International Journal of Security and Its Applications
Vol. 10, No. 5 (2016) pp.95-106
http://dx.doi.org/10.14257/ijsia.2016.10.5.09

A XSS Attack Detection Method based on Skip List

Shan Chun, Cui Jing, Hu Changzhen, Xue Jingfeng*,
Wang Hao and Mhagama Raphael

School of Software, Beijing Institute of Technology, Beijing 100081, China
**xuejf@bit.edu.cn*

Abstract

In recent years, the destruction of cross-site scripting (XSS) attacks is gradually increasing, but the existing XSS defense methods have some problems, such as, complex disposition, low performance, high non-response rates or high rate of false positives. Therefore, it is imminent to establish an efficient XSS defense mechanism. This paper presents a cross-site scripting attacking detection method, which is based on improved algorithm skip list. It can fulfill the rapid detection of cross-site scripting attacks by using some steps, for example, creating a skip list signature, detecting suspicious strings, marking attack vectors and so on. Experiment shows that compared with the traditional defense method of XSS attack, the new defense method can reduce the false alarm rate of detection, and improve the time performance effectively.

Keywords*: skip list, XSS, feature vectors, Web security*

1. Introduction

Cross-site scripting (XSS) is a security vulnerability that affects web application [1]. When it occurs due to improper or lack of sanitization of user inputs. The security vulnerability can produce many problems for users and server applications. An attacker can exploit cross-site scripting vulnerabilities, which can cause some damages such as Cookie or other user's information stolen by a third party, forcing pages jumping, performing Trojans. According to the annual OWASP Top Ten Web Application Security threat report in 2013 [2], the risk of XSS vulnerabilities is ranked as the third degree, XSS attacking does not affect the Web services side, but client is the actual attack victim. As an attack vector, Web server XSS vulnerability is used by malicious attackers to attack user client. Therefore, a success XSS attack has almost no attacks record in the Web server, which makes it difficult to find XSS vulnerabilities. According to official incomplete statistics of XSSED [3], there are 45,884 collecting XSS vulnerability popular sites, 3,025 of them have been repaired. It shows that most Websites have security risks and threats of XSS vulnerability; therefore, it is imminent to establish an efficient XSS defense mechanism.

There is only a few kind of existing XSS attack defense methods. Based on the structural integrity of a file, Yacin Nadji [4] raised a detect XSS method, which could effectively block XSS attacks. However, when it refers to the modification of the interpreter engine in client script, this method is complex and difficult to deploy. Xiali Wang proposes a behavior-based client XSS prevention method in her study [5]. Through certain behavioral analysis, this method can prevent dangerous HTTP request packet, and can make the suspicious request send to the user for judgment. This method is a pure client prevention method, it does not have universality, and ordinary users do not recognize its risks or not, this method is suitable for advanced users. Matthew Van Gundy

* Corresponding Author

and others presents a security policy in the literature [6], it divides the clients into trusted and untrusted group. Through implementation of the security policy to prevent XSS attacks, this method integrates Web server verbosely, but, it is not flexible. Xuejie Zhao presents a Prior Merge algorithm in the literature [7], his method is to dig frequent ordered set in order to obtained XSS attack patterns. This method has high rate of false positives and operational complexity. Cao Wen and others present an improved algorithm of hash tree in the literature [8], which combines with the finite state machines to extract similarity algorithm attack vector to detect XSS attacks. Nevertheless, hashing tree has high spatial complexity, and has lower rate of taking prime numbers and featuring comparison. Due to the particularity of XSS attacking scenario, the testing strategy of it has many problems, such as, cross-platform, deployment, simplicity, feasibility and performance testing, *etc*. There have not been ideal testing defense strategies to protect against XSS attacks. Therefore, this study puts forward an algorithm based on improved skip list library. This method can combine with the finite state machine with a threshold, in order to protect against XSS attacks rapidly and accurately.

This paper analyzes the characteristics of the sample provided by HA.CKKERS, it adopts an improved level step skip list generation algorithms to build skip list feature library. In this paper, it detects XSS attacks efficiently through a number of steps, such as scripts standardization, MD5 coding, pre-screening, and multi-mode screening. Experiments show that, through POST and GET methods， by using some test sample provided by XSSed sites, the method reduces the false alarm rate, and improves the time performance effectively. The remaining sections of this paper is organized as follows: Section 2 discusses the improved Skip list algorithm. Section 3 describes the cross-site scripting attack prevention methods based on Skip list. Section 4 discusses how to use the pre-screening to enhance the performance of the method. Section 5 designs some the relevant comparative experiments, and analyzes the experimental results. Finally, it is the summary and outlook of this study.

2. Improved Skip List Algorithm

Skip list is an efficient data searching randomly structure. It is a multi-level sorted link list; this list is composed of the link list in order, as shown in Figure 1. The first floor contains the complete list sorted element; layer of each node is completely random. Each node in the skip list contains a plurality of pointers; pointers are used to find the corresponding data. Therefore, in addition to the first level of the link list, other levels of the list can be seen as a multi-level index. In general, the highest level of the list is called first level index. The numbers of pointers in each node are set up by the random algorithm when the skip list are established. The random algorithm is similar to the dropping coin experiment, as shown in table 3.1. If the right side of coin is shown (the result of random algorithm is odd), then go on the experiment until the reverse side of coin is shown (the result of random algorithm is even number. In the experiment, the times of dropping coin per node is the number of the level which contains the corresponding node. The level number of each node confirms to the geometric distribution of parameter P =1/2. The desired value of level is as follows: E[level] = 1/p = 2, that is to say, the average value of level number per node is 2.
The properties of the skip list are as follows:
1）The skip list consists of many levels;
2）Each level consists of orderly lists;
3）The bottom level contains all the information;
4）If one level contains the data, the next level will also contains the same data;
5）Each node contains several pointers (the number of pointers equal the number of levels)，one connects to the data which is at the same level, the other connects to the data which is at the next level.

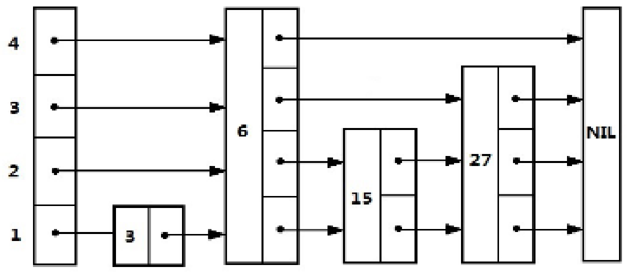


Figure 1. Skip List Structure Diagram

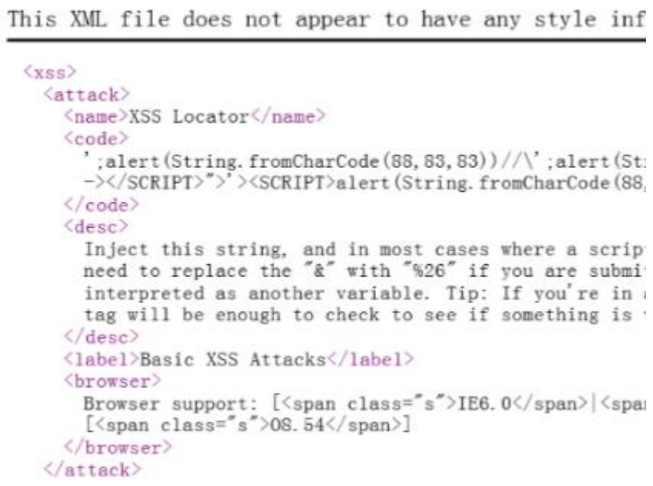
The advantages of the skip list are as follows:

1) It is easy to establish the skip list;
2) The efficiency is high，time complexity of all operations is O(logN).
3) The efficiency of locking mechanism for multithreading is high.

Skip list has all the advantages above, which can increase the searching efficiency of cross-site scripting attack detecting and reduce the wastage of system.

Skip list consists a series of nodes, which are connected together one after another. Each node has a pair of key / value and one or more pointers to the successor node. Typically, the number of pointer of each node is distributed randomly when the link list is established. However, during XSS attack feature vectors matching, the distribution probability of matching each attack vector should be taken into account. The study is initialized Skip list layers according to the probability distribution of each attack vectors.

The acquisition of probability is identified by the attacking feature vector description of HA.CKKERS library provided initially. In this XML file, <code> field is the main content feature of XSS attack vectors, <desc> is the description of the feature vector attack, <label> is the label of feature vectors attack.



```
This XML file does not appear to have any style inf

<xss>
  <attack>
    <name>XSS Locator</name>
    <code>
      ';alert(String.fromCharCode(88,83,83))//\';alert(St
      -></SCRIPT>">'><SCRIPT>alert(String.fromCharCode(88,
    </code>
    <desc>
      Inject this string, and in most cases where a scrip
      need to replace the "&" with "%26" if you are submi
      interpreted as another variable. Tip: If you're in
      tag will be enough to check to see if something is
    </desc>
    <label>Basic XSS Attacks</label>
    <browser>
      Browser support: [<span class="s">IE6.0</span>|<spa
      [<span class="s">O8.54</span>]
    </browser>
  </attack>
```

Figure 2. XML Attacking Feature Vector

According to the label and description, the occurrence probability of the attack feature vector can be identified initially, as shown in Table 1. Probability values can be adjusted by later experiments.

Table 1. Probability Statistics of Attack Feature Vector (Part)

Attack feature vector	Probability value
<SCRIPT>alert('XSS')</SCRIPT>	0.9
<BODY BACKGROUND="javascript:alert('XSS');">	0.8
<SCRIPT ="blah" SRC="http://ha.ckers.org/xss.js">	0.7

After determining the probability value of attacking feature vector preliminary, this study can build attacking vectors feature library by using Skip list level build algorithm based on the probability distribution.

As shown in Algorithm 1, it is the skip list level construction algorithm based on probability distribution. As described in the second row of Algorithm 1, it is sorted in descending order according to the probability of each node' size. The fourth row of the table is setting the max level of skip list to N. In 6-14 row, it will iterate the node set in proper order, and then, compared with the previous node; it will test whether their occurrence probability is the same as previous. If the probability is same, it will use the same level value; otherwise, it will decrease a bit. Therefore, the strategy of algorithm 1 shows that the same probability will get the same level value; in addition, the node that has higher probability will acquire a higher-level value.

```
Algorithm 1   Skip list level Construction algorithm based on probability distribution
// Probability for each eigenvector appear in descending order

NodeList = SortByNodeProbe();
// skip list level
Level = N+1;
PreProbe = 0;
For I = 1 : NodeList.length
      NowProbe = NodeList.get(I).probe;
If(PreProbe != NowProbe)
      Level=Level-1;
End
PreProbe = NowProbe;
// Sets the number of skip list level

NodeList.get(I).Level = Level;
End
```

3. The XSS Attack Defense Method based on Skip List

The main reason that XSS attack can succeed is imprecise input logic judgment of Web system. For example, Web system allows inputting and outputting the untrusted data. The main source of data is users' untrusted HTTP request data, such as URL parameters, form fields, Cookie, *etc*. In fact, the data also includes some invalidated data from the database, web server, or other sources and so on. Therefore, it should ensure that the untrusted data has no malicious attacks before the response to user data of Web applications.This paper proposes XSS attack protection model that is based on Skip list, as shown in Figure 3. The model can identify, either efficiently or not, whether untrusted data have malicious attacks.

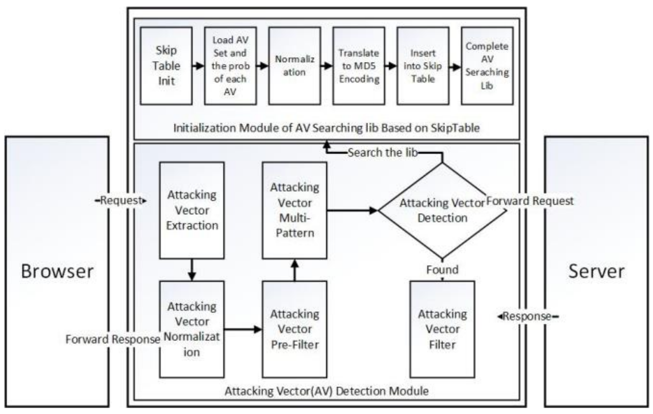

Skip Table Init
Load AV Set and the prob of each AV
Normalization
Translate to MD5 Encoding
Insert into Skip Table
Complete AV Seraching Lib
Initialization Module of AV Searching lib Based on SkipTable
Search the lib
Browser
Request
Attacking Vector Extraction
Attacking Vector Multi-Pattern
Attacking Vector Detection
Forward Request
Server
Found
Attacking Vector Normalization
Forward Response
Attacking Vector Pre-Filter
Attacking Vector Filter
Response
Attacking Vector(AV) Detection Module

Figure 3. Module Chart

In this paper, the characteristics of the attack vector library is provided by the HA. CKKERS. The basic flow of defense model is:

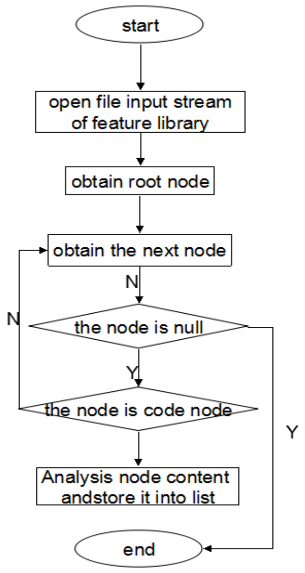


Figure 4. Parsing HA.CKKERS Feature Library

1) Creating an attack vector-matching module based on improved Skip list. This module is consisted of two steps which are parsing HA.CKKERS feature library and establishing cross-site scripting attacking feature library based on skip list. The Signatures attack vectors that are adopted in this paper is an XML-based description document. After the document is parsed, it will take MD5 hashing[11], then, create improved skip list structure like section 2 said according to its hash value sorting, and then, the attack vector information will be stored. Improved Skip list algorithm has a high search efficiency, time complexity is O (logN). In addition, due to the improvement of the level of construction strategy, it is beneficial for the vector matching of attack characteristics with different nodes probability.

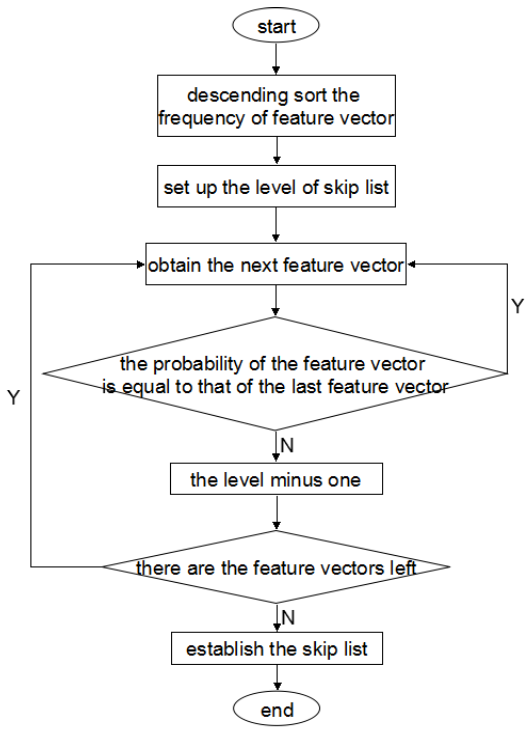

start
descending sort the frequency of feature vector
set up the level of skip list
obtain the next feature vector
the probability of the feature vector is equal to that of the last feature vector
Y
Y
N
the level minus one
there are the feature vectors left
N
establish the skip list
end

Figure 5. Establishing Skip List Feature Library

2) Detecting attacking vector. Generally, attacking script lies in the HTTP parameter. If a parameter list for HTTP requests is (p1, v1), ... , (pm, vm), (v1, v2, ..., vm) needs to be identified in the HTTP request.

Table 2. Related Character Encoding

Encoding name	Example
URL encoding	< → %3c
Unicode encoding	< → %u003c
HTMLencoding	< → &lt

3) Standardizing parameter values List of HTTP request. The XML description document do not use URL encoding, Unicode encoding and HTML entity encoding. As shown in Table 2, through using different form of encoding to the ' < ' character, the result is different. In order to identify the hidden attack code segment, each parameter of the (v1, v2, ... , vm) needs to be decoded, and then each possible code script need to be identified, as shown in Figure 6.

4) Multi-mode technology. In order to perceive the uncertainty of attack script, the method of multi-mode feature vector selection is utilized in this paper. In order to increase the accuracy of detection, this paper adopts the longest common subsequence multi-pattern matching algorithm. This paper sets L as the minimum length of a common subsequence to improve the matching efficiency.

As for multi-mode screening, the length of common sub-sequence is set to L and utilize DFA[13] to conduct sub-sequence firstly, then, we can obtain one sub-sequence to decide whether it is the malicious sequence using skip list.

If the sequence is malicious, then the procedure will be finished. If the sequence is not malicious, then we can judge whether there are sub-sequences and judge whether the sub-sequences are malicious until there is no sub-sequence left. The process is shown in Figure 7.

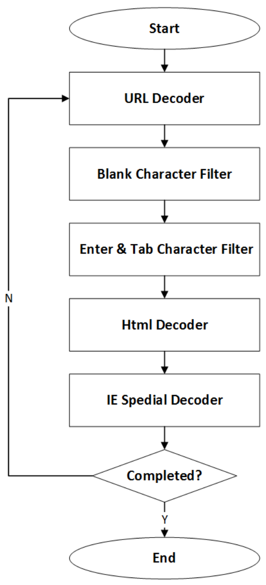


Figure 6. The Script Decoding Process

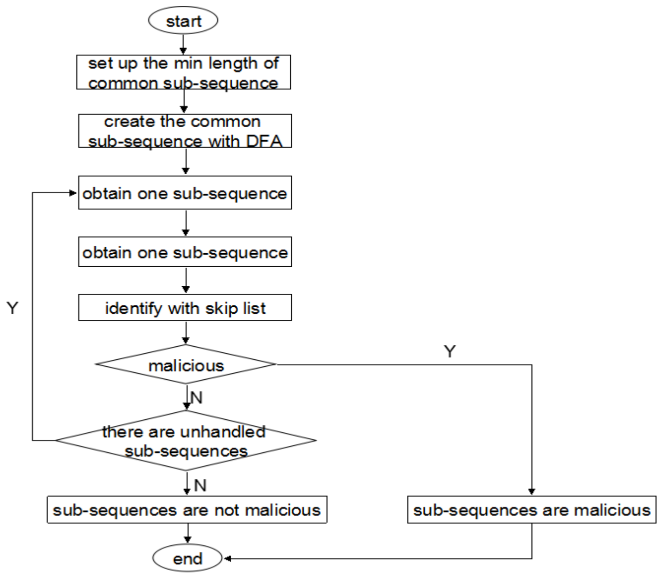

start
set up the min length of common sub-sequence
create the common sub-sequence with DFA
obtain one sub-sequence
obtain one sub-sequence
identify with skip list
malicious
Y
N
there are unhandled sub-sequences
Y
N
sub-sequences are not malicious
sub-sequences are malicious
end

Figure 7. Multi-Mode Screening

5) After handling the attack script set in step 4, it uses 16-bit MD5 code to generate a 64-bit binary representation of an integer, and then it employs skip list generated in step 1 for matching.

4. Performance Improvement of Skip List-Based Cross-Site Scripting Attack Prevention Method

As above-mentioned, the greatest performance bottleneck of skip list-based cross-site scripting attack prevention method is in step 4, because DFA could produce large amounts of subsequence. Although it adopts the length limitation of the littlest subsequence with L in step 4, its performance bottleneck phenomenon is still obvious. To further improve the overall performance of cross-site scripting attack protection method, this study will utilize a simple pattern matching method to conduct a pre-screening. The process is shown in Figure 8.

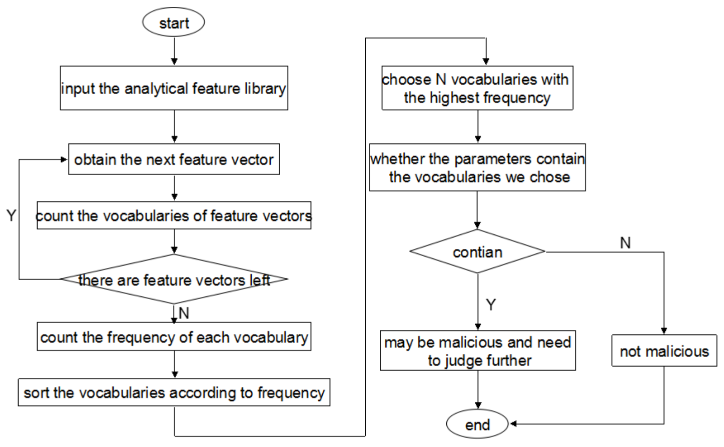

start
input the analytical feature library
obtain the next feature vector
count the vocabularies of feature vectors
there are feature vectors left
Y
N
count the frequency of each vocabulary
sort the vocabularies according to frequency
choose N vocabularies with the highest frequency
whether the parameters contain the vocabularies we chose
contian
Y
N
may be malicious and need to judge further
not malicious
end

Figure 8. Pre-Screening

Because the number of the attacking vector feature library supplied by HA.CKKERS is not very big, this study can count the main vocabulary of it. For example, there would be higher proportion of XSS attack when some vocabularies (*e.g.* document、javascript、alert) appear. If the attacking string does not contain some vocabularies (*e.g.* document、javascript、alert), we can exclude it as a possible malicious script code. After the pre-screening, if we could not exclude it as a possible malicious script code, we can adopt the step 4 for judgment.

To pre-screen, we should obtain the next feature vector according to the feature library which is parsed and count the vocabularies according to the feature vector at first. Then we should judge whether the unhandled sequences are left， we should obtain the sequences until all of them be handled. If all the feature vectors are handled, we will count the frequency of each vocabulary and sort the vocabularies according to the frequency. We can choose N vocabularies which have the highest frequency according to the sorting result; then, we can judge whether the parameters contain the vocabularies we choose. If the parameters contain the vocabularies we choose, then the vocabularies may be malicious and need to be judged further. If the parameters do not contain the vocabularies we choose, then the vocabularies are not malicious.

As shown in algorithm 2, it is the skip list-based cross-site scripting vulnerability defense method. In line 2, it creates the skip list for attack vectors matching. In line 4, it can get the vocabulary form library from the attacking vector feature library supplied by the analytical HA.CKKERS, which can be used for the pre-screening of the attacking vector. In line 6, it gets a HTTP request, which can analyze its parameter list of GET/POST method respectively. In line 8, it normalizes the list of HTTP request parameter value, which can prevent the attack code from using different coding bypass recognition process. In line 10, it adopts the vocabulary list of feature vector library to conduct pre-screening for the attacking vector. From line 12 to 21, it recognizes the attacking vector that failed to exclude suspected. From line 14 to 15, it produces all the public subsequence whose length is more than L using DFA. From line 16 to 20, it can deal with the entire attacking vector with multi model, and after adopting MD5 hash for all vectors, which will have a matching in the skip list.

```
Algorithm 2   Skip list-based cross-site scripting vulnerability defense
method
// Adopt algorithm 1   conduct the skip list construction
Skip list = CreateTheSkip list();

// Obtain the vocabulary list of   attack characteristic vector library
Voc = GetVocFormLib();

// Capture an HTTP request, and get its argument list
ParamList = GetParametersFromHttpRequest();

// Standardized HTTP   Request the parameter values list
NormalParamList = NormalHttpParamList(ParamList);

// pre-screening
Flag = PreScreen(NormalParamList);

// not pass pre-screening
If(Flag)

        // Multi-mode of HTTP   Request the parameter values list
MultiModelParamList =
                MultiModelParamList(NormalParamList);
For I = 1 : MultiModelParamList.length
               ScreenBySkip list(
Skip list，
MultiModelParamList.get(I));
End
End
```

Algorithm 2 adopts the improved skip list initialization algorithm, when conducting the cross-site attacking feature vector matching; the probability distribution of matching each attack vectors has been taken into consideration. Because of the skip list to store, it is simple and easy to use. In addition, it has higher search efficiency. In addition, it can count vocabulary list of the feature vector library, and it could conduct the pre-screening for the attacking vector, which makes the performance improved effectively. For the attacking vectors that could not be excluded suspicion with pre-screening, this study employs multi-mode screening of DFA method, which can improve the screening accuracy effectively, and can reduce the false alarm rate.

5. Experiment Result

Most of the web services use JSP technology and server forwarding technology, so they use JAVA encoding. Cross-site scripting attack detection method based on skip list in this article uses JAVA encoding. The system deploying structure is shown in Figure 9.

In order to describe the cross-site scripting attack detection method based on skip list and the ability to detect cross-site scripting attack sample in this article, this study use the cross-site scripting attack sample provided by XSSed site to test this method. The XSSed site provides a large number of test samples, some of which are solved. And some of the test samples were proposed not long ago[3]. In order to test the cross-site scripting attack detection method based on skip list more accurate, this study will collect 1000 cases with POST method and GET method individually on the XSSed website, and also collect 1000 cases of normal testing samples. As a comparative experiment, it will employ IISUTM for comparison.

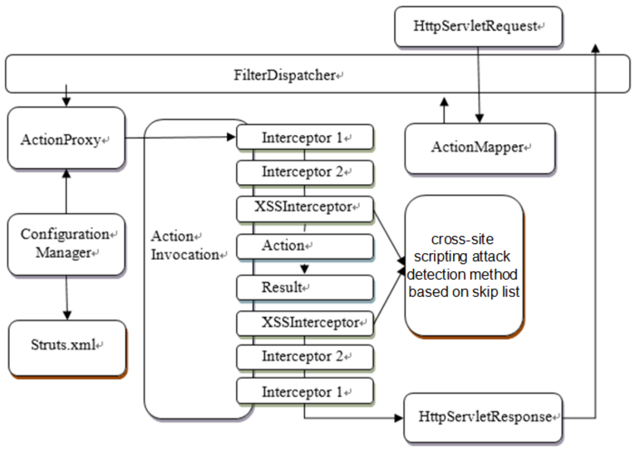


Figure 9. System Deploying Structure

Testing result is as shown in Table 3. After analysis of the experimental result, in the POST method, we find that there are 880 cross-site scripting attacks in the testing method of this study, while the NO.of IISUTM testing method is 869.Thus,we can see that the method of this study can test 11 more attacks, and the testing accuracy increases 11% than IISUTM method. In the GET method, we find that there are 887 cross-site scripting attacks in the testing method of this study, while the NO.of IISUTM testing method is 873.So we can see that the method of this study can test 14 more attacks, and the testing accuracy increases 14% than IISUTM method. In the false alarm testing of normal sample, the false alarm rate of testing method in this study is 21.3% lower than the IISUTM testing method.

Table 3. Comparison of Testing Result

Category	IISUTM	Testing method in this study
NO. of POST testing method	869	880
NO. of GET testing method	873	887
False alarm of normal sample	178	140

As shown in Figure 10, the result indicates that, compared to IISUTM, the new method that based on skip list cross-site scripting attacks has higher testing accuracy and lower false alarm rate.

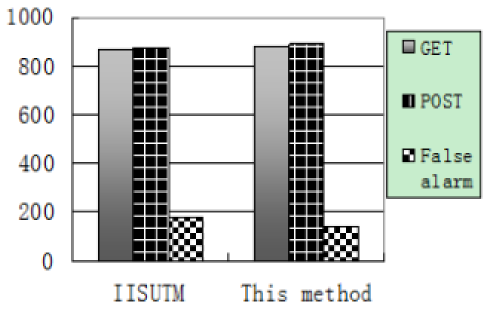


Figure 10. Accuracy Histogram

In order to illustrate the performance of new method that skip list-based cross-site scripting attacks, this research adopts contrast experiment. The first experiment employs the testing method of cross-site scripting based on general skip list, while the second experiment utilizes the improved testing method. In addition, under different scale, this research would have a comparison of testing time between the first and second experiment.

As shown in Table 4, when the number of requests during thousand unit level, the performance of the method is improved 60 percent; in the ten thousand unit level, performance is improved 82%, which has the most significant effect; in the one hundred thousand unit level, the improved rate is 41%, which is declined of the improving performance; in the million unit level, the performance improvement is 20%, although performance has improved, it is of smaller enhance rate. We can use Line chart to analyze the improving performance, as shown in Figure 11.

Table 4. Comparison of Detection Efficiency

Request Number	Experiment 1 （ms）	Experiment 2 （ms）	Promotion (%)
1000	10	4	60
10，000	45	8	82
100，000	81	47	41
1000，000	485	390	20

As shown in Figure 11, under the small scale, the second experiment, which adopts the improved testing method, has significant improvement of time performance than the first experiment that employs the general testing method. When the request size increases, compared with the first experiment adopting the general testing method, the second experiment using improved testing method has relatively slower improving performance. Therefore, when requesting for large-scale concurrent, it should utilize the distributed WEB services to improve performance.

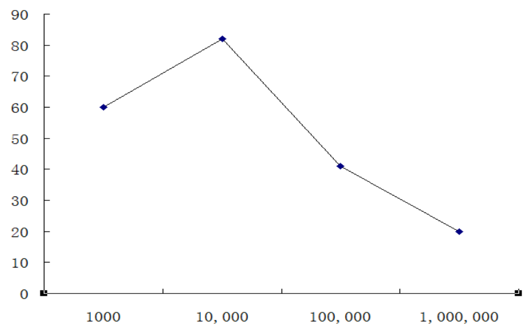


Figure 11. Time Performance Line Chart

6. Conclusion

This study proposes a new cross-site scripting attacks detection method, which is based on an improved skip list level build algorithm to build the vector library. The times of matching feature library are no more than 8(that is to say, log113, and 113 here is the number of cross-site attacking feature vector in HA.CKKERS feature library) because of the capacity of the feature library and the features of improved skip list algorithm. Establishing similar vectors with help of deterministic finite state machine algorithm to compare with attack vector of the multiple mode to improve the quality of detection method. The whole detection method use some steps, which are establishing skip list feature library, regulating script, using MD5 encoding and converting long integer, pre-screening attack vectors, judging attack vectors, to make up entire modular cross-site scripting attack detection framework[14]. We use this framework and the existing general XSS detection methods for comparative experiment. The experiment shows that the accuracy of the method in this study is higher than that of ordinary defensing method when detecting the cross-site scripting attack. What's more, the detection method in this study shows better function and performance in the experiment. The principle of the detection method is simple. It is easy to realize this method, so the method is better at detecting the cross-site scripting attacks.

The studies provide the improved skip list level build algorithm according the features of the skip memory. The method is aimed at mission success rate and time performance of cross-site scripting attacking. A finite state machine algorithm is used to generate a similar attack vector so that we can match cross-site scripting attacking vectors with multi-pattern. Using the method for pre-screening of feature characters to increase the performance of detection method and achieve the modular cross-site scripting attack detection framework. The advantages of this technology are reflected in the use of the theory of skip list and the improvement of the skip list so that the number of matching are no more than logN when searching and matching the skip list feature library. Moreover, we can filter out the string which do not contain the feature vocabularies, so the detecting performance will be increased when counting the feature vocabularies in feature library. Owing to the various forms and a large number of cross-site scripting attack vectors, we must collect comprehensive feature library to increase the accuracy of detecting. We must use distributed technology[15] to increase the performance of cross-site scripting attack, which is our target in the future.

Acknowledgement

This work was supported by the Key Project of National Defense Basic Research Program of China(Grant No. B1120132031).

References

[1] I. P. Salas M and Martins E, Electronic Notes in Theoretical Computer Science, vol. 302, no. 133, **(2014)**.
[2] Category: OWASP Top Ten 2013 Project [EB/OL]. [2013-06-23]. https://www.owasp.org/index.php/Top_10_2013-Table_of_Contents.php/Top_10.
[3] XSSED. XSS Attack Information [EB/OL]. [2015-04-08]. http://xssed.com.
[4] Y. Nadji, P. Saxena and D. Song, Editors, "Document Structure Integrity: A Robust Basis for Cross-sitescripting Defense", Proceedings of the 16th Network and Distributed System Security Symposium, San Diego, USA, **(2009)**.
[5] X. Wang and Y. Zhang, Chinese Academy of Sciences Univ., vol.5, no.668, **(2011)**.
[6] M.V. Gundy and H. Chen, Editors, "Noncespaces: Using Randomization to Enforce Information Flow Tracking and Thwart CrosssiteScripting Attacks", Proceedings of the 16th Network and Distributed System Security Symposium, San Diego, USA, **(2009)**.
[7] X. Zhao and Y. Tang, "Information Network Security", vol. 11, no. 62, **(2011)**, Network Security Technology and Application, vol.4, no.14, **(2013)**.
[8] W. Cao, F. Guo and M. Yu, Computer Engineering, vol.6, no.154, **(2013)**.
[9] E.N. Hanson and T. Johnson, Lecture Notes in Computer Science, vol. 10, no.153, **(1991)**.
[10] I. Hydara, A.B.M. Sultan, H. Zulzalil and N. Admodisastro, Information & Software Technology, vol. 58, no. 170, **(2015)** .
[11] J.D. Touch, Acm Sigcomm Computer Communication Review, San Diego, USA, vol.77, **(1995)**.
[12] M. Becchi and P. Crowley, Acm Transactions on Architecture & Code Optimization, vol. 1, no. 152, **(2013)**.
[13] S.Yacoub and H. Ammar, Pattern Languages of Program Design, vol. 8, no.19, **(2000)**.
[14] J. Bozic and F. Wotawa F,Editors, "XSS pattern for attack modeling in testing. Automation of Software Test (AST)", 2013 8th International Workshop on IEEEB, San Francisco, USA, **(2013)**.
[15] J.J. Davis and A.J. Clark, Computers & Security, vol. 30, no. 353, **(2011)**.
[16] X. Liu, M. Sun and L. Tang, Journal of Harbin University of Science and Technology, vol. 2, no.42, **(2010)**.