# *CROSS SITE SCRIPTING (XSS) ATTACK DETECTION USING INTRUSTION DETECTION SYSTEM*

Kunal Gupta [1] Rajni Ranjan Singh [2] Manish Dixit [3]
Department of CSE & IT [1,2,3]
M.I.T.S. Gwalior, M.P. (India)
kunalgupta007@gmail.com [1] , rrsingh@mitsgwalior.in [2] , dixitmits@gmail.com [3]

***ABSTRACT* –Everyone is now relying on the Internet for our innumerable kind of work; this has increased the opportunity for attackers to corrupt data and make vulnerable. Nowadays diverse kind of attacks is being launched in Cyber Space among which Cross-Site Scripting (Web Application Attack) is amongst top attacks of all time. Proposed work, suggest an outline for a system that can detect Cross-Site Scripting (known as XSS) attack using Intrusion Detection system (IDS). This work focuses on the detection of XSS attack using intrusion detection system. Here attack signature is utilized to detect XSS attack. To test the usefulness and effectiveness of proposed work a proof of concept prototype has been implemented using SNORT IDS. It is observed that proposed system correctly detected XSS attack.**

***Keyword -- Cyber Space, Intrusion Detection system, Web Application, Cross-Site Scripting***

## I. INTRODUCTION

Our life has become dependent on the Internet, due to its vast scope and its useful functionalities that can solve our daily problems in very few time. However, due to this, we have become prone to attacks that can begin to massive loss of personal or organization level as we save our personal info including financial info on the net, which charms an attacker. Among various kind of attack most common web attack is web application attack which includes Cross–Site Scripting Attack and SQL Injection Attack. In this research paper, we are doing our research on Cross Site Attacks.

Cross Site Attacks are most common Network attacks across the web where we inject Payload (Malicious Code) on the client side to a website. It's a weakness found on poorly coded website which attacker exploits and tries. It uses victim's browser to deliver malicious script from a vulnerable site as vehicle. There are three actors in this attack (XSS) the attacker, the website and the victims can take use of JavaScript, Flash, VBScript and ActiveX. But mostly used is JavaScript.

Cross Site Script execution processes showed in Fig. 1:

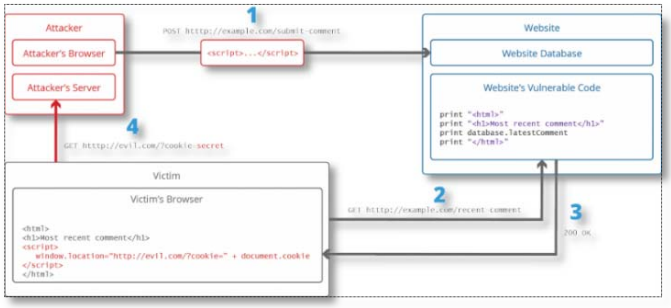


Fig. 1. – Cross Site Scripting Procedure [10]

1. Website database is infected through a payload by an attacker after submitting form with JavaScript Code.

2. Web page from the website is requested by a victim browser.

3. Victim's Browser is served the web page that was earlier requested by the victim with the payload attached to HTML body.

4. The payload will be executed inside the Victim's browser. Now attacker server will receive the victim's cookie. Victims cookie is extracted by the attacker. He uses victim's cookie to make the HTTP request to the server. After which attacker is granted request by the name of victim.

*CATEGORIES OF CROSS - SITE SCRIPTING ATTACKS*

1. *Stored XSS*: It take place when a target server stores the input from the user in the form of a message a database or visited log after this data becomes the part of website but instead of data user inputs a payload. When the stored payload is run locally, after which enables the malicious code that is saved as an data input by the user.

2. *Reflected XSS*: When a web application returns an error message, or any other response that comprises all input provided by the user immediately after user input was made.

*INTRUSION DETECTION SYSTEM (IDS)*

Intrusion Detection System inspects packets going to and from the network and makes a log for packet that are involved in attack or are vulnerable according to rules.

Types of IDS on the basis of Architecture:

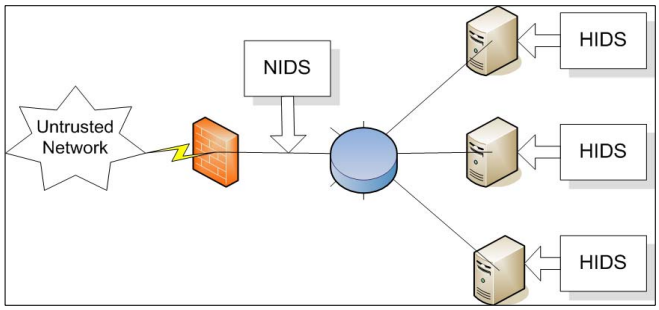


Fig. 2. – IDS on the basis of their Architecture [14]

**1.** Host-based Intrusion Detection System (HIDS): This kind of System is placed on a Host as an Agent. They examine the activities on each host autonomously which include sniffing the Network traffic coming through and going out towards the Host.

**2.** Network-based Intrusion Detection System (NIDS): This sort of System is placed on a Network. It examines the activities at the Network which include sniffing (the Cable Wire and Wireless Devices') packets and then matching them with the signature database to monitor the detection of an attack.

*TYPES OF IDS ON THE BASIS OF DETECTION METHOD:*

**1.** Signature Based Detection: This class of attack looks for a pattern or a signature to match with the incoming packets from the database. So that same attack can be prevented in future from happening again.

**2.** Anomaly Based Detection: It inspects ongoing traffic and activities for any erratic behaviour on network, system that could recognize an attack.

## II. RELATED WORK

[1] Piyush A. Sonewar et al. in their work identified the threats of SQL injection and XSS Attack using .NET Framework of Windows OS. They also signified use additional security measures provision using stored procedures. The approach applied mapping model to detect SQL Injection and XSS Attack.

[2] Rathod Mahesh Pandurang et al. concluded that IDS based on a mapping model is constructed to detect and prevent SQL Injection and XSS attack. They emphasize that the restriction of the damage created by an attacker to a container will be confined to that container only if the client has its container. They observed the excellence in the work of their system based on average pace time memory pages per second compared to existing one.

[3] Jinkun Pan and Xiaoguang Mao et al. found DOM XSS Micro, a Micro Benchmark for measuring DOM based XSS vulnerability also proposed a study of 6 DOM based XSS detection tools to show the use of DOM XSS Micro they plan to propose the benchmark with more betterment of each component (like complex lang. features, web framework and Libraries and Browser quirks) making it a standard benchmark.

[4] Akash Garg et al. concluded that IDS system helps in detection of dangerous attacks. Signature based IDS has usefulness for detection of known attacks whereas attack is detected by anomaly-based IDS. Snort is an open Source IDS Solution for detection if attacks as well as for prevention also by blocking the connection thus stopping entrance of any malicious attack.

[5] Hu Zhengbing et al. concluded an algorithm to find signature of the related attack quickly, He applied scan reduction method to decrease the scanning time for a database this enables us to discover out new attacking signatures more efficiently.

It is clear from the above literatures that none of existing work, none of techniques has a solution to protect web application from XSS attack. Therefore, it is an open challenge to the researcher to find an optimum solution against the XSS attack according to current cyber-attack statistics.

## III. PROPOSED WORK

Our motive is to detect most common attack of all time i.e. Web Application attack (which includes SQL Injection & XSS). Here we are proposing rules to identify XSS (Web Application) attack with the help of IDS, and we will monitor our incoming & outgoing packets which will further match with our database rules. This section provides method established by SNORT IDS Method.

IDS Working Model as Showed in Fig 3.

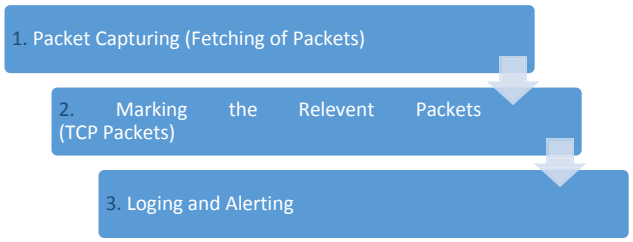


Fig. 3. – IDS Working Steps

1. The first process includes capturing of all the packets that are in transition on the network.
2. This step involves the selection of relevant packets that are filtered. Here we are doing selective packet capturing.
3. In this steps, we apply signature based detection on the selective packets in this case we are only using TCP packets.
4. After detection, the alerts are generated, and metadata log are created

*SNORT*

Here we are using Cisco tool SNORT IDS which is free, lightweight and Open Source tool. We can create rules according to yourself as you need them to be. We will install SNORT on the Network to inspect the network traffic data, i.e. all packets in transition and then filter out packets according to rules provided on the tool Snort.

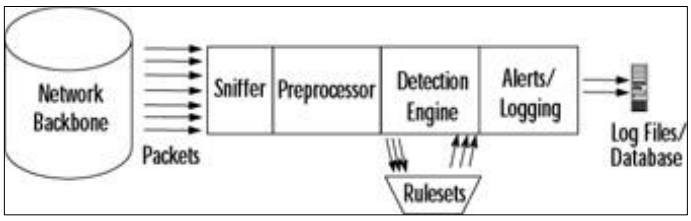


Fig. 4. – SNORT Architecture [13]

Components of Snort architecture as shown to Fig. 4. :

1. The Sniffer: It collects the traffic packets on the web. After collection process, it transfer the raw packet to the next component i.e. pre-processor.

2. The pre-processor: It performs certain action to conclude what kind of packet and its behaviour Snort is dealing with. After this job, packets are in transition towards detection engine.

3. The Detection Engine: It compares every packet to the predefined rule on snort.conf file. If the packets match with the rule, then it is forwarded to the output.

4. The output: It will trigger the alert system as well create a metadata log. The log can be saved in the variety of formats according to need. The Alert file is also generated which also contains the evidence of the attack.

*PROPOSED SNORT RULES*

Here we are introducing snort rule that can detect XSS attack with efficiently and create an alert entry for in snort alert file:

alert tcp any any -> any any (msg:"XSS Regular Expression Rule Matched"; pcre: "/((\%3C)|<)[^\n]+((\%3E)|>)/i"; sid:1001008)

This expression with all of their possible values in hexadecimal values for response in text field by the user are considered to detect the attack. It must be noted that all the script or tag are written between < and > brackets to insert a malicious script inside a database on the server side.

Here in Fig. 5. A Finite state machine is created to show the snort rule functioning that we have created. In here Q consists of all the states element while $S_1$ being initial state and $S_3$ as final state in between these states and < > brackets user input is captured (alphabet, numerical, special-character). If rule is matched, then alert file is generated as shown in Fig. 7.

This involves selective packet capturing for our database we are using TCP data packets only the working of rule is shown in Finite state machine for our above Snort rule:

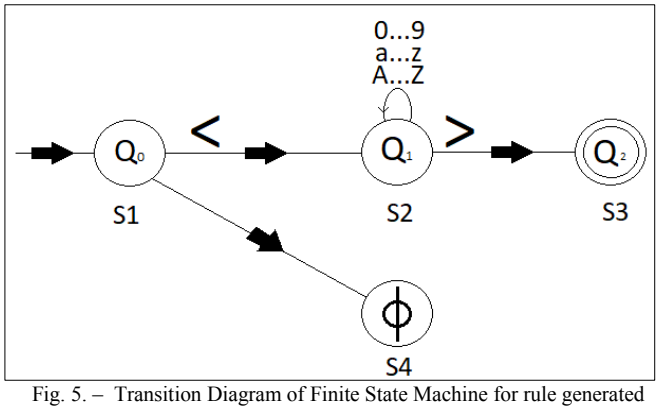


Fig. 5. – Transition Diagram of Finite State Machine for rule generated

Transition Function of Finite State Machine :

$S_1$, $S_2$, $S_3$, $S_4$ are states; $S_1$ and $S_3$ as initial and final state

$Q = \{Q_0, Q_1, Q_2, \phi\}$ *Elements Of States*

$\delta(Q_0, <) = \{Q_1\}$

$\delta(Q_0, \#|\^|\$|..|+) = \{\phi\}$ *$S_4$ Null State*

[Any i/p except < will result in $\phi$ in state $S_4$]

$\delta(Q_1, A \mid B|...|Z) = \{Q_1\}$

$\delta(Q_1, a \mid b|...|z) = \{Q_1\}$

$\delta(Q_1, 0|1|...|9) = \{Q_1\}$

$\delta(Q_1, >) = \{Q_2\}^*$ *$S_3$ Final State*

## IV. EXPERIMENTAL SETUP AND RESULTS

A test bet topology has been implemented consist of 4 systems, which comprised of a Server and 3 PC attacked the server, and all the packets on the server were collected using Wireshark. After the collection of dump files, we analysed using Wireshark to create Snort Rules.

Table – 1: Configuration of Summary Experimental Setup

| SYSTEM NO. | Software Installed | Version | System Processor | IP Address |
|---|---|---|---|---|
| 1 (Server) | Wireshark | 2.2.3.0 | Intel 3rd Gen (i7) | 172.16.50.241 |
| | SNORT | 2.9.9.0 | | |
| | XAMPP Server | 3.2.2 | | |
| 2 (Client) | Windows | 10 | | 172.16.50.76 |
| | Google Chrome | 57.0.2987 | | |

Snort was configured according to the proposed system, and then rules were also introduced in Snort to detect the XSS attack efficiently with the help of deep packet inspection (DPI) that works on Application layer of the OSI reference model, it examines packet and its packet header. We tried to attack server using various common tags and scripts on the server side then we captured the attack using wireshark then we introduced rule to snort for detection. When we ran the console, all attacks were detected effectively then it created alerts and logs successfully as shown in Fig 6 given below.

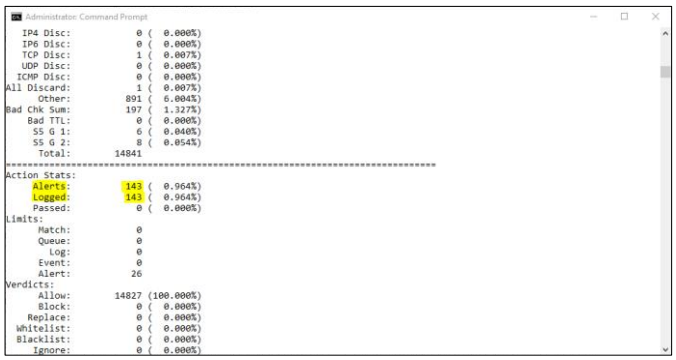


Fig. 6. – Packet Capturing Summary

After successful implementation of rule, XSS attack was able to be detected through Alert File containing details of the attack and attack was also logged in the form of dumps as metadata in same directory in snort.

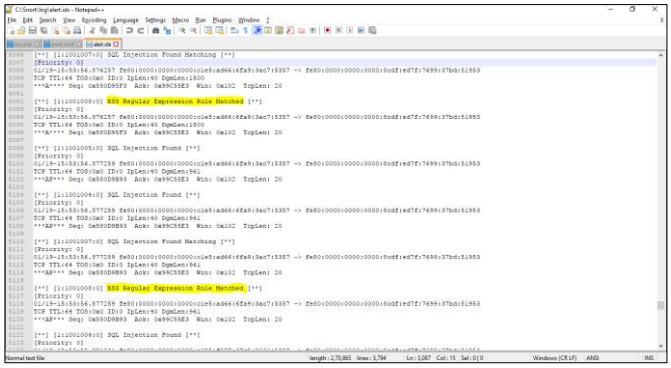

Fig. 7. – Alert File

## V. CONCLUTION

In this work snort rule are introduce to detect XSS attack. Experiments has been done in real network environment. To Improve the speed of detection, we must use “content” or “uricontent” with the PCRE. Few false-positive were generated, but it caught all the scripting attack successfully.

In future, we can create more rules for another type of attacks by using the vast and diverse dataset which will allow us to be more efficient in detection of the attack. The speed of the detection process can be increased using other keywords with Perl Compatible Regular Expression (PCRE) in future.

## VI. REFERENCES

[1] Piyush A. Sonewar , Nalini A. Mhetre “A Novel Approach for Detection of SQL Injection and Cross Site Scripting Attacks” International Conference on Pervasive Computing (ICPC) 2015.

[2] Rathod Mahesh Pandurang, Dr. Deepak C. Karia “Impact Analysis of Preventing Cross Site Scripting and SQL Injection Attacks on Web Application” 2015 IEEE Bombay Section Symposium (IBSS)

[3] Jinkun Pan, Xiaoguang Mao “DomXssMicro: A Micro Benchmark for Evaluating DOM-based Cross-Site Scripting Detection” 2016 IEEE TrustCom/BigDataSE/ISPA2324-9013/16 2016

[4] Akash Garg, Prachi Maheshwari “Performance Analysis of Snort-based Intrusion Detection System” IEEE 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS -2016), Jan. 22 – 23, 2016, Coimbatore, INDIA

[5] Hu Zhengbing, Li Zhitang, Wu Junqi “A Novel Network Intrusion Detection System(NIDS) Based on Signatures Search of Data Mining” IEEE 2008 Workshop on Knowledge Discovery and Data Mining 10-16

[6] Hossein Jadidoleslamy “Weaknesses, Vulnerabilities And Elusion Strategies Against Intrusion Detection Systems” International Journal Of Computer Science & Engineering Survey (IJCSES) Vol.3, No.4, August 2012.

[6] Mr. Chandrapal U. Chauhan Mrs. V.A. Gulhane“ Signature Based Rule Matching Technique in Network Intrusion Detection System” International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 4, April 2012.

[7] Kapil Wankhade, Sadia Patka and Ravinndra Thool,“An efficient approach for intrusion detection using data mining methods”, IEEE, 2013.

[8] Natttawat Khamphakdee, Nunnapus Benjamas and Saiyan Saiyod, “Improving intrusion detection system based on snort rules for network probe attack detection”, International conference on information and communication technology, IEEE, 2014.

[9] Alnabulsi, H.; Islam, M.R.; Mamun, Q., “Detecting SQL injection attacks using SNORT IDS,” Computer Science and Engineering (APW Con CSE), 2014 Asia-Pacific World Congress on , vol., no., pp.1,7, 4-5Nov. 2014.

[10] OWASP https://www.owasp.org/index.php/Top 10 2013-Top10”.

[11] Johari, R.; Sharma, P., “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, vol., no., pp.453,458, 11-13 May 2012.

[12] 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.

[13] Snort's Features http://flylib.com/books/en/3.100.1.200/1/

[14] Types of IDS (NIDS and HIDS) https://keamanan-informasi.stei.itb.ac.id/2013/10/30/menangani-serangan-intrusi-menggunakan-ids-dan-ips/

[15] Alnabulsi, H.; Islam, M.R.; Mamun, Q., "Detecting SQL injection attacks using SNORT IDS," Computer Science and Engineering (APW Con CSE), 2014 Asia-Pacific World Congress on, vol., no., pp.1,7, 4-5, Nov. 2014.