International Journal of Information Technology (IJIT) – Volume 6 Issue 4, Jul-Aug 2020

RESEARCH ARTICLE OPEN ACCESS

Detection of Cross Site Scripting Attack using MONOSEK

Chaya P [1], Anjali N Menon [2], Madhurya Aithal A [3], Pratheeksha J [4], Varsha S [5]

Assistant Professor [1], Student [2], [3], [4], [5]

Department of Information Science and Engineering

GSSS Institute of Engineering and Technology for Women

Karnataka-India

ABSTRACT

The advent of network in all kinds of business technologies has made every individual more dependent on the internet for all the purposes. So are the threats for the same is increasing and the network security has become a major issue. Our project aims in detecting on the most popular attacks, the XSS attack in the websites using the monosek- a network processor-based network packet processing and network session analysis system. Also, the traffic generated in this attack produces packets which a recollected in the database and analyzed for further use.

Keywords :— XSS attacks, networks, cross-site scripting, packet analysis, monosek, network security

I. INTRODUCTION

Cyber-attack is a deliberate exploitation of computer systems, technology-dependent enterprises and networks. They use malicious code to alter computer code, logic or data, resulting in disruptive consequences that can compromise data and lead to cybercrimes, such as information and identity theft. Cross-site scripting (XSS) is a form of web security attack which involves the injection of malicious codes into web applications from untrusted sources. There are three actors in this attack (XSS) the attacker, the website and the victims can take use of JavaScript. XSS occurs even when the servers and the database engine contain no vulnerability themselves.

To perform XSS attack we use DVWA. It is a Damn Vulnerable Web Application coded in PHP/MYSQL. It is too vulnerable web application. In this application, security professionals, ethical hackers test their skills to practice some of the most common web vulnerability attacks like XSS attack, with various difficulties levels and run these tools in a legal environment. MONOSEK is intrusion detection software that monitors high speed network traffic by developing own traffic pattern with API calls. This software is embedded software for packet analysis, session analysis and deep packet inspection. MONOSEK plays a major role in order to analyse each packet that is transmitted in the network traffic and to detect the occurrence of XSS attack in the network using Deep Packet Inspection. XSS attack detection is the major aim of the project where we have an attacker system and victim system along with a MONOSEK server to monitor the packet transmission. As the attacker inject malicious script to web page and victim visit that page then attack occurs and victim data, cookies are stolen. In order to detect the attack occurrence, we use MONOSEK server which alerts the user as soon as the XSS attack occurs.

II. RELATED WORK

Various methodologies have been implemented till date on different platforms. Even though no IDS are 100 % secure. In this section we will take a look at previously proposed systems. [1]. DEXTERJS tool to detect and prevent the DOM-based XSS vulnerability on the web application by using the taint tracking mechanism. The tool is evaluated by the Alexa top 1000 sites which contain 820 distinct zero-day DOM-XSS. [2]. Nonce spaces tool to prevent XSS attacks by using the Instruction Set Randomization (ISR) techniques to differentiate between benign and malicious contents for thwart the XSS vulnerabilities exploitation. But Doesn't contain any defensive mechanism for inserted JavaScript code when downloading from the remote web site. Paper [3], uses the methodology Web Vulnerability Scanners (WVS) which has three major components: A crawling component, an attack component and an analysis component. It merging the mechanisms provided from XSS and SQL Injection. But the detection rate of certain type of XSS vulnerabilities is disappointing. In particular, scanners face problem in detecting stored XSS properly. In [4], A content security policy (CSP) can help Web application developers and server administrators better control website content and avoid vulnerabilities to cross-site scripting (XSS). Implementing CSP can help Web application developers specify allow-able content type and resource locations and can be an early warning system for any policy violations greatly assists system administrators' website control. The authors in [5] proposed a model combining techniques like Support Vector Machine classifier, fuzzy neural network and K-means. The input dataset is clustered using K- means algorithm in to k clusters, which are trained with the help of neuro fuzzy logic. Vectors are generated by passing each of the data generated through neuro fuzzy classifier and at last classification based on redial SVM (Support Vector Machine) is done to detect intrusion in the system. Paper [6] investigates using SVM, k-NN and Random Forests to detect and limit these attacks,

whether known or unknown, by building classifiers for JavaScript code. It demonstrated that using an interesting feature set combining language syntax and behavioural features results in classifiers that give high accuracy and precision on large real-world data sets without restricting attention only to obfuscation. [7] The system is black-box based approach which does not need to have a source code of a target application. The URL filtering took about 5 minutes. This result shows the effectiveness of the URL filtering. Among 245 XSS vulnerabilities, only 55 XSS vulnerabilities are unique. Hybrid analysis based XSS vulnerability detection approach, called HXD. It uses a heuristic to decide analysis approach tactic (static or dynamic) to accelerate XSS detection. [8] To identify any XSS or redirection vulnerabilities that could be initiated by using a maliciously crafted URL to introduce mischievous data into the DOM of inputted webpages (both statically and dynamically generated). It involves the dentition of more DOM-based features that could lead to detection of other code and server-side injection vulnerabilities like SQL and cross-site request forgery attacks. Paper [9] uses Rule-Based Detection Approach. The developed extension works accurately to stripping out the XSS queries however it is restricted to Google Chrome browser and JavaScript. Attacks on actuator signals are analysed from a system theoretic context. Paper [10] An automated framework to detect XSS attacks at the server side based on the notion of boundary injection and policy generation. It is proposed to detect the attack at the server side. The results indicate that the approach detects most of the well-known XSS attacks.

The work in this paper is divided in two stages. 1) XSS attack 2) Attack detection using MONOSEK. XSS attack is performed using DVWA application; it is most vulnerable web application where malicious script is injected. MONOSEK plays a major role in order to analyse each packet that is transmitted in the network traffic and to detect the occurrence of XSS attack in the network using Deep Packet Inspection. DPI rely on comparing to parts payload and signature (IP header). It compares them with known signatures to decide if the packet is harmful (similar to any of attacks database signatures).and deletes it or pass it through the network flow.

III. METHODOLOGY

The proposed system uses Monosek which is intrusion detection software that monitors high speed network traffic by developing own traffic pattern with API calls. Using signature comparison program, we inspect the payloads to detect the attack. It detects all types of XSS attack.

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

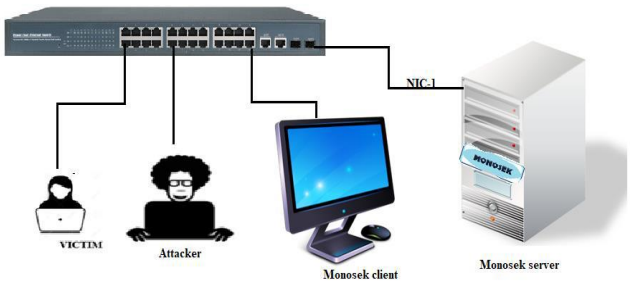

Fig 3.1: System architecture for XSS attack detection

System Modules:

1) Attacker: The attacker is the one who will insert the malicious unfiltered code to the server to get the required information for him. Attacker inserts the malicious code to the web page where the victim visits. Whenever the victim visits the web page he will be under attacked by the attacker and will get the information which is needed. And also, attacker get the control over the user data or system via injected exploit.

2) Victim: The victim module is the one where he will be affected by the attacker once he gets into the malicious page and the malicious data is sent to get required information. Once this has been done by the attacker, the victim will be in the control of the attacker.

3) Server: This is the module where the unfiltered code is stored and sent to victim unknowingly. This is where the packets are generated and processing is done. When attack occurs, each packet is generated and details like the IP address, names of protocols will be generated and filtered.

4) Monosek client: This is where all the generated packets are present. The GUI created will be linked to the packets which will display the same. The packets will be filtered and only the required ones will be displayed as per the requirement in the GUI.

The attacker logins into the web hosting site and checks whether the victim has clicked on the link where malicious script is being added by the attacker. If this particular link has been clicked by the victim, then the code is being executed where the attacker gets the cookies of the victims account and that will get appended to the blank cookie file i.e cookie.txt file. So now the attack has been occurred and this will be

detected using monosek which is intrusion detection software that can detect all kinds of attack such as XSS attack which is of two - stored, reflected. Monosek detects the packets that's been detected as the malicious and then displays it accordingly. The data diagram of attack is showed below in Figure 3.2 that describes how the victim's cookies are stolen from victim.

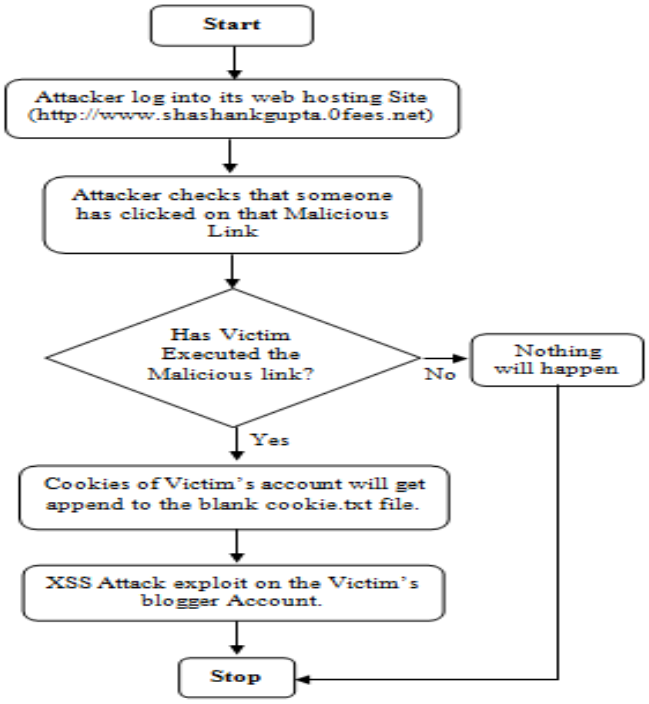

Figure 3.2 Data Diagram

The data diagram 3.2 explains the overall flow of the project working. The attacker logs into the host site and adds the malicious code into it. If victim clicks on the link with the malicious code, his site will be hacked and the cookies will be appended in the attacker's cookie.txt file. If not, then nothing will happen and the cookie.txt file will remain empty.

Sequence Diagram:

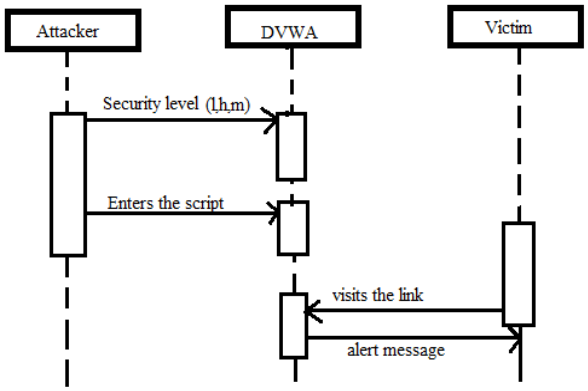

Figure 3.3 Sequence diagram with respect to attacker

The sequence diagram 3.3 references the attacker side. Here the attacker first sets the security level in the DVWA tool before scripting his malicious code. Once the security level is set, the attacker sends the script to the host site. The victim clicks on the site where the victim's IP address will be linked to the DVWA. The alert message will be sent to the victim saying "The Attack has been attempted".

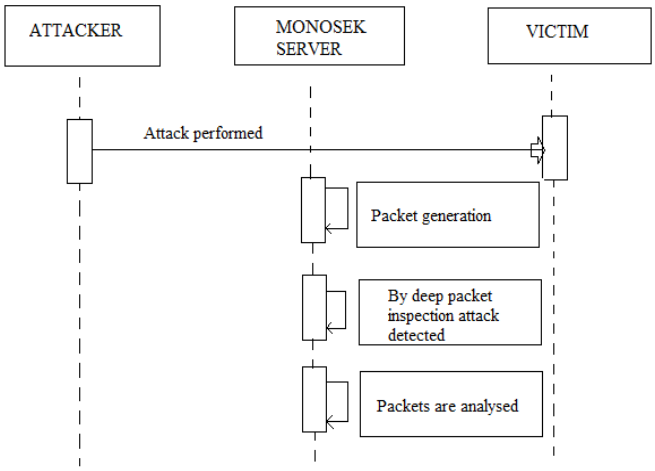

Figure 3.4 Sequence diagram with respect to Monosek server

The diagram 3.4 depicts the Monosek server. When the attack performed by the attacker, the packets will be generated using the deep packet inspection and will detect it. Using this detection, the monosek server will analyses the packets which is generated during this attack.

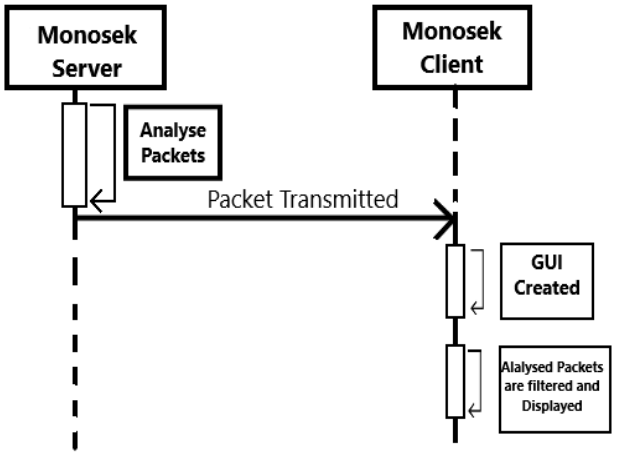

Figure 3.5 Sequence diagram with respect to Monosek client

The diagram 3.5 depicts the Monosek client. Once the monosek server analyse the packets, it will transmit all the packets to the Monosek client. The monosek client will display all these packets in the form of GUI in which packets are analysed, filtered and displayed.

IV. EXPERIMENTAL RESULTS

Figure 4.1 is describing how the malicious code can be inserted XSS attack using the scripting language.

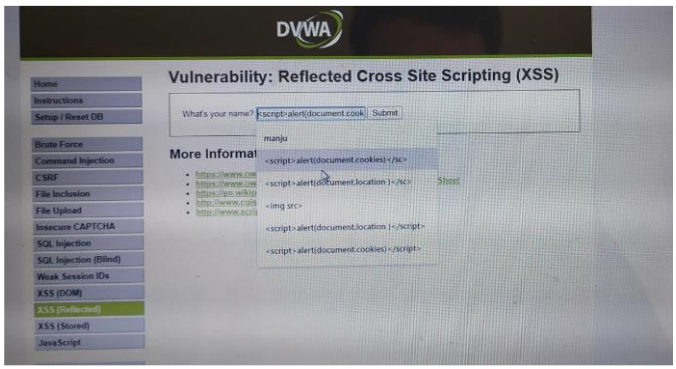

Figure 4.1: Depicts the scripting code which is being inserted

In figure 4.2 we have entered the scripting code and we are submitting to know that the site has been attacked by the attacker.

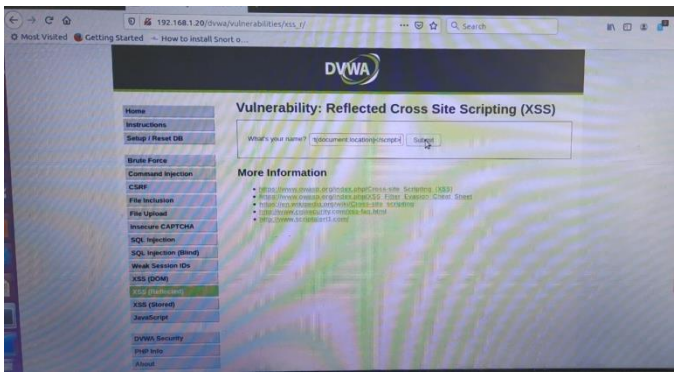

Figure 4.2: The correct script is chosen and submitted.

Figure 4.3 depicts the attempt of the XSS attack and the exact location will be detected of where the attack has happened and will record it.

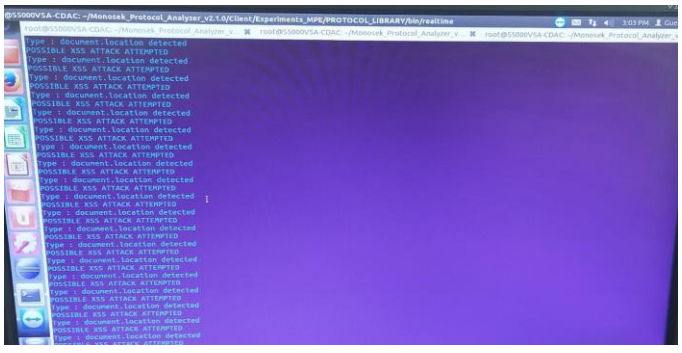

Figure 4.3: The attempt of attack and the location detected is recorded

V. ADVANTAGES

- Provided effective protection for the client against XSS attack.
- Help us to understand what sort of traffic is going over the network.
- This helps to researcher to create more effective detection and prevention system

VI. APPLICATIONS

- Used in companies for maintaining secured data.
- In military and defence areas where important information are stored.
- In educational institutions where personal data are stored.

VII. CONCLUSIONS

The XSS attacks are still exploiting the web application vulnerabilities to steal the user credential. The techniques that are used to detect and prevent the XSS attack still needs more work to enhance the accuracy of XSS detection and prevention. The architecture proposed and developed during this research work is effective in terms of providing analysis of network data to provide evidence of suspicious traffic. The future work is to develop a defensive mechanism that uses data mining and machine learning techniques, to detect and prevent the Stored XSS attack and DOM based XSS attack in order to reduce the false negative and false positive.

REFERENCES

[1] Inian Parameshwaran , Enrico Budianto , Shweta Shinde ,Hung Dang, Atul Sadhu, Prateek Saxena "DEXTERJS: Robust Testing Platform for DOM-based XSS Vulnerabilities" 10th Joint Meeting on Foundations of Software Engineering(August 30-September 4), pp. 946-949 Bergamo, Italy, 2015.

[2] Matthew Van Gundy and Hao Chen "Noncespaces: Using randomization to defeat cross-site scripting attacks" Computers & Security, No. 31, pp. 612 – 628, Elsevier, 2012.

[3] Punam Thopate, Purva Bamm, Apeksha Kamble, Snehal Kunjir, Prof S.M.Chawre "Cross Site Scripting Attack Detection & Prevention System" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)Volume 3 Issue11, November 2014.

[4] T. Scholte, D. Balzarotti, and E. Kirda, "Mitigating Cross-Site Scripting Attacks with a Content Security Policy", International Islamic University Malaysia, 2016

[5] A. M. Chandrasekhar, K. Raghuveer "Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM classifiers", 2013 International Conference on Computer and Informatics (ICCCI), Coimbatore, INDIA, Jan04-06,2013.
[6] Fawaz A.Mereani, Jacob m.Howe, University of london, "Detecting cross site scripting attacks using Machine learning" University of london,UK,2018.
[7] Hyunsang Choi, Seongjin Hong, Sanghyun Cho, Young-Gab Kim, "Hybrid XSS Detection by using a Headless Browser", 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), 2017.
[8] Bakare K. Ayeni, Junaidu B. Sahalu, and Kolawole R. Adeyanju, "Detecting Cross-Site Scripting in Web Applications Using Fuzzy Inference System", 2018.
[9] Divya Rishi Sahu, Deepak Singh Tomar, "Robust Defense against XSS through Context Free Grammar", 2015.
[10] Hossain Shahriar and Mohammad Zulkernine, "A Server Side Approach to Automatically Detect XSS Attacks",Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing,2011.
[11] V. K. Malviya,S. Saurav, "On Security Issues in Web Applications through Cross Site Scripting (XSS)",20th Asia-Pacific Software Engineering Conference,2013
[12] M.I.P. Salas and E. Martins, "Security Testing Methodology for Vulnerabilities Detection of XSS in Web Services and WS-security," Electron. Notes Theor. Comput. Sci. Elsevier, vol. 302, pp. 133– 154, 2014.
[13] Piyushkumar A. Sonewar, Nalini A. Mhetre, "A Survey of Intrusion Detection System for Web Application", International Journal of Engineering Research and Technology Vol. 1 (02), ISSN 2278 – 0181, 2014.
[14] S. Gupta and B. B. Gupta, "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of- theart," Int. J. Syst. Assur. Eng. Manag. Springer, 2015.
[15] A. Kiezun,M. D. Ernst, "Automatic Creation of SQL Injection and Cross-Site Scripting Attacks",ICSE, May 16- 24, 2009.
[16] Y. Minamide, "Static Approximation of Dynamically Generated Web Pages," in WWW '05 Proceedings of the 14th International conference on World Wide, New York, NY, USA, 2005.
[17] A. Kiezun, M. D. Ernst, "Automatic Creation of SQL Injection and Cross-Site Scripting Attacks", ICSE, May 1624, 2009.
[18] Dukes, L.; Xiaohong Yuan; Akowuah, F., "A case study on web application security testing with tools and manual testing," Southeastcon, 2013 Proceedings of IEEE, vol., no., pp.1,6, 4-7 April 2013.
[19] W. Alcorn, "Cross-site Scripting Viruses and Worms - A New Attack Vector," Netw. Secur. Elsevier, vol. 2006, no. 7, pp. 7–8, 2006.
[20] Punam Thopate, Purva Bamm, Apeksha Kamble, Snehal Kunjir, Prof S.M.Chawre, "Cross Site Scripting Attack Detection & Prevention System", International Journal ofAdvanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue11, November 2014.