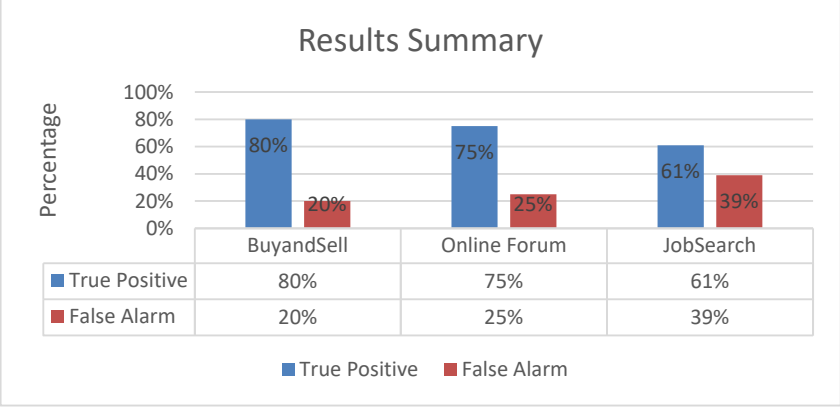

Fig. 10. Results summery for the experiments

The proposed system is found to be efficient in any web application that is not using modern partial page refreshment or does not require a user credential to crawl the pages. Even though the proposed system is equipped with a number of attack patterns required to bypass login applications, the proposed system faces challenges in bypassing such applications. The future improvement includes enhancing the attack pattern to successfully bypass all login applications in web applications, which will improve the chances of discovering potential malicious links.

As mentioned earlier, Viper and M.S Aliero et al. (2015) are only effective if the target application server requires an authentication mechanism. However, both Viper and M.S Aliero et al. (2015) achieved coverage of less than 80% (~78%) accuracy, while the proposed malicious links detector is within 80%. This is because these detectors were tested on applications that configure to log a user out when the page requires an authentication mechanism. Therefore, the aim from experiment 1 is to improve the crawling activities of Viper and Aliero et al. (2015), where the proposed malicious links detector reaches all links on tested applications. The result of this analysis shows the proposed malicious links detector detects them effectively. Moreover, a proposed malicious links detector achieved 61% coverage of true positive and 39% false positive accuracy compared with the work done by M.S Aliero et al. (2015), which achieved 52% coverage of true positive and 48% as false positive using JobSearch web application. Furthermore, the proposed malicious links detector is also compared in terms of crawling, partial page refreshment and authentication bypassing ability with some other available malicious detector tools in the literature as provided in Table 1.

Table 1
Comparison with an existing approach in the literature

Method	Crawling	Partial page refreshment	Authentication bypassing
(Aliero, Ghani, Qureshi & Rohani, 2019)	Yes	Yes	Partially
(Hosseini, Fakhar, Kiani & Eslami, 2019)	Yes	No	No
The Proposed approach	**Yes**	**Yes**	**Yes**

The two most related approaches are those of Aliero et al. (2019) and Hosseini et al. (2019), which share the same features of crawling web applications before discovering malicious links on the web application. The effectiveness of the approach can be determined by the ability to finish crawling activities. However, many modern applications use partial page refreshment or authentication requirements while crawling web applications. In this case, Hosseini et al. (2019) cannot achieve a high-performance application that requires partial page refreshment or authentication mechanism; Aliero et al. (2019), on the other hand, can complete the crawling session each time a partial page refreshment occurs, but cannot be stuck for login authentication in many scenarios. For example, if the login page is designed not to return the whole record in the database, the approach cannot bypass such a login authentication. This work, however, reshapes the attack patterns in various works, such as that of Aliero et al. (2019), for example, which is known to “select from an account where 1=1”, which cannot bypass the application

that is not returning the whole from the database. In our approach, we added “select from an account where 1=1 limit by 1” together with numbers of attack patterns which allows the proposed approach to bypass any login authentication encountered.

4. Conclusion and Future Work

This paper proposed a malicious links detector to conduct an efficient dynamic security assessment. The aim of this study was to reduce the amount of false negative and positive results from malicious connections by proposing a malicious links detector which can take into its consideration testing all the available crawling links in the web applications rather than selecting random crawling links from the website and testing them. In case these links are protected, the proposed malicious links detector is able to deal with them using authentication bypassing techniques. This study chooses to perform three separate experiments to replicate similar scenarios that researchers have performed, and accuracy measures were used to interpret the results in order to test the proposed malicious link detector in relation to the various literature studied. The experimental results showed that the proposed malicious links detector resulted in a significant improvement. The proposed system is proved to be efficient in any web application that does not use modern partial page refreshment, nor one requiring a user credential to crawl the pages. Even though the proposed system is equipped with a number of attack patterns that require bypassing login applications, the proposed system faced challenges in bypassing such applications. Future improvements include enhancing the attack pattern to successfully bypass all login applications in web applications, which will improve the chances of discovering potential malicious links.

Acknowledgment

This research is funded by the Deanship of Research and Graduate Studies at Zarqa University /Jordan.

References

Adi, E., Baig, Z., & Hingston, P. (2017). Stealthy Denial of Service (DoS) attack modelling and detection for HTTP/2 services. *Journal of Network and Computer Applications*, *91*, 1-13.

Aliero, M. S., & Ghani, I. (2015). A Component Tool IEEE Access.

Aliero, M. S., Ghani, I., Qureshi, K. N., & Rohani, M. F. a. (2019). An algorithm for detecting SQL injection vulnerability using black-box testing. *Journal of Ambient Intelligence and Humanized Computing, 11*(1), 249-266.

Almasalha, F., Naït-Abdesselam, F., Trajcevski, G., & Khokhar, A. (2018). Secure transmission of multimedia contents over low-power mobile devices. *Journal of Information Security and Applications, 40*, 183-192.

Al-Mousa, M. R. (2021). Analyzing Cyber-Attack Intention for Digital Forensics Using Case-Based Reasoning. arXiv preprint arXiv:2101.01395.

Al-Mousa, M. R. (2021, July). Generic Proactive IoT Cybercrime Evidence Analysis Model for Digital Forensics. In 2021 International Conference on Information Technology (ICIT) (654-659). IEEE.

Al-Mousa, M. R., Al Zaqebah, M ., Al-Sherideh, A. S., Al-Gghanim, Mohammed., Samara, G., Al-Matarneh, S., Asassfeh, M, R. (2022). Examining Digital Forensic Evidence for Android Applications. *In 2022 23nd International Arab Conference on Information Technology (ACIT).*

Al-Mousa, M. R., Sweerky, N. A., Samara, G., Alghanim, M., Hussein, A. S. I., & Qadoumi, B. (2021, December). General Countermeasures of Anti-Forensics Categories. In 2021 Global Congress on Electrical Engineering (GC-ElecEng) (pp. 5-10). IEEE.

Alqahtani, F. H., & Alsulaiman, F. A. (2020). Is image-based CAPTCHA secure against attacks based on machine learning? An experimental study. *Computers & Security*, *88*, 101635.

Al-Sherideh, A. S., Ismail, R. (2020). Motivating path between security and privacy factors on the actual use of mobile government applications in Jordan. *International Journal on Emerging Technologies, 11*(5), 558-566

Al-Sherideh, A. S., Ismail, R., Wahid, F. A., Fabil, N., & Ismail, W. (2018). Mobile government applications based on security and privacy: a literature review. International Journal of Engineering and Technology (UAE). https://oarep.usim.edu.my/jspui/handle/123456789/1614

An, C., Lim, H., Kim, D. W., Chang, J. H., Choi, Y. J., & Kim, S. W. (2020). Machine learning prediction for mortality of patients diagnosed with COVID-19: a nationwide Korean cohort study. *Sci Rep, 10*(1).

Buldas, A., Gadyatskaya, O., Lenin, A., Mauw, S., & Trujillo-Rasua, R. (2020). Attribute evaluation on attack trees with incomplete information. *Computers & Security*, *88*, 101630.

Bunder, M., Nitaj, A., Susilo, W., & Tonien, J. (2018). Cryptanalysis of RSA-type cryptosystems based on Lucas sequences, Gaussian integers and elliptic curves. *Journal of Information Security and Applications, 40*, 193-198.

Chen, Y., & Zhou, Y. (2020). Machine learning based decision making for time varying systems: Parameter estimation and performance optimization. *Knowledge-Based Systems*, *190*, 105479.

Chin, W.-L., Li, W., & Chen, H.-H. (2017). Energy Big Data Security Threats in IoT-Based Smart Grid Communications. *IEEE Communications Magazine, 55*(10), 70-75.

Dai, X., Liu, J., & Zhang, X. (2020). A review of studies applying machine learning models to predict occupancy and window-opening behaviours in smart buildings. *Energy and Buildings, 223*. doi:10.1016/j.enbuild.2020.110159.

He, X., Xu, L., & Cha, C. (2018). Malicious JavaScript Code Detection Based on Hybrid Analysis. *Paper presented at the 2018 25th Asia-Pacific Software Engineering Conference (APSEC).*

Hosseini, N., Fakhar, F., Kiani, B., & Eslami, S. (2019). Enhancing the security of patients' portals and websites by detecting malicious web crawlers using machine learning techniques. *International journal of medical informatics*, *132*, 103976.

Liu, J., Xu, M., Wang, X., Shen, S., & Li, M. (2018). A Markov Detection Tree-Based Centralized Scheme to Automatically Identify Malicious Webpages on Cloud Platforms. *IEEE Access, 6*, 74025-74038.

Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for internet of things data analysis: a survey. *Digital Communications and Networks, 4*(3), 161-175.

Paananen, H., Lapke, M., & Siponen, M. (2020). State of the art in information security policy development. *Computers & Security, 88*.

Rasmi, M., & Al-Qawasmi, K. E. (2016). Improving Analysis Phase in Network Forensics by Using Attack Intention Analysis. *International Journal of Security and Its Applications, 10*(5), 297-308.

Sadeghi, A.-R., Wachsmann, C., & Waidner, M. (2015). Security and privacy challenges in industrial internet of things. *Paper presented at the Proceedings of the 52nd Annual Design Automation Conference on - DAC '15.*

Sahoo, S. R., & Gupta, B. B. (2019). Classification of various attacks and their defence mechanism in online social networks: a survey. *Enterprise Information Systems, 13*(6), 832-864.

Saxe, J., Harang, R., Wild, C., & Sanders, H. (2018). A Deep Learning Approach to Fast, Format-Agnostic Detection of Malicious Web Content. *Paper presented at the 2018 IEEE Security and Privacy Workshops (SPW).*

Silva, C. M. R. d., Feitosa, E. L., & Garcia, V. C. (2020). Heuristic-based strategy for Phishing prediction: A survey of URL-based approach. *Computers & Security, 88*.

Singh, A., Parizi, R. M., Zhang, Q., Choo, K.-K. R., & Dehghantanha, A. (2020). Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Computers & Security, 88*.

Singh, K. J., & De, T. (2017). MLP-GA based algorithm to detect application layer DDoS attack. *Journal of Information Security and Applications, 36,* 145-153.

Wang, H.-h., Yu, L., Tian, S.-w., Peng, Y.-f., & Pei, X.-j. (2019). Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network. *Applied Intelligence, 49*(8), 3016-3026.

Wang, R., Zhu, Y., Tan, J., & Zhou, B. (2017). Detection of malicious web pages based on hybrid analysis. *Journal of Information Security and Applications, 35*, 68-74.

Wang, Z., Feng, X., Niu, Y., Zhang, C., & Su, J. (2017). TSMWD: A High-Speed Malicious Web Page Detection System Based on Two-Step Classifiers. *Paper presented at the 2017 International Conference on Networking and Network Applications (NaNA)*.

Xu, X., Zhao, Z., Xu, X., Yang, J., Chang, L., Yan, X., & Wang, G. (2020). Machine learning-based wear fault diagnosis for marine diesel engine by fusing multiple data-driven models. *Knowledge-Based Systems, 190.*

Yasasin, E., Prester, J., Wagner, G., & Schryen, G. (2020). Forecasting IT security vulnerabilities – An empirical analysis. *Computers & Security, 88.*