ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

Analysis of Input Manipulation Attacks in Web

K.G. Maheswari

Assistant Professor (Senior), Department of MCA, Institute of Road and Transport Technology, Anna University, Erode, Tamil Nadu, India

ABSTRACT: With the tremendous development of internet with web-based applications such as online banking e-commerce etc, have been increased popularly using web applications. Due to the availability of high-level tools and technology, have made the attacker to attack the target easily in different forms. So, the security has become the big challenge for the web application and other internet services. In this paper Hybrid detection technique has been proposed and implemented. The proposed Hybrid Detection Technique algorithm achieves better accuracy than compare to all other existing algorithm, to find SQL injection and XSS attacks detection. Form the experiment, it has been shown that the proposed algorithm help to detect the input manipulation attack such as SQL injection attack and XSS attack efficiently .

I. INTRODUCTION

For the past one decade, many organization and most government activities are changed from the traditional approach to web based applications. Web application is a program that runs on a computer with web server, via user agent web browser. In general, all distributed web based applications are normally referred to as web application or web app i.e. an application which is accessed by the web through the internet. When the application can be accessed from the database through the server by an unauthorized person, it is considered as a malicious attack. The malicious attack can occur due to many reasons like source code has a loophole which could be exploited, application program interface not properly validate, user input query may not be validated, and also poor programming skills and so on.
When the dynamic web request interface with the web server or database server, attacker manipulate the web request. Suppose it is not properly checked, it will lead to the input manipulation attacks, namely the SQL injection attack (SQLIA) and the cross site scripting (XSS) attack. These attacks are occurs due to manipulation with already exiting query or script by attacker.

II. RELATED WORK ON WEB BASED INPUT MANIPULATION ATTACK

There are many works in the literature that deal with detection of SQL injection attack in web environment (Artzi, S, et al. [1] ,Livshits, B,et al. ([2],Manmadhan, S&Manesh [3], Kruegel, C, et al.[4] Among them , Yeole, AS &Meshram, BB [6] have analysis seven different technique for detection of SQL Injection in client side and server side both in static analysis and dynamic analysis and they suggest server side input validation technique is most important because most of the attacker by pass the client side validation .

Khoury, et al. [7] evaluated three states of art black-box scanners that support detecting stored SQL injection vulnerabilities.They developed custom test bed that challenges the scanners capability regarding stored SQL injections. The results show that existing vulnerabilities are not detected even the automated scanners are taught to exploit the vulnerability. Because black-box scanners identified weaknesses reside in many areas like crawling input values,

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

attack code selection, user login, analysis of server replies, and the process functionality. Due to the poor detection rate, they examine the different phases of black-box for progress of detection rate stored SQL injection vulnerabilities.
Ali, S, et al [8] scheme adopts the hash value approach to further improve the user authentication mechanism. They use the user name and password hash values. SQLIPA (SQL Injection Protector for Authentication) prototype was developed in order to test the framework. Wassermann, G, et al [9] present generating test inputs algorithm to analysis the web application and it used to get information from static pages and dynamic language also support for the test . For this approach constraint resolution algorithm model was proposed to improve scalability. Sheykhkanloo, NM [10] have proposed pattern recognition NN model using machine learning technique and also they perform the classification of SQL injection attack. Lee, I, et al. [11] have proposed a simple detection method for SQL injection attack and compare the proposed method with dynamic SQL query attribute value of web pages which is already exiting. If the query attribute value match then remove the query attribute value. This method is based on both static and dynamic analysis.

Fonseca, J et al. [12] presented a field study on two of the most widely spread and critical web application vulnerabilities, SQL injection and XSS. They analysis the source code of web applications written in different languages. Results show that small subset of software fault types, affecting a restricted collection of statements and it will affect the outcomes of this study can be used to train software developers and code inspectors in the detection of such faults, and are also the establishment for the research of realistic vulnerability and attack injectors that can be used to assess security mechanisms, like intrusion detection systems, vulnerability scanners, and static code analysers. Hydara, I et al. [13] have proposed genetic algorithm to detect the XSS and remove XSS from the source code and they implement in the java language for their work ESAPT escaping rules are used. Mewara, B, et al [14] discuss various browser evaluation line IE, grome, and firebox and add-ons, and using this evaluation to find the XSS attacks dynamically. It provides client side solution for the XSS attacks also.
Selvamani et al.[15] proposed browser side protection scheme. The major ability of client side solution effectually decreases XSS attacks. It presents an XSS defence without depending on web application providers by supporting an extenuation mode that greatly diminishes the amount of associated alert prompts whereas, simultaneously, it provides protection against stealing sensitive information such as session ID and cookies. This paper used a procedure to decide if a resource requested is through a native link by examining the Referrer HTTP header and matching its domain with the requested webpage domain. The value of the domain is demarcated by way of splitting and analysing URLs. From the research knowledge this is a noble approach but it drops the performance as it follows numerous phases to resolve whether the website is defenceless or not. Maheswari KG et al [15,17,18] discuss the various types of SQL injection attacks , XSS attacks and detection of input manipulation attacks in web and cloud environment. Nalinipriya G et al [19,20] discuss the various security schemes to detect the different types of attacks in cloud environment.

1.1.1 Web application architecture

The web application is a software, which can be accessed by only through a web browser on the client machine. Web application architecture is three tires architecture model. The first tire is client as a web browser and it is a user interface. The second tire as web server it gets the inputs from both user interface and database server. The third tire is database server. It manages all the determined data.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

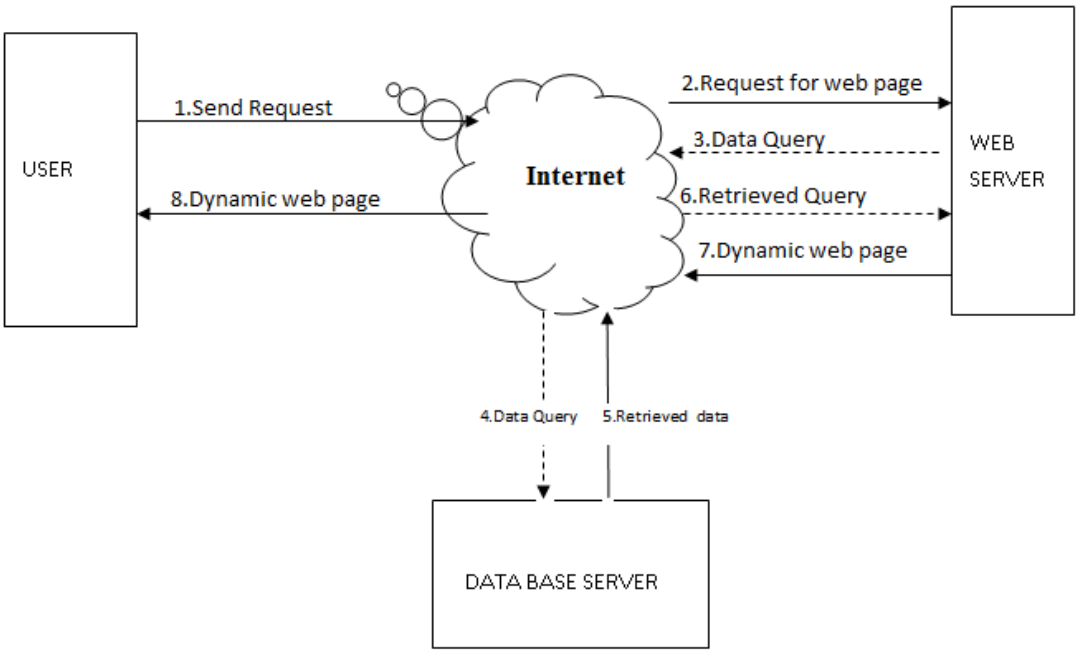


Figure 1.1 Web Applications Architecture

The Figure 5.1 explains the architecture for web applications. When the user send the web request query to the web server, the web server again send the query to the database server. When web server dynamically construct code for input between tiers and rise the input validation for the web application. One tire output is input for other tire. In each tire the input string may check or modify before processing from one input tire into another tires for execution. If it is not properly validate the input, it will generate the web application security problem.

1.1.2 Web application vulnerabilities, threats and risk

The vulnerabilities in application layer are very hard to define. Attacker can manipulate input and get confidential data without detected by network define system. The objective of the (OWASP) open web application security project is educating and informing developers about application vulnerabilities. For three years one, it will realise top10 web application attacks and the latest one sis in 2013.

Internet threat reports shows that web application attack have been increased over last five years (Source:https://www.info-pointsecurity.com/sites/default/files/cenzicvulnerability-report-2014). Before 2009, the vulnerability research report shows that SQL injection attack and XSS attack are tops of the list. But in the year 2014, security report, smart phone application attack are topped the list. In the web application, user give the input to the web server through web browser. The web server provide web request to the database server with dynamically generated SQL query and get the result from the database server. Suppose the input is not properly evaluated by the input validation method, it will lead input validation based attack.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

1.1.3 Input manipulation vulnerability

Input manipulation attack defines as "using the unexpected inputs attacker can get the desired output. The most predominant input based manipulation attacks are SQL injection and XSS attacks. Because of low level API (Application Program Interface) between with the browser and database. An attacker can inject SQL query or HTML document causes the vulnerabilities in application server to produce malicious SQL query and HTML documents.
The web application server construct dynamic queries and HTML document through low level string manipulation. The manipulated queries are isolated into lexical entities. This is common in scripting language like java script. Suppose some part in application code may incorporate by the attacker, the modified user input on the one tire executed by another tire. So low level integrity data is used in high integrity channel that is the database server executes manipulated code with the permission of web application server. When the manipulation is done in SQL query, it is called as SQL injection attack. When the manipulation is done in scripting language, it is called as XSS attack (cross site scripting).

1.1.4 Malicious Application

Malicious means unwanted (or) harmful to the system and application. In web application development, some source code structure have some vulnerabilities using that an unauthorized person can perform unwanted activities or compromise the data integrity and confidentiality. This type of compromised application is called malicious applications. There are different techniques are available to explore the malicious application and some of the exiting detection techniques are listed below in the Table 5.1

Table 1.1 List of Existing Detection Techniques

Techniques	Work flow
Brute – force Technique	When an attacker using guessing name etc to get the access into the application.
Web Spridering	The web request URL is parsed into links and then requesting those links repeating until new content is found .
Discovering Hidden	The searching contents in an application are not direct link to other web pages.

1.2 SQL INJECTION OVERVIEW

When an unauthorised person can access database through the database server. It is considered as malicious attacks. When dynamically generated query manipulated with attacker's injected query, at the time of database server execution. It will leads the manipulation attacks like the SQL injection attack, tautology, inference, union query and piggy backed queries.

1.3 MANIPULATION METHODS OF WEB APPLICATION

There are two most possible methods to manipulate the web request in the web environment and these manipulations are based on mostly string manipulation.

1. Manipulate through login page/ user input
2. Manipulate through URL

MANIPULATION THROUGH LOGIN PAGE / USER INPUT

Through the login form it is possible to manipulate the query and access the database server. For example Figure 5.2 shows the "login form" which is written in JSP for server side source code.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

```
1.String Login Action (HttpServletRequest request,…) throws IOException {
2.String customer_login = getParam(request, “Login”);
3. String customer_password= getParam(request, “password”);
4.java.sql.ResultSet rs =null;
5.String qry= “select customer_id, customer_nam from customer _detaikswher “;
6.qry = qry +”customer_login=”+ customer_login+” and customer_password=”+ customer_password+
“”;
7.java.sql.ResultSet rs = stat,executeQuery(qry);
8. if(rs.next())//passed login and  password
9. session.setAttribute(“customerID,rs.getString(1));
……}
```

Figure 1.2 Login form for server side source code

In this example line2 and line3 used to get the user input values and stored in the customer_login and custome_password. The corresponding dynamicSQL query is generated in the line5 and line6. Suppose the user gives the valid login is ‘user’ and the password is “guest”. In the line6 it will generate SQL query as shown in Figure 5.3. The line7, the generated query will execute and authentication is valid at line9.

```
select  customer _id, customer_name from customer_details where  login =”user “  and password =”guest”
```

Figure 1.3 SQL generated query for the login form

Suppose in the line2, the input value is given as "'or 1 = 1 "-- the second input field is blank. The request query becomes as shown in figure 5.4

```
select  customer _id, customer_name from customer_details where  login =”‘ OR 1=1 -- “  and password =” ” ;
```

Figure 1.4 Manipulated SQL generated query for the login form

In this query first check the condition, 1=1 this is true, and then it will give the result. The single cods (‘) used to terminate the string and [--] is used to mark the SQL comment and ignored the reaming query. This is the tautology type of SQL injection. Using this type of attack, attacker gain confidential information, to manipulate the database, and by pass authentication mechanisam.

2. MANIPULATE THROUGH URL

1.3.1 OR Condition

In URL manipulation, using ‘or condition’ attacker can manipulate the SQL query statement. It is very serious attack. For example When an user send the web request, the original structure is as shown in Figure 5.5

```
http// www. mydomain.com/ item/ iteam.asp? item_ id = 01
```

Figure 1.5 The original HTTP web request

The corresponding SQLquery structure of above web request, as shown in Figure 5.6

```
select item_no, item_name from  item where  item_ id =01
```

Figure 1.6 Dynamic SQL web request

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

In the above web request, item_id is not validated, so attacker can able to manipulate the malicious SQL statement as shown in Figure 5.7 & Figure 5.8.

```
http// www. mydomain.com/ item/ item.asp? item_ id = 01  or 1 =1
```

Figure 1.7 Manipulated HTTP web request

```
select  item_name, item_price from  item where  item_ id =01 OR  1=1
```

Figure 1.8 Dynamic manipulated SQL statements

The 'OR condition' is always true and all records from the corresponding item_id will display.

1.3.2 Stored Procedure

Another type of manipulation attack is stored producer. It is also very serious attack. For example, in the Figure 5.9 shows that web request, (;) semicolon indicate to the server that multiple statements in a single line execution.

```
http// www. Mydomain.com/ item/ iteam.asp? item_ id = 01 ; DROP  TABLE  item.
```

Figure 1.9 Manipulated web request stored procedure

The server first execute the second query, DROP TABLE procedure, which causes the entire product table is deleted in SQL server.

1.3.3 Union Select Statement

Using this statement an attacker can get the information from the different table. For example Figure 5.10 & 5.11 describe malicious union select statement

```
http// www.mydomain.com/ item/ iteam.asp? item_ id = 01 UNION SELECT username,
password from  users ;
```

Figure 1.10 Manipulated web request using union select statement

The equivalent SQLquery is

```
select item_ no, item_ name from item WHERE item_ id =01
        UNION SELECT username, password from users;
```

Figure 1.11 Manipulated SQL query using union select statement

Many web application belief that the SQL query is trusted to use real time environment, this assumption enable the attacker to exploit SQL injection in real time environment. The quick solution for these types of injection is to avoid display of database error message, minimize the use of stored procedure and sanitizing the user input values. But these security measures are not enough for highly sophisticated SQL injection attack. The Table 5.2 shows the some of examples for SQL injection attacks in web environment.

Table 1.2 Examples for SQL injection attacks in web environment

SQL injection Type	**Examples**
Tautology	"' OR 1=1 --
Union	'UNION SELECT 1.1--

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

Piggybacked	'; show tables; -- '; shutdown;--
Inference	/*select concat ('1','2') */
Hex encoded query	0x206F7220313D31 ,this hexadecimal representation of the string "' OR 1=1 --

Attacker BenificsThrough Manipulation

- Disclosure of stored data
- Misuse of stored data
- Data leakage
- Un authorisation Control

1.4 CROSS SITE SCRIPTING (XSS) ATTACKS OVERVIEW

In early stage, web application programs are written only by the java scripting language. So it is possible to manipulate script in web page may added into the content of a trusted website. When the user visits a page on the site, the added script is loaded and executed in user's browser with the trusted sites privilege. Then injected script gain privileged information like cookies and confident content from the site. such a unwanted manipulated script is called a cross site scripting (XSS) attack. The malicious scripts utilize the visitor's browser and do malicious activities Kerschbaum, F(2007). Consider the following example, an user requests a web page and the requested responds is a personalized web page which is given below in Figure 5.12 and manipulated script in the Figure 5.13.

```
http://www.website.com/index.php?name=uma
<html>
<body>
 Uma
```

Figure 1.12 Original HTTP script

Suppose an attacker can craft the input with additional scripting language like given Figure 5.13.

```
http://www.website.com/index.php?name=uma<script>alert("XSS-Script")</script>
```

Figure 1.13 Manipulated script

Malicious script injected in the web request. Then response may be in Figure 5.14.

```
<html>
<body>
Hello, uma
<script>alert("XSS Script")</script>
```

Figure 1.14 Manipulated web request response

The URL of the web request would be executed in the attacker user's security context with web site. The consequences can be more severe like Cookie theft, monitoring the activities or fake transactions etc.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

1.4.1 Anatomy of XSS Attack

In general, an XSS attack involves three actors the website, the victim, and the attacker. The Figure 5.15 explains the input manipulation JavaScript code through victim's browser.

```
<script>
window.location="http://evil.com/?cookie=" + document.cookie
</script>
```

Figure 1.15 Malicious scripts by the attacker

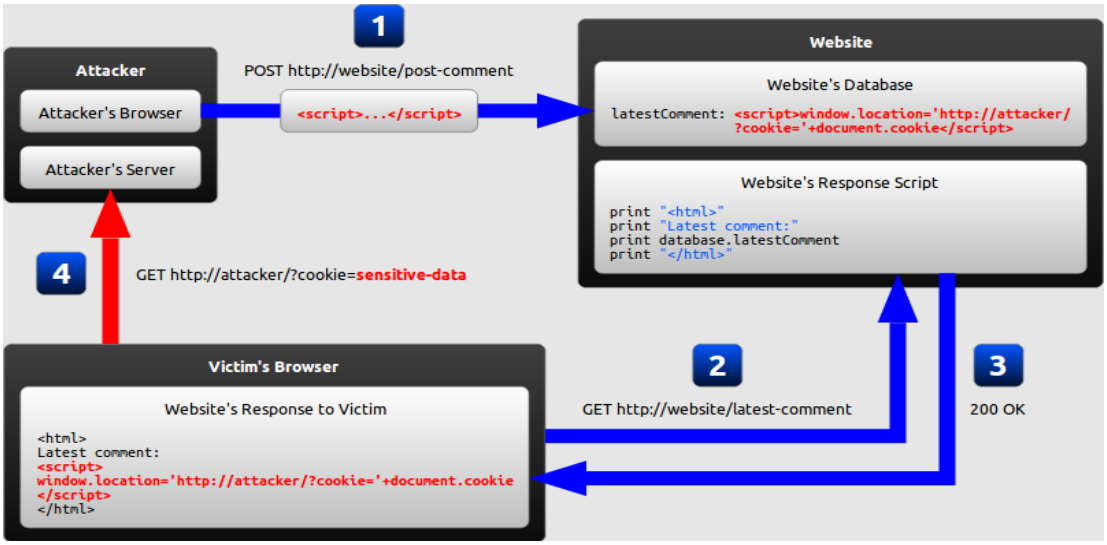


Figure 1.16 Simple XSS ATTACKS (Source https://excess-xss.com)

When an attacker, inject a malicious script into the website database. Suppose that particular website request by the victim, the malicious script in the database will sends it to the victim. The victim browser execute the script and also send the cookies to attacker server.

Steps for executing the XSS attacks through URL

- Attacker send web request (URL) with hidden script.
- Legitimate user follows the URL containing the script.
- Web server process the user request web pages.
- User's browser execute the malicious script.

III. PROPOSED WORK

Figure 5.20 shows the proposed architecture of hybrid detection technique which is used to detect the web based input manipulation attacks SQL injection attack and XSS attacks in dynamic web environment.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

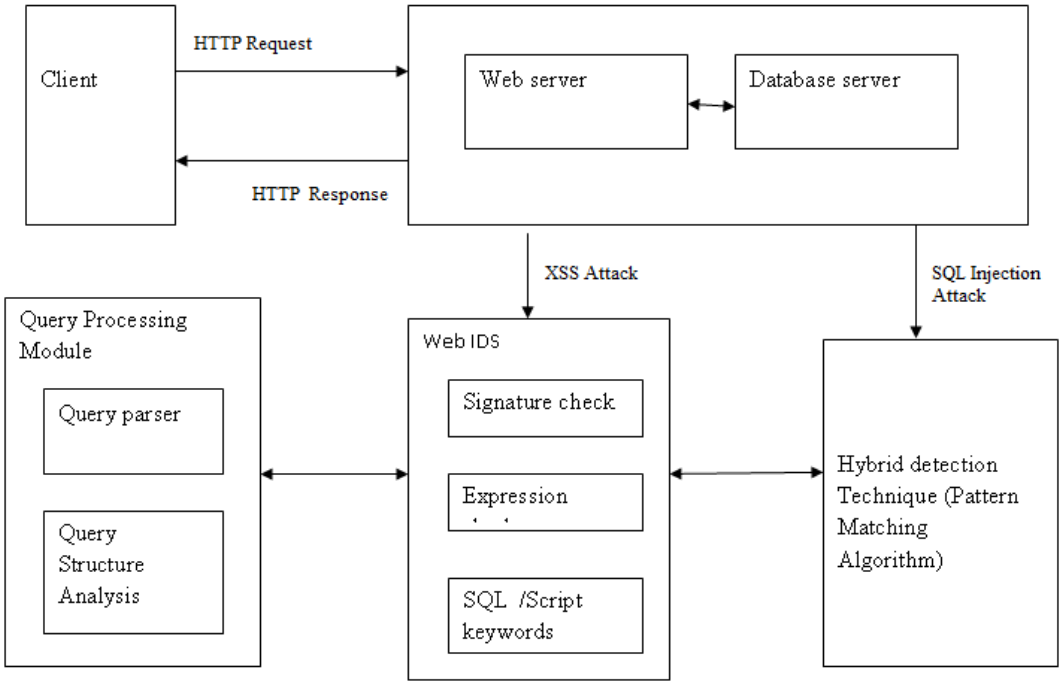


Figure 1.17 Architecture of Web Hybrid Detection Techniques

This proposed architecture consists of three modules Query processing modules, web intrusion detection system and detection module which consists of string matching based hybrid detection technique.

1.4.2 Query Processing Module

This module consists of two components query parser and query structure analysis used to analysis the URL web request. Table 5.3 shows the URL components. In this module, URL web request separate into the three components hostname, path and parameter and further separated into tokens. For example

http://www.website.com/index.php?name=uma

Table 1.3 URL Components

http	www.website.com/index.php	name=uma
Host Name	Path	Parameter

1.4.3 Web Intrusion Detection System

Web IDS used to detect the manipulated attacks, which consists of three components signature check, expression check and SQL /Script keywords.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

Signature check

This component is used to check SQL and XSS attack signature and Table 5.4 and Table 5.5 show the SQL injection check and the XSS attack check signature. **Input : HTTP Anomaly Request**
Output : SQL injection , XSS attack

STEP 1: Get the Input from HTTP CSIC 2010 Dataset
STEP 2: Check the method GET, POST and PUT
STEP 3: perform the SQL Injection Test
Count the total number of SQL Attacks
STEP 4: perform the XSS Injection Test
Count the total number of XSS Attacks
STEP 5: Repeat the step1 to step 4

Figure 1.18 Hybrid detection techniques Algorithm

Detection of SQL injection

Figure 5.22 show the block diagram for detection of SQL injection attack and also explain the different types of SQL injection.

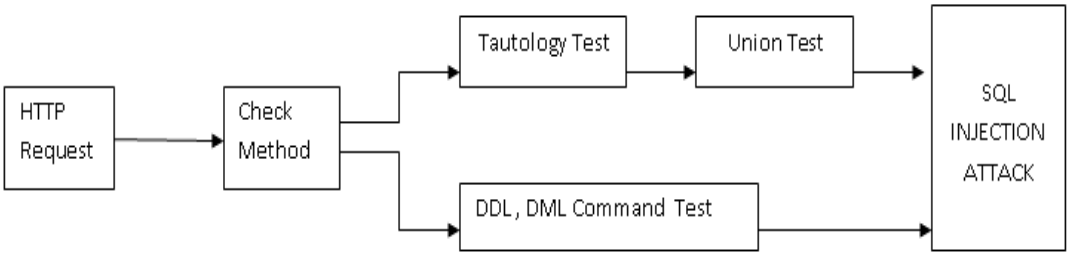


Figure 1.19 Block diagram for Detection of SQL Injection Attack

The First perform tautology test, which is the test condition of 'OR condition' is injected in the web request. Then it perform the UNION test,which is used to check, any additional condition is in the web request and after that check the data manipulation command (like insert, delete and update) is injected in the web request and also detect the DDL commands like drop is injected in the web request. The web request is also analysised and compare to the signature check and keywords which is defined in the Table 5.4 and Table 5.6 for the SQL attack.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

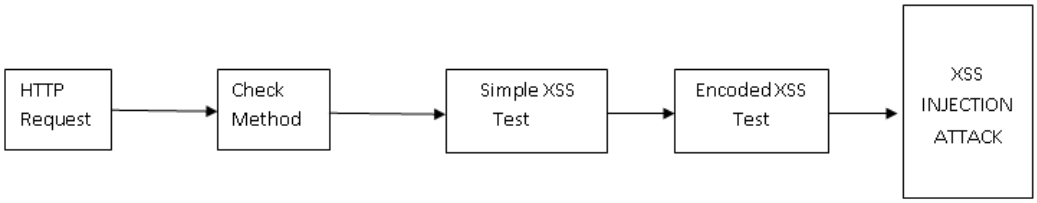
(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

Detection of XSS

In the proposed hybrid detection technique, the web request analysis is compare to the signature check and keywords. It is defined in the Table 5.5 and Table 5.6 for the XSS attack. The Figure 5.23 shows the block diagram for detection of



XSS Attack.

Figure 1.20 Block diagram for Detection of XSS Injection Attack

The Figure 5.24, and Figure 5.25shows the samples are used in the alert message and encoded XSS test used to find the XSS attacks in the URL web request.

Figure 1.21 Sample Java Script Alert Message (Simple XSS)

```
<IMG SRC =”javascript:alert(‘XSS’):”>

<IMG SRC =javascript:alert(‘XSS’)>

<IMG SRC=javascript:alert(String.formCharCode(88,83,83))>

<IMG SRC=JaVaScRipt:alert(‘XSS’)>

<IMG “ “” ><SCRIPT>alert(“XSS”)</SCRIPT>
```

```
<script>alert(396606905532)</script>.

<ScRiPt %20%0a%0d>alert(396616905532)%3b</ScRiPt>

>”><scRIpt%20%0A%0d>alert(396616905532)%3b</ScRiPt>.

-- ><scRIpt%20%0A%0d>alert(3966666905532)%3b</ScRiPt>.

>’><scRIpt%20%0A%0d>alert(3966666905532)%3b</ScRiPt>.
```

Figure 1.22 Sample normal Encoded Java Script Alert Message

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

Table 1.4 SQL attack Detection Rate

Algorithm	No.of detected vulnerabilities	No.on.known SQL vulnerabilities	Detection rate(in percentage)
KMP	42	52	80.7
Boyer Moore	48	52	92.3
Hybrid Detection Technique Algorithm	51	52	98.7

Table 1.5 XSS Attack Detection Rate

Algorithm	No.of detected vulnerabilities	No.on .known XSS vulnerabilities	Detection rate (in percentage)
KMP	52	64	81.25
Boyer Moore	57	64	89.06
Hybrid Detection Technique Algorithm	62	64	96.8

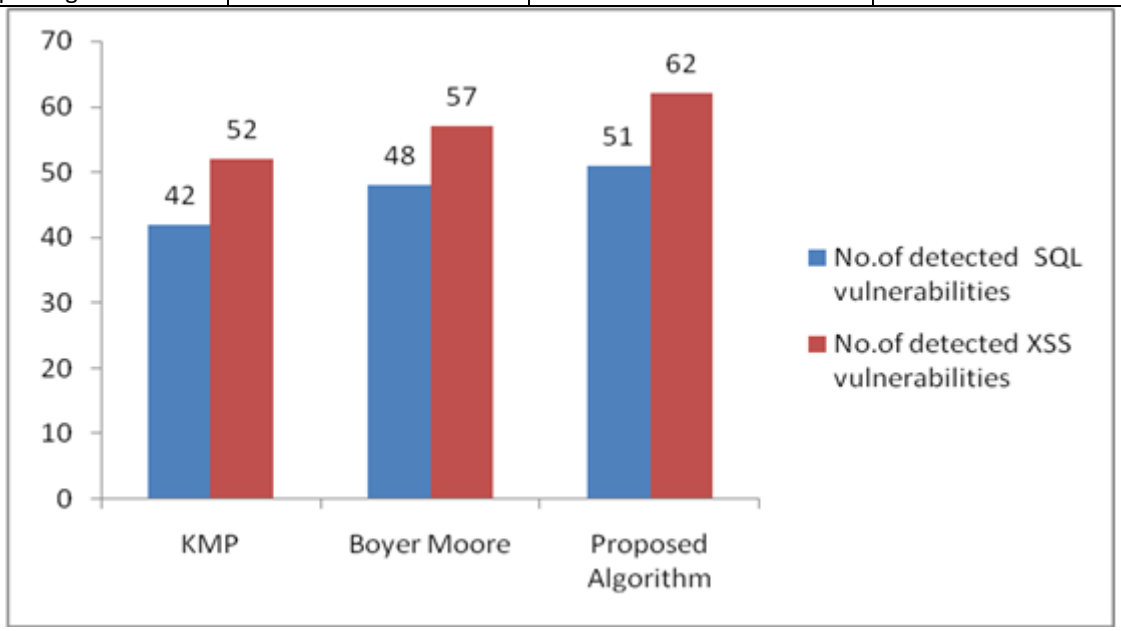


Figure 1.23 Comparison of web based attack detection

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

In Figure 5.26 X- axis represents various detection algorithms and Y-axis represent the accuracy of the algorithm and compare to other existing algorithm, the proposed hybrid detection technique algorithm performs good.

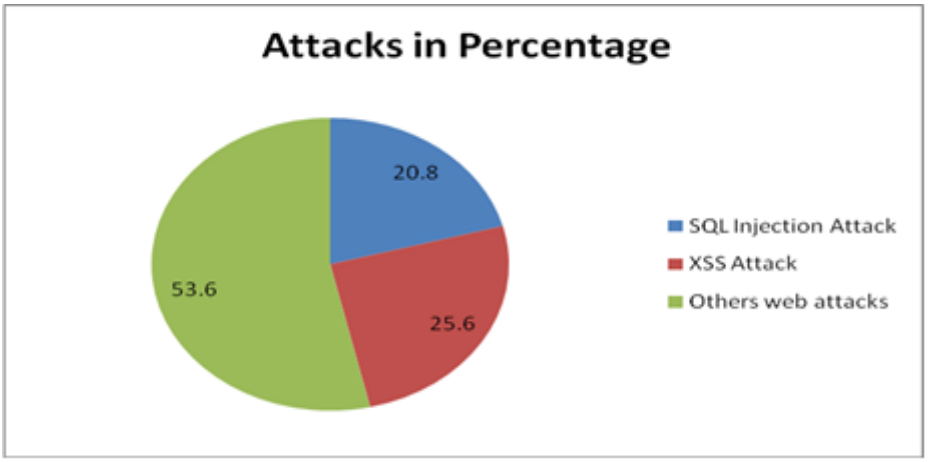


Figure 1.24 comparison of web-based attacks in HTTP CSIC 2010 dataset

The Figure 5.27 shows that the percentage of input manipulation attacks in HTTP CSIC2010 dataset. From this analysis, in the dataset, SQL injection attacks is 20.8 % and 25.6 % in the XSS attacks. The remaining attacks like session ID, stack overflow etc are included in the other types of web attacks.

1.5 PERFORMANCE ANALYSIS OF ALGORITHM

The performance of the proposed algorithm is measured in the different types of inputs and the results are shown in the Table 5.9.

Table 1.6 Performance analysis of SQL injection attack detection rate

Exp. No.	KMP **(in percentage)**	Boyer Moore **(in percentage)**	Proposed Algorithm **(in percentage)**
1	81.3	91.23	95.6
2	84.2	90.04	97.5
3	84.5	90.43	98.9
4	82.4	93.04	98.3
5	80.3	90.43	97.6
Average	82.54	91.10	97.57

The execution time of each algorithm, for the detection of SQL injection attack is tabulated in the Table 5.10

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

Table 1.7 Execution Time of Algorithm

Algorithm	Execution time
KMP	10 m 20 sec
Boyer Moore	5 m 10 sec
Hybrid detection technique	2 m 22 sec

Compare to other existing algorithm, the proposed algorithm in the detection of SQL injection attack, perform good in detection rate and execution speed.

The Table 5.11 shows the performance analysis of XSS detection with different types of inputs. From the table shows that, compare to other existing algorithm, the proposed algorithm performs good to the detection of XSS attack and the execution time of each algorithm is tabulated in the Table 5.12.

Table 1.8 Performance Analysis of XSS attacks Detection

Exp. No.	KMP	Boyer Moore	Proposed Algorithm
1	86.3	93.23	96.6
2	84.2	92.04	97.8
3	86.5	91.43	98.5
4	87.4	93.04	98.7
5	88.3	90.43	98.6
Average	86.54	92.034	98.04

Table 1.9 Execution Time of Algorithm

Algorithm	Execution time
KMP	12 m 15 sec
Moore	7 m 30 sec
Hybrid detection technique	3 m 21 sec

Comparison and the overall performance of analysis of SQL and XSS attack detection is shown in Figure 5.28

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

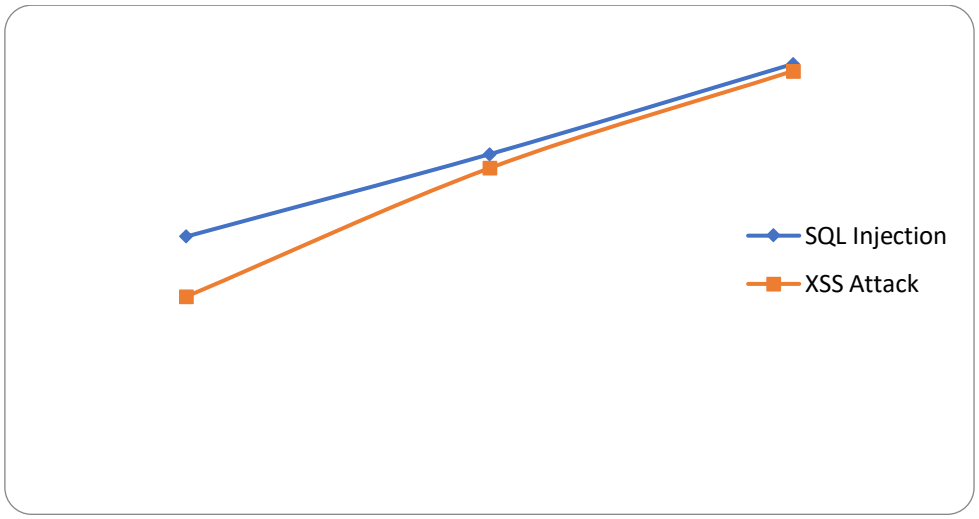


Figure 1.25 Performance analysis of SQL and XSS attack detection

X- axis represent various detection algorithms and Y –axis represent the accuracy of those algorithms. Out of all algorithm, our proposed algorithm performance well.

IV. CONCLUSION

In this work, from the renowned HTTP CSIC 2010 anomaly dataset is used to find two input manipulation attacks the SQL Injection and the XSS attack. Form the related work, it is known that various method have been proposed for detection of SQL injection and the XSS attacks. But in this work, hybrid detection technique has been proposed and implemented. Detection of accuracy of proposed algorithm has been compared to various string matching algorithms. The proposed algorithm has been implemented in java programming and empirical results are tabulated. Form the experiment,it has been shown that the proposed algorithm help to detect the input manipulation attack such as SQL injection attack and XSS attack efficiently

REFERENCES

1. Artzi, S, Kiezun, A, Dolby, J, Tip, F, Dig, D, Paradkar, A & Ernst, MD 2010, 'Finding bugs in web applications using dynamic test generation and explicit-state model checking', IEEE Transactions on Software Engineering, vol. 36, no. 4, pp. 474-494.
2. Livshits, VB & Lam, MS 2005, 'Finding Security Vulnerabilities in Java Applications with Static Analysis', Architecture, pp. 18-18.
3. Manmadhan, S &Manesh, T 2012, 'A Method of Detecting SQL Injection Attack To Secure Web Applications', International Journal of Distributed and Parallel Systems, vol. 3, no. 6, pp. 1-8.
4. Kruegel, C &Vigna, G 2003, 'Anomaly detection of web-based attacks', Proceedings of the 10th ACM conference on Computer and communications security, vol. 03, no. November, pp. 251-251.
5. Yeole, AS & Meshram, BB 2011, 'Analysis of different technique for detection of SQL injection', Proceedings of the International Conference & Workshop on Emerging Trends in Technology - ICWET '11, vol. 1, no. Icwet, pp. 963-963.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

6. Minhas, J 2012, 'Blocking of SQL Injection Attacks by Comparing Static and Dynamic Queries', International Journal of Computer Network and Information Security, vol. 5, no. 2, pp. 1-9.
7. Khoury, N, Zavarsky, P, Lindskog, D &Ruhl, R 2011, 'Testing and assessing web vulnerability scanners for persistent SQL injection attacks', in Proceedings of the First International Workshop on Security and Privacy Preserving in e-Societies, pp. 12-18.
8. Ali, S, Rauf, A &Javed, H 2009, 'SQLIPA: An Authentication Mechanism Against SQL Injection', European Journal of Scientific Research, vol. 38, no. 4, pp. 1450-1216.
9. Wassermann, G & Su, Z 2008, 'Static detection of cross-site scripting vulnerabilities', ICSE '08 Proceedings of the 30th international conference on Software engineering, pp. 171-180.
10. Sheykhkanloo, NM 2015, 'SQL-IDS: Evaluation of SQLi attack detection and classification based on machine learning techniques', https://www.researchgate.net/publication/281859247 SQL-IDS:, vol. 1, no. 1, pp. 1-10.
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, vol. 55, no. 1-2, pp. 58-68.
12. Fonseca, J, Seixas, N, Vieira, M & Madeira, H 2014, 'Analysis of field data on web security vulnerabilities', IEEE Transactions on Dependable and Secure Computing, vol. 11, no. 2, pp. 89-100.
13. Hydara, I, Sultan, ABM, Zulzalil, H &Admodisastro, N 2014, 'An Approach for Cross-Site Scripting Detection and Removal Based on Genetic', ICSEA 2014 : The Ninth International Conference on Software Engineering Advances, no. November 2015, pp. 227-232.
14. Mewara, B, Bairwa, S &Gajrani, J 2014, 'Browser's defenses against reflected cross-site scripting attacks', 2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014, pp. 662-667.
15. Selvamani, K, Duraisamy, A &Kannan, A 2010, 'Protection of web applications from cross-site scripting attacks in browser side', arXiv preprint arXiv:1004.1769.
16. **Maheswari KG** & Anita R 2015 "Generalization of minkowski distance metrics in mixed case analysis for web intrusion detection" in **International Journal of Soft Computing,** vol. 10, no. 4, pp. 274-278, ISSN ISSN 1816-9503.
17. **Maheswari KG** & Anita R 2015 "A dynamic tool for detection of xss attacks in a real time environment" **ARPN Journal of Engineering and Applied Science**, vol. 10, no. 10, pp. 4627- 4634, ISSN 1819-6608.
18. **Maheswari KG** & Anita R 2016 'A dynamic virtual machine security scheme against co resident attack in cloud computing' **Asian Journal of Information Technology**, vol. 15, no. 13, pp. 2116-2122, ISSN 1682-3915. (Updated Journal 2016 IF: 0.22)
19. Nalinipriya, **K.G. Maheswari**, Balamurugan Balusamy , K. Kotteswari and Arun Kumar Sangaiah , "Availability modeling for multi-tier cloud environment" **IntelIIgent AutomAtIon & Soft ComputIng,** Vol. 23, no. 3, PP. 485–492, ISSN: 1079-8587, 2017.
20. G. Nalinipriya, P.J.Varalakshmi, K.G.Maheswari, R. Anita, 'An Extensive Survey on Co- Resident Attack in Dynamic Cloud Computing Environment" published in **International Journal of Applied Engineering Research ,** ISSN 0973-4562 , Volume 11, Number 5 , pp 3019-3023 , **2016.**