International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

Malicious link detector: a dynamic approach

*Muhammad Saidu Aliero[1], Bashar Umar Kangiwa[2] Abubakar Bashir[3], Muzzammil Mansur[3], Musa Ibrahim Kamba[3]

[1] *Department of ICT, Faculty of Engineering, Kebbi State University of Science and Technology Aliero, Nigeria*

[2]*Department of Computer Science, Faculty of Science, Kebbi State University of Science and Technology Aliero, Nigeria*

[3]*Deparment of Computer Science, Waziri Umaru Federal Polytechnic Birnin Kebbi, Nigeria*

Date of Submission: 15-10-2020 Date of Acceptance: 15-11-2020

ABSTRACT; Malware refers to the shell codes written by malicious users to redirected users of the trusted application to their malicious or proxy site with intent for criminal activities such as information or identity theft, downloading virus, or spy on users. Many researchers presented overview and solutions to address this challenge using static approach. However the practice is not feasible when source code of target applications is not available. Therefore this study proposed dynamic approach to addressing this challenge that enable remote users to scan target application for any malicious codes without the access to source code. The research focus improving the related approaches in order to help controlling of false positive an false negative result as well as to provide the room for improving proposed studies by the potential researchers. To test and validate the accuracy of research work, three vulnerable web applications were developed with different type of vulnerabilities and accuracy metric were used to analyze the result of three experiments. The result of analysis shows significant improvement by achieved 76% accuracy for the first experiment, 80 accuracy for the second experiment and 83% accuracy for the third experiment and 79.7% overall accuracy.

Keyword: Malicious link, web application, web application security, web application vulnerably dynamic approach

I. INTRODUCTION

Today malcious codes are written by malicious users with intent to redirected users of the trusted application to their malicious or proxy site with intent for criminal activities such as information or identity theft, downloading virus, or spy on users. Study in (MS Aliero, 2015) presented analytical evaluation on existing security assessments tools and approaches and conclude the 64% of web applications worldwide are at risk of being hack.

This is because, most of the developer and vendors for malicious code claim efficiency and effectiveness of their tool, and only few of them have the capability to test and evaluate the defect in target application efficiently on concrete scenarios. In fact, the effectiveness of true defects detected by these tools is within 5-45% and false positives produced by these tools are within the range of 10-30%.

Another reason is that, many of the automatic security assessment tools today do not detect existence of this malicious codes or links in trusted applications and report published by Software Administration, Network and Security and Institute of Computer Security/FBI, show almost 500 computer security analyst concludes that 55% of web penetration tester uses such tools for testing and evaluating effectiveness of their applications (Antunes & Vieira, 2010) (Nkima et al, 2016).

To address this challenge (MS Aliero, 2016) suggest that knowing the type of a threat enables estimation of severity of the attack and helps adopt a counter measure but this approach is costly and time consuming when web applications tends to grow. Therefore this study attempt on proposing feasible solution for automatic detection

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

of malicious links on web applications dynamically.

1.2 Problem statement

The dynamic interactions of online job seeking website present threat to the users as it became in house for scamming, fishing and fraud activities. With this regard many researchers proposed methods using a simple representation in terms of short path segments that describe the redirection relations among domains. Previous work (Sahoo, D., Liu, C., & Hoi, S. C. 2017), (M.S Aliero et al, 2015) , (Liban and Hilles, 2014) (Djuric, 2013), (Agosta et al., 2012) , (Li, Z., et.al, 2012), (Shakhatreh, 2010) ,, and (Ciampa et al., 2010) shows that many of the current scanners that uses crawler to detect malicious or vulnerable links in web applications were not able to complete their crawling session which result in false positive and false negative alarms in detecting malicious links.

II. PROPOSED SOLUTION

Design is the process of transforming all information gathered and structured in phase 1 into concrete idea about the new or replacement of new information system. It's not recommended to start coding a new system without having demented details on how system component are brought together, how different component of system interact, and classifying dependent and non-depended components. This section provides the details architecture, activities and algorithm design of proposed malicious link detector.

2.1 Architecture of Proposed Malicious Link Detector

Proposed malicious link detector architecture consists of four components; crawling, attacking classification and reporting. These components represent basic fundamental elements for structuring proposed malicious link detector. During implementation stage, these components are presented in form of modules, classes, objects or as a set of related function. The malicious link detector will consist of components that include, attack pattern, classifier, prediction component. Figure 1 show architecture of proposed malicious link detector.

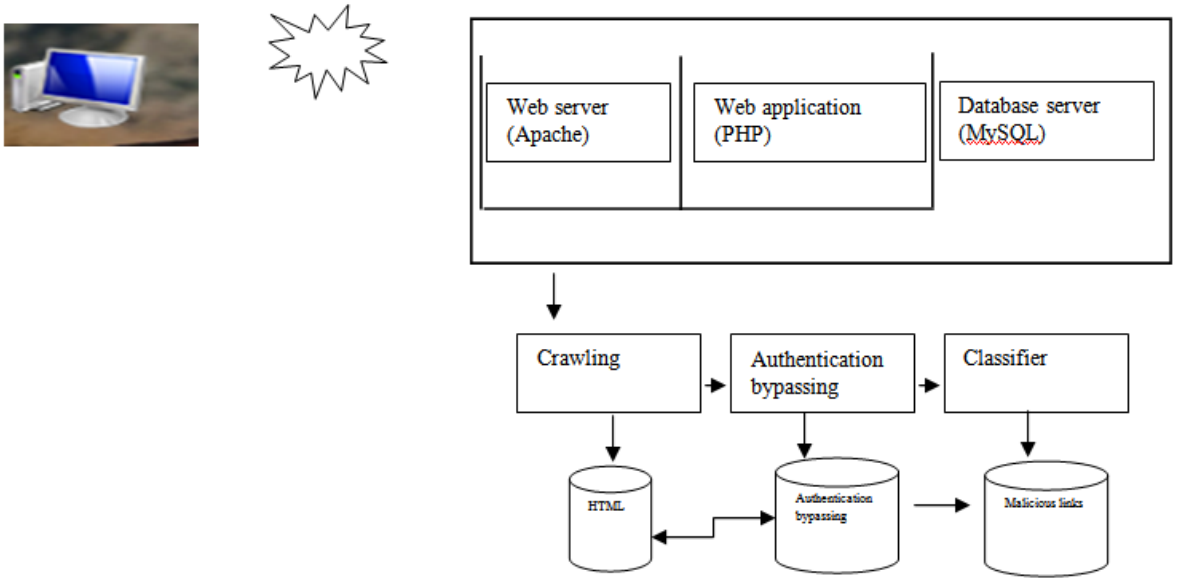


Fig 1 Architecture of proposed malicious link detector

The components of the proposed scanner can be explained as follows;

i. When user provide seed URL, the first component called crawler take seed URL and collects information about the structure of the target Web application and potential injection parameters.

ii. The second component called request test is where injection attack take place, different type of tautology SQL injection attacks in other to by bass log in authentication and go beyond to identify and index all the links available in target application.

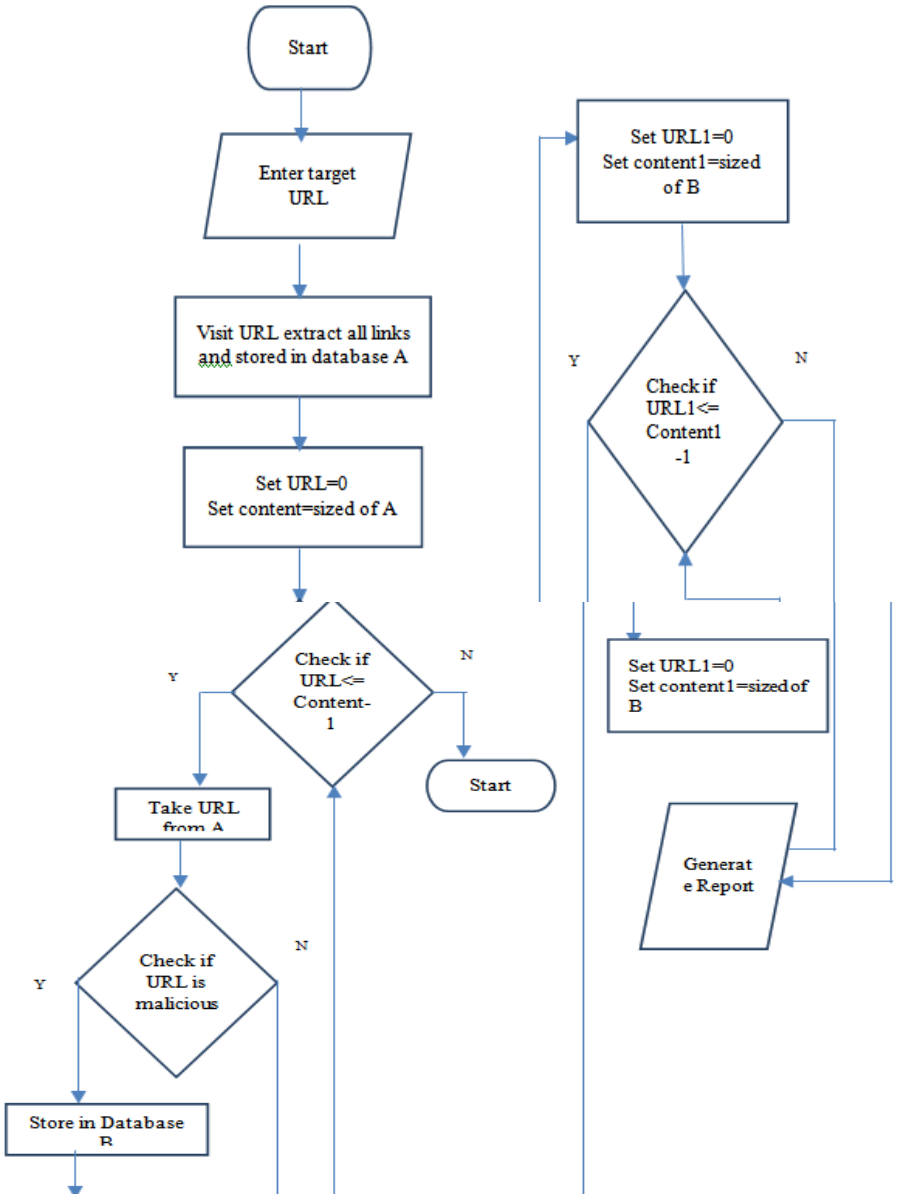
International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

iii. Classification, this component has knowledge to analyze links provided in database server to determine whether it is malicious or not.
It start by searching malicious links features in Table 1. Another feature to look for is authentication requirements for some pages that needs user credential before using the system. Such pages log out the proposed malicious link detector if care is not taken to bypass those authentications using SQLIA.
iv. The fourth component is reporting which display result to the users.

2.2 Algorithm of Proposed Malicious Link Detector

Algorithm design is the one of the fundamental elements in software design.It describes steps, procedure, sequence, variable and decision that is needed to bring designed software into reality. Having algorithm designed enables developer to examine and image the solution in more concrete manner.

To make proposed malicious link detector easier to implement, its component need to be divided into series of phases of processes and decisions. This will reduce the complexity of implementing designed algorithm. Fig 2 below represent the description of logical procedure of the proposedmalicious link detector undergoes to discover malicious link in a target application.



International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

Fig 2 Description of logical procedure of the proposed malicious link detector

III. EXPERIMENTAL AND RESULT DISCUSSION

Proposed Malicious link detector implement dynamic approach required tounderstand how pages in target application are connected to each other. Knowinghow pages are connected to each other would enable malicious link detector to identify everysingle page that exists in an application in effective way.

Many of the reviewed malicious link detectors failed to identify malicious link on page that has unidirectional relationship toits parent page. For example, consider web application with thirteen pages (13) as shown in fig 3 below, in which user page is only accessible after accessing login page and profile page is only accessible after accessing user page. Detector need to analyze these series of requests from navigation perspectives so that it can creates a graph that of bidirectional communication through linked pages. The graph indicates which page can be visited through which page based on the navigation trace.

What matters here is not how to navigate but how request will influence web application internal state machine.

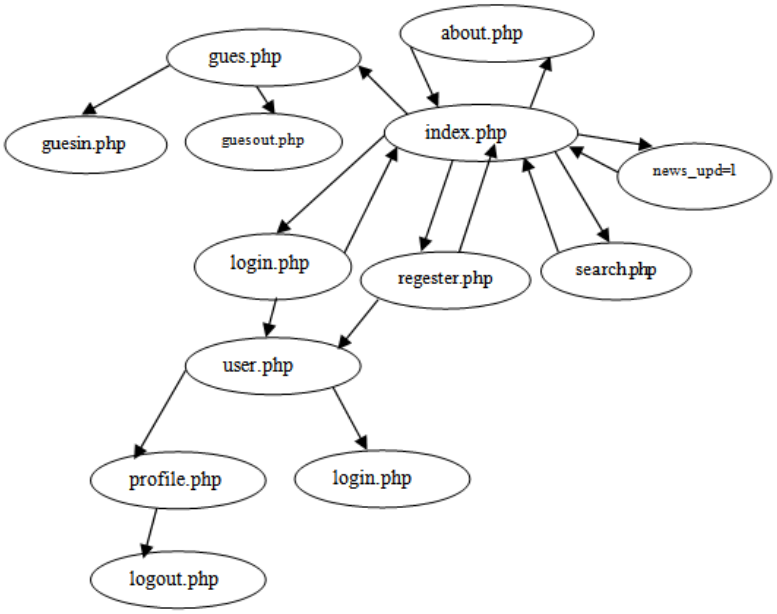


Fig 3**:** Graph showing site map of news forum online application

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

Table 1: Guess login form

Form action	POST		
Form type	Guesinuser.php		
Variable	name	Type	value
	Username	Text	--------
	password	passoword	--------
	submittoken	submit	Login

Table 2: Browser news update

Form action	REQUEST		
Form type	search.php id=echo$newsupadte		
Variable	name	type	value
	Null	Null	Null

3.1 Identifying Relative Links

The next step after visiting the site by Malacious link detector is to extract relative links which invoked by Http GET, POST, Request Cookies and other form variables that can be used in sending appropriate request to the application. Once Malicious link detector was able to extract all the relative links in that application. Many of the malacious link detectors faces challenge in entifying all existing links in target application especially if the page requires partial page refresment of authentication mechanism.

To adress this challenge, proposed malacious link detector is equipped with database of attack pattern use to bypass login authentication so as to achieve total coverage of every links. To achieve that, the proposedmalacious link detector needs to visit and ididentify right fields to field up with attacks because there is no standard for define forms parameters. Many developers defined the field in forms in the context of the purpose of their application. For example in table 1 authentication credentials fields in form are defined as “Username”, Password” “Submittoken” which cannot be the same as other forms in future. Some may decide to user “Email” instead of “Username”, “Login” instead of “submittoken”. To tackle this problem, information was gathered regarding most common names developers used to define in forms field and all those names have been simulated in proposed malicious link detector. Tables 1, 2 show injection parameters in some forms use in OnlineForumWeb application

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

Table Error! No text of specified style in document.**3:** Guess login form

Form action	POST		
Form type	Guesinuser.php		
Variable	name	Type	value
	Username	Text	--------
	password	passoword	--------
	submittoken	submit	Login

Table Error! No text of specified style in document..4: Browser news update

Form action	REQUEST		
Form type	search.php id=echo$newsupadte		
Variable	name	type	value
	Null	Null	Null

3.2 Vulnerable Target Applications Used

Despite there are number of potential web applications designed to allow individual or vendor to validate their tools against required Malicious links, we choose to designe three custom Web applications. One of the reasons is that most of the related malacious links detectorsor scanner studied in literature review are tested in different scenarios and different malacious link web application type. Therefore , the research chooses to develop these application to simulate these scenarios so that each malacious link detector or scanner can be evaluated based on its original vulnerable target application . Another reason is that, most of the individual or vendors adjust the effectiveness of their tool with respect to vulnerabilities in these applications which may not predict effectiveness of the tool in other application as different developer have different ways of writing same query(Antunes and Vieira, 2010), (Djuric, 2013).

The First vulnerable target application is JobSearch application consisting of six malicious links, there are linked directly from malicious sites developed to scam people about joining the club for online gaming. The application also has three more relative malicious links planted for the purpose of this research.

The second vulnerable target application is online BuyandSellapplication with five (5) known malicious links, three (3) relative malicious links and two other malicious links that requires login authentication.

The third vulnerable application is an online news application with six (6) known malicious links. The different between this application and other two vulnerable applications is that, in this,links application are designed in such a way that login authentication needs to be performedto achieve total indexing of all links in target application. This practice is mostly found in news websites. for example you may find online news website that provides description or headline of the news but the actual or remaining part of the news is stored in database.

All three vulnerable target applications are running on window 7 32 bit operating system and 6GB Ram, first and second application running on apache 2.4 with MySQL 5.5.19, and third application running on Apache 2.2.3 with MySQL 5.0.77.

3.3 Experiment 1

The first experiment was carried out on JobSearch which contain similar malicious links used in testing (Sahoo, D., Liu, C., & Hoi, S. C.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

2017), andthe other added malicious link that original author of the malicious link detector work on. Proposed malicious link detector was able to crawl all pages, extract and index all links to identified pages with malicious links, As can be seen below (see Figure 5.2), the input URL of JobSerach application is given to proposed malicious link detector and is displays the detecting result (See Fig 4.3).

Proposed malicious link detector identified five out of six malicious link pages in HR application. This is because proposed malicious link detector has quiet number of attack patterns required to bypass login authentication which cannot be done by the (Sahoo, D., Liu, C., & Hoi, S. C. 2017). However proposed malicious link detector failed to bypass "admin.php" page which is meant for authenticating administrator, this is because the query that is connecting Admin to database is developed with safety check (connect admin if and only if one record is return from the query). Proposed malicious link detector has successfully sent attacks that results in successful login but due to this safety check, proposed malicious link detector failed to connect to application (because the attack return all records from database). In this experiment the proposed malicious link detector achieved 79% accuracy metrics.

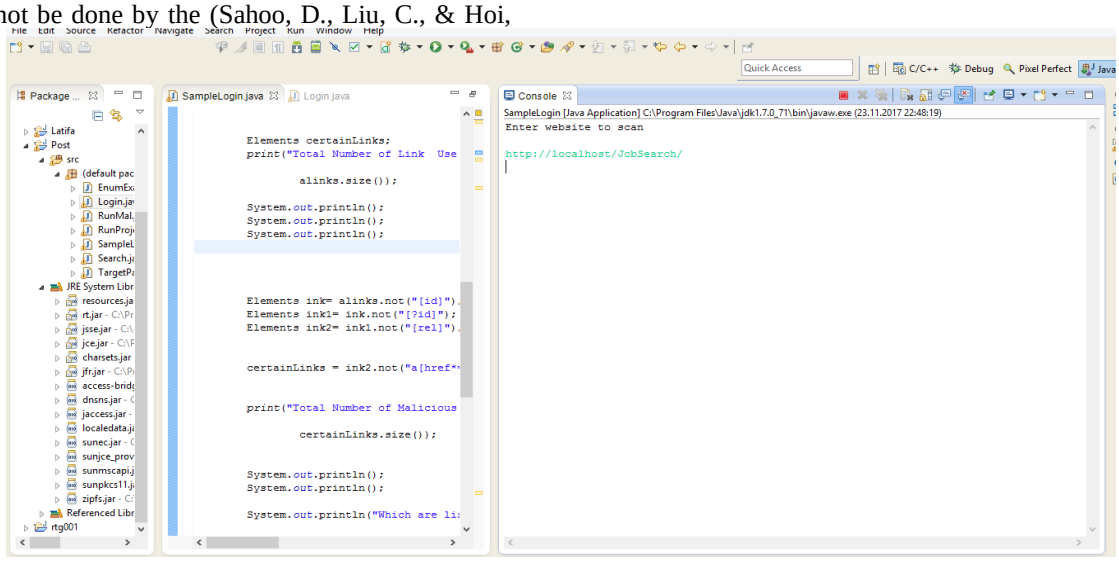


Fig 4.2: Input URL of JobSearch application

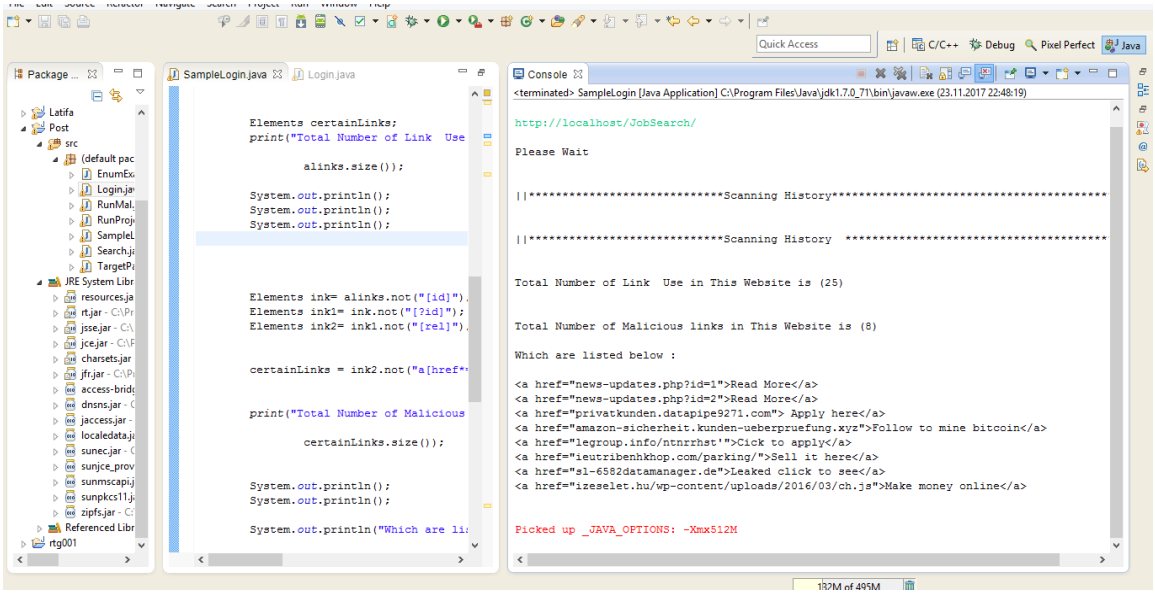


Fig 4.3: Output result scanning of JobSearch application

3.3.1 Result Analysis

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

This section present metrics use to evaluate the accuracy of proposed malicious link detector. This analysis is similar to the one used in evaluating (Sahoo, D., Liu, C., & Hoi, S. C. 2017).
Accuracy for experiment 1= TPL/TKV * 100, TFPV/TKN *100

Total number of true positive link (TPL) =5, Total number of known link (TKV) =6 Therefore, accuracy = 5/6 * (100%) , =79%
Total number of false positive link (TFPV) =1, Total number of known vulnerability (TKV) =6 Therefore, accuracy = 1/6 * (100%) , =21%

Proposed malicious link detector achieved 79% coverage of true positive and 21% false positive compared with (Sahoo, D., Liu, C., & Hoi, S. C. 2017) which achieved 72% coverage of true positive and 28% false positive. Similarly, proposed malicious link detector achieved less coverage of malicious links in(Sahoo, D., Liu, C., & Hoi, S. C. 2017) with total coverage of 76% and missing only 24% of malicious link on target applications.

3.4 Experiment 2

Second experiment was on BuyandSell application of five (5) malicious links two of the links are underneath of login page which requires login authentication to reach beyond those links. Two malicious links can only be reached when proposed detector can successfully bypass login authentication otherwise those links will remain unreached.

To bypass above mentioned page using proposed malicious link detector, it is required for the proposed malicious link detector to know at least one valid user that exists in database. In this case, the proposed malicious link detector uses admin as username which has high potential in most of the database. Proposed malicious link detector successfully bypasses this login application but still was not able to bypass other login (admin login similar to the one failed to bypass in JobSearch application). As can be seen below (See Figure 4.5), input BuyandSell application is given to proposed malicious link detector and successfully identified total number of four (4) malicious links out of five as displayed in Figure 4.6. In this experiment proposed malicious link detector achieved 80% accuracy when measure using accuracy metrics (see section 4.5.1 and in Figure 5.7)

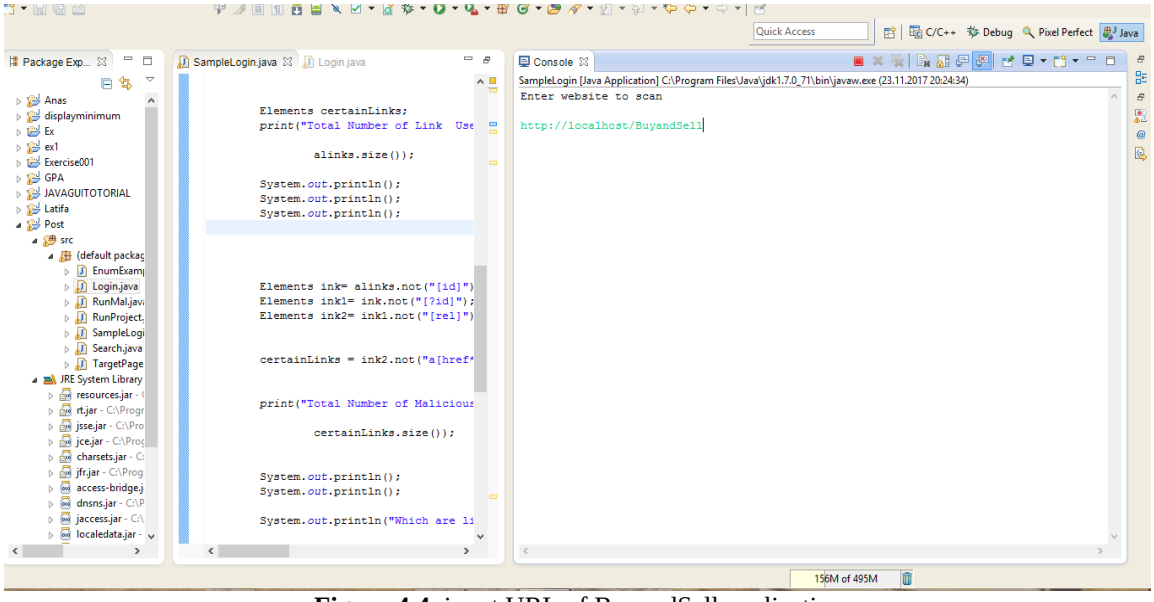


Figure 4.4: input URL of BuyandSell application

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

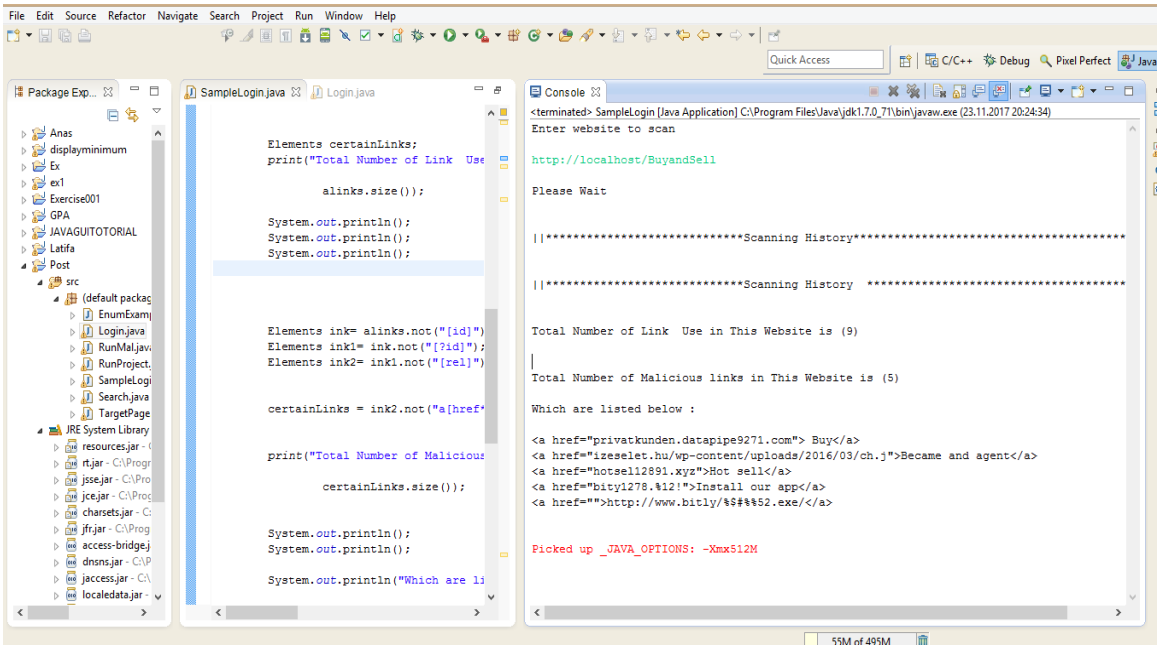


Figure 4.5: Output scanning result of BuandSell application

3.4.1 Result Analysis

This section present analysis result similar to the experimental setup used in Viper. Although Viper was previously compared with proposed malicious link detector (see section 4.4.1), however there is little difference between how these two applications (JobSearch and BuyandSell) are designed in their queries (see section 4.4).

As mention earlier that Viper and (M.S Aliero et al 2015) are only effective if target application server requires authentication mechanism. It's not surprise to see Viper and (M.S Aliero et al 2015) achieved coverage of less than 80% (~78%) accuracy while proposed malicious link detector with 80%. This is because these detectors ware tested on applications that is configure to log user out when page requires authentication mechanism. Therefore, the focus of this work in this experiment is to improve the crawling activities of Viper and (M.S Aliero et al 2015) reach all links on tested applications. Result of this analysis (see Fig 4.7) shows our malicious link detector.

Accuracy for experiment 2 = TPV/TKV * 100, TFPV/TKN *100

Total number of true positive malicious link (TPV) =4, Total number of known malicious link (TKL) = 5 Therefore, accuracy = 4/5 * (100%) , =80%

Total number of false positive malicious link (TFPM) =1, Total number of known malicious link detector (TKL) =5 Therefore, accuracy = 1/5 * (100%) , =20%

3.5 Experiment 3

The third experiment is on OnlineForum application which is similar to scenario proposed by (Djuric, 2013). Proposed malicious link detector successfully identified fivemalicious link application. As can ne seen URLs is given (see Fig 4.8) but failed to identify one malicious linkwhich is underneath of valid link (see Fig 4.9) and three(3) false negative alarm. This is because proposed malicious link detectordoes not have intelligence to classifylinks that has valid domain name i.e (".com", ."org" ".ng" e.t.c) as malicious links since most of the revised malicious links does not have valid domain in them. In this experiment, proposed malicious link detector achieved 83% (see Figure 4.10) accuracy when measured using accuracy metrics.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 8, pp: 492-503 www.ijaem.net ISSN: 2395-5252

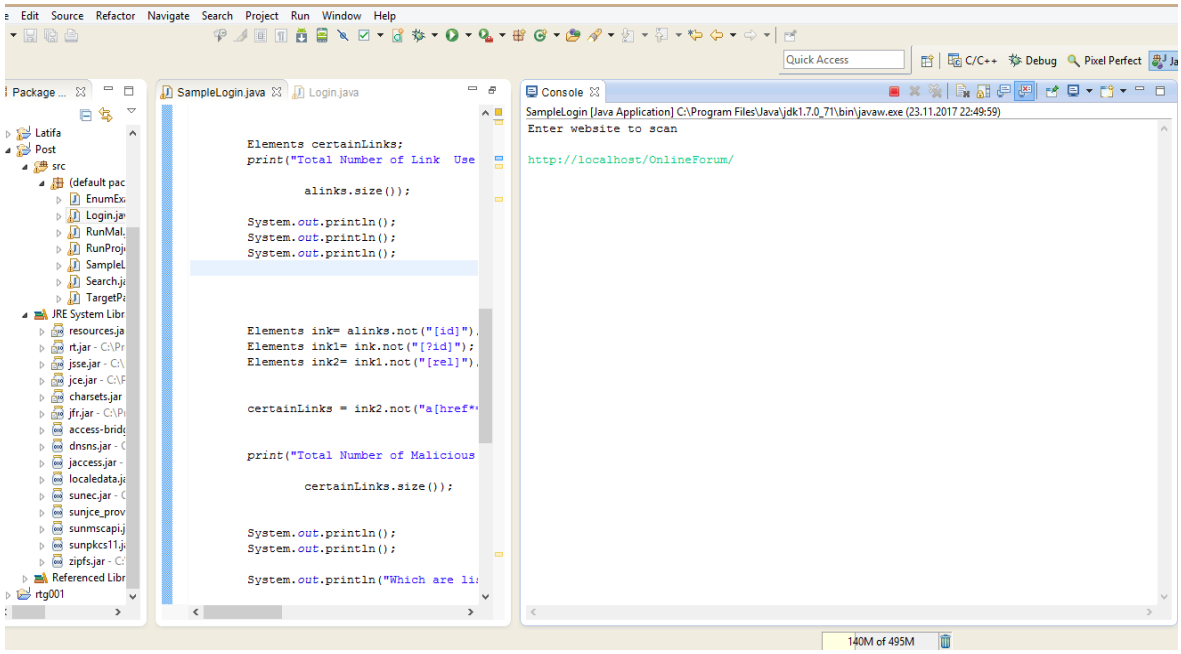


Fig 4.6 InputURL of OnlineForum application

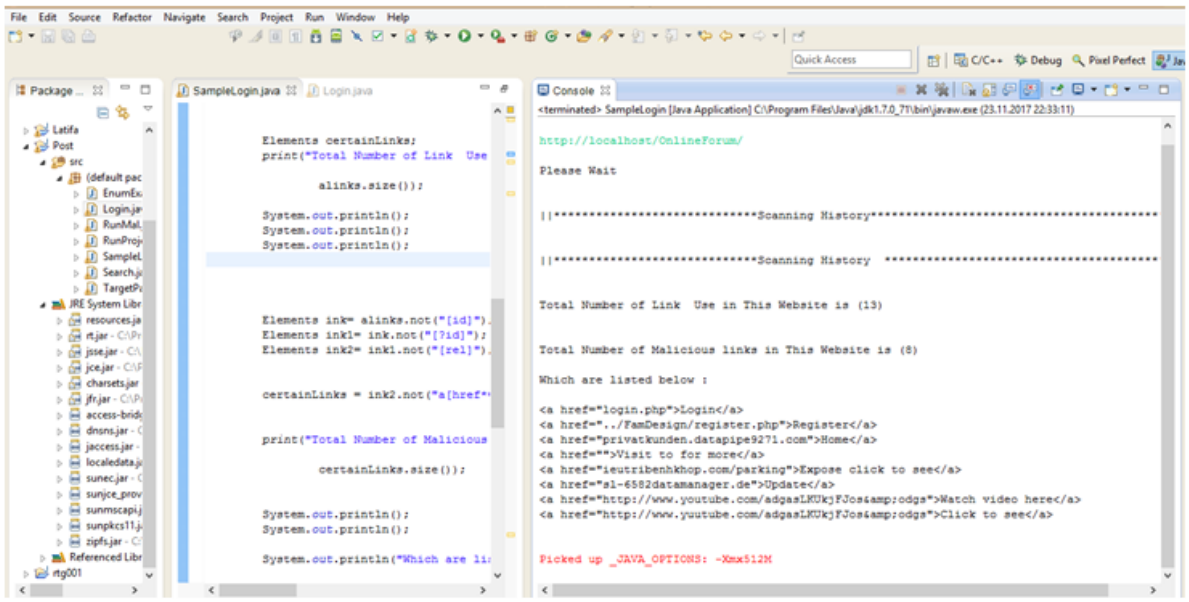


Fig4.7: Output scanning result of OnlineForum application

3.5.1 Result Analysis

This section present analysis result of proposed malicious link detector by (Djuric, 2013). This setup is different from other two setup (JobSearch and BuyandSell) as explained earlier (see section 4.3). Proposed malicious link detector uses three malicious link exposed to the agent. Therefore, the focus in this experiment is to improve accuracy of (Djuric, 2013).

Experimental result shows that proposed malicious link detector in (Djuric, 2013) achieved 64% accuracy while reporting 36% false positive. In this experiment our proposed malicious link detector show achievement of 62% accuracy and report with 25% false positive (see Fig4.10). Although in this research only PHP applications platforms ware considered unlike experiment in (Djuric, 2013) in which three different applications platform were used to validate malicious link detector.

Accuracy for experiment 3= TPV/TKL * 100, TFPV/TKN *100

Total number of true positive malicious links (TPV) =5, Total number of known malicious links (TKL) =5 Therefore, accuracy = 5/6 * (100%) , =83%

Total number of false positive malicious link detectors (TFPL) =1, Total number of known malicious links (TKL) =4 Therefore, accuracy = 1/6 * (100%) , =17%

IV. CONCLUSION

Adversaries have used the Web as a vehicle to deliver malicious attacks such as phishing, spamming, and malware infection. Automatic security assessment tools are used to automatically detect existence of defect, weakness or security flaws that can be exploited by potential attackers. These tools provide automatic way of security assessment either by examining the source code of applications or through penetration testing

Malware refers to the shell codes written by malicious users to redirected users of the trusted application to their malicious or proxy site with intent for criminal activities such as information or identity theft, downloading virus, or spy on users. this study proposed dynamic approach to addressing this challenge that enable remote users to scan target application for any malicious codes without the access to source. Our exprimental analsis shows promising results in detection of malacious links.

REFERENCES

[1]. Aliero, M. S., Ghani, I., Zainudden, S., Khan, M. M. & Bello, M. 2015. REVIEW ON SQL INJECTION PROTECTION METHODS AND TOOLS. Jurnal Teknologi, 77.

[2]. Indrani, B. & Ramaraj, E. 2011. X–LOG AUTHENTICATION TECHNIQUE TO PREVENT SQL INJECTION ATTACKS. International Journal of Information Technology and Knowledge Management, 4, 323-328.

[3]. Johari, R. & Sharma, P 2012. A survey on web application vulnerabilities (SQLIA, XSS) exploitation and security engine for SQL injection. Communication Systems and Network Technologies (CSNT), 2012 International Conference on, 2012. IEEE, 453-458.

[4]. Joshi, A. & Geetha, V 2014. SQL Injection detection using machine learning. Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on, 2014. IEEE, 1111-1115.

[5]. Kals, S., Kirda, E., Kruegel, C. & Jovanovic, N. Secubat 2006. a web vulnerability scanner. Proceedings of the 15th international conference on World Wide Web, 2006. ACM, 247-256.

[6]. Khoury, N., Zavarsky, P., Lindskog, D. & Ruhl, R 2011. An analysis of black-box web application security scanners against stored SQL injection. Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, 2011. IEEE, 1095-1101.

[7]. Kindy, D. A. & Pathan, A.-S. K. 2011. A survey on SQL injection: Vulnerabilities, attacks, and prevention techniques.

[8]. Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J. & Linkman, S. 2009. Systematic literature reviews in software engineering–a systematic literature review. Information and software technology, 51, 7-15.

[9]. Kumar, K., Jena, D. & Kumar, R 2013. A Novel Approach to detect SQL injection in web applications. June 2013.

[10]. Kumar, P. & Pateriya, R. 2013. Enhanced Intrusion Detection System for Input Validation Attacks in Web Application.

[11]. Lee, I., Jeong, S., Yeo, S. & Moon, J. 2012. A novel method for SQL injection attack detection based on removing SQL query attribute values. Mathematical and Computer Modelling, 55, 58-68.

[12]. Liban, A. & Hilles, S 2014. Enhancing Mysql Injector vulnerability checker tool (Mysql Injector) using inference binary search algorithm for blind timing-based attack. Control and System Graduate Research Colloquium (ICSGRC), 2014 IEEE 5th, 2014. IEEE, 47-52.

[13]. Liu, A., Yuan, Y., Wijesekera, D. & Stavrou 2009. A. SQLProb: a proxy-based architecture towards preventing SQL injection attacks. Proceedings of the 2009 ACM symposium on Applied Computing, 2009. ACM, 2054-2061.

[14]. Manoj, R. J., Chandrasekhar, A. & Praveena, M. A. An Approach to Detect and Prevent Tautology Type SQL Injection in Web Service Based on XSchema validation.

[15]. Masri, W. & Sleiman, S. 2015. SQLPIL: SQL injection prevention by input labeling. Security and Communication Networks, 8,

2545-2560.
[16]. Mishra, N. & Gond, S. 2013. Defenses To Protect Against SQL Injection Attacks. International Journal of Advanced Research in Computer and Communication Engineering, 2.
[17]. Narayanan, S. N., Pais, A. R. & Mohandas, R. 2011. Detection and Prevention of SQL Injection Attacks Using Semantic Equivalence. Computer Networks and Intelligent Computing. Springer.
[18]. Prabakar, M. A., Karthikeyan, M. & Marimuthu, K 2013. An efficient technique for preventing SQL injection attack using pattern matching algorithm. Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on, 2013. IEEE, 503-506.
[19]. Randive, P. U., Khatke, M. B. & Reddi, M. B. 2014. An Approach for Prevention of SQL Injection Attacks on Database: A Review.
[20]. Sadeghian, A., Zamani, M. & Manaf, A. A 2013. A taxonomy of SQL injection detection and prevention techniques. Informatics and Creative Multimedia (ICICM), 2013 International Conference on, 2013. IEEE, 53-56.
[21]. Shahriar, H. & Zulkernine, M 2012. Information-theoretic detection of sql injection attacks. High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on, 2012. IEEE, 40-47.
[22]. Shakhatreh, A. Y. I. 2010. SQL-Injection Vulnerability Scanner Using Automatic Creation of SQL-Injection Attacks (MySqlinjector). Universiti Utara Malaysia.
[23]. Shin, Y., Williams, L. & Xie, T 2006. Sqlunitgen: Sql injection testing using static and dynamic analysis. 17th IEEE International Conference on Software Reliability Engineering, ISSRE, 2006.
[24]. Singh, A. K. & Roy, S 2012. A network based vulnerability scanner for detecting SQLI attacks in web applications. Recent Advances in Information Technology (RAIT), 2012 1st International Conference on, 2012. IEEE, 585-590.
[25]. Son, S., McKinley, K. S. & Shmatikov, V. Diglossia 2013. detecting code injection attacks with precision and efficiency. Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013. ACM, 1181-1192.