ISSN: 2460 – 7223

Method for Detection and Mitigation Cross Site Scripting Attack on Multi-Websites

1st Hartono
Master of Information Technology Department
Institute of Technology and Business Darmajaya
Lampung, Indonesia
harton.2021210011@mail.darmajaya.ac.id

2nd Joko Triloka
Master of Information Technology Department
Institute of Technology and Business Darmajaya
Lampung, Indonesia
joko.triloka@darmajaya.ac.id

***Abstract*—Cross-Site Scripting (XSS) attack exploits scripting security bugs and issues on the website. XSS attack focuses and occurred on client browser application or frontend. It consists of three types of attacks: stored, reflected, and document object manipulation. The XSS attacks can cause fatal and dangerous problems, such as theft of user data, account takeovers, and illegal access to banking transactions or important data. Studies on XSS detection and mitigation have been carried out by some researchers, but it still leaves some problems, such as there is no connected mitigation to respond to the attack, using only a single-layer security mechanism and fewer payload data to test, weak measurement of the defense effectiveness from XSS attack, and the use of insufficient experiment and data testing. In addition, the method used in previous research still fails to solve all types of XSS attack. Most of the previous research also separates the method of attack detection and its mitigation. Therefore, this study proposes not only for detection but also for mitigation to overcome XSS attacks. The proposed method in this study is divided into two parts: detection and mitigation method. The proposed detection method is by using machine learning, based on lexical analysis. Then, the proposed mitigation method is the multi-layer security method which consists of five layers of the security. The proposed method has been structured systemati-cally and procedurally. In previous research, the partial methods proposed in this paper has been effectively implemented. There-fore, the proposed method is regarded as appropriate method to detect and mitigate XSS attack.**

***Keywords*—XSS, cross site scripting, mitigation system, machine learning, cyber-attack, lexical analysis**

I. INTRODUCTION

Cross-Site Scripting (XSS) attacks can cause very serious problems. Based on some cases that previously occurred, XSS attacks can be used to do data theft, account takeover, manipulate users' decisions, or become pre-initial attack to do further attacks [1]. A big company like eBay and Amazon for example has been recorded to have experienced an XSS attack. This method exploits a security vulnerability on the scripting or sanitation side [2]. In other words, XSS will be executed through the client's browser. XSS security vulnerabilities occur when applications or software do not sanitize or validate input, variables, or parameters properly. That vulnerability allows attackers to send JavaScript code using browser requests. To send the malicious code, the attacker can send the code using forms, URL, or document object manipulation (DOM).

XSS attack is not actually the new method of cyber-attack. This type of attack has long been discovered, and not as new term in cyber security [2]. However, in fact, based on Open Web Application Security Project (OWASP) Top-10 Web Vulnerabilities 2017, XSS becomes one of vulnerability which is often found on most of cyber-attack in the world [3]. In addition, XSS vulnerabilities also still found OWASP Top-10 Web Vulnerabilities 2021 (statistics-based proposal) [4]. The appearance of XSS attacks on the data certainly shows that XSS attacks occur and develop consistently. XSS security vulnerabilities are still considered a common problem, so attackers can take advantage of this attack to carry out further attacks. In fact, as already explained, this XSS attack can cause serious problems. Therefore, it is important to formulate and develop more effective method to overcome this type of attack. The detail of OWASP Top-10 web vulnerabilities could be seen on table 1.

TABLE 1. OWASP TOP-10 WEB VULNERABILITIES 2017 AND 2021

OWASP Top-10 2017		OWASP Top-10 2021 Proposal	
A1	Injections	A1	Injections
A2	Broken Authentication	A2	Broken Authentication
A3	Sensitive Data Exposure	A3	Cross-Site Scripting (XSS)
A4	XML External Entities (XXE)	A4	Sensitive Data Exposure
A5	Broken Access Control	A5	Insecure Deserialization
A6	Security Misconfiguration	A6	Broken Access Control
A7	Cross-Site Scripting (XSS)	A7	Insufficient Logging & Monitoring
A8	Insecure Deserialization	A8	Server-Side Request Forgery (SSRF)
A9	Known Vulnerabilities	A9	Known Vulnerabilities
A10	Insufficient Log & Monitoring	A10	Security Misconfiguration

Based on data from the Indonesia National Cyber and Crypto Agency (usually called BSSN), there were around 12.9 million cyber threat attempts to Indonesia during 2018. A total of 513,900 of the total attacks were malware. Not only that, during January - April 2020, BSSN recorded that there were around 88,414,296 cyber-attack activities in Indonesia [5]. With that high attack statistic level, each attack used very complex and varied methods and techniques. Unfortunately, no single Indonesian institution has complete data about the attack description. Therefore, it is not easy to give exact prediction about what methods used by attackers. To detect and mitigate the attack and in relation to the XSS attack, some research data published by cyber-security company and researchers may be used as represented reference to predict the attackers' methods.

As mentioned, XSS attack has appeared in OWASP Top-10 web vulnerabilities [6]. This type of attack become one of common cyber-attack method used by several attackers. In addition, international cyber-security companies, Rapid7 and Netwrix also put XSS attack method as one of top most common types of cyber-attack [7-8]. They also explain several impacts when attacker exploit XSS vulnerability. In this case, since XSS mentioned as top-most common types of cyber-attack and web vulnerabilities, it is indisputable fact that XSS attack should get attention and cannot be underestimated. Therefore, based on those facts, detection and mitigation method or mechanism become an important thing to continuously developed and studied.

II. PREVIOUS RESEARCH

Several research have discussed XSS attack detection and mitigation method. In general, the study of XSS attacks discusses how to develop accurate attack detection and take preventive actions against XSS attacks. Most recent studies that discuss XSS detection are still use limited sample of XSS attack datasets and have lack procedure in testing the effectiveness of detection. Those studies did not use powerful attack application to test whether the detection was successful implemented or not. In addition to detection, the study also discusses how to defend or prevent XSS attacks. Research [9] used machine learning with hybrid features to detect XSS attack. The use of hybrid features in XSS detection is quite accurate, but there is no mitigation mechanism explained and developed. Still related to the machine learning based detection method, research [10] explore artificial intelligence (AI) term to detect XSS attack. The term is multilayer perceptron technique. The same as previous research, it was only focused on how to build or develop accurate XSS detection. Using multi-layer perceptron to detect XSS attack pattern is not easy to implement, it may need more or high computer resource only for detecting XSS attack. In addition, it was not also easy to embed the detection engine in web application architecture.

Most studies on the prevention of XSS attacks have not been carried out comprehensively. The experiment done by research [11] for example, the defensive method was implemented in very simple web application with 8 pages only. The similar case also occurred on research [12], XSS attack simulation or testing is only based on single pieces of JavaScript code so the result of experiment may not be reliable and representative. The data used are also limited, taken from small scope of cases. It is related to the research [13]. This research use Code Igniter XSS filtering library to filter or sanitize user requests. Unfortunately, there was still no sufficient explanation or measurement about the effectiveness of XSS filtering in solving all types of XSS attack: stored, reflected, and DOM. Similar to previous researches, the research [14] only focus on how to detect XSS attack by using OWASP Security Shepherd. Mitigation and defensive system were not implemented yet.

Based on several previous research explained, there are still five common weakness and problem found, they are:

- the use of limited data in detecting XSS attack.
- weak implementation of XSS attack simulation.
- lack of use attack application or technique to simulate XSS attack to the proposed or developed method.
- the use of very common JavaScript code to examine the effectiveness detection method, so the level of effectiveness is not easy to be concluded.
- unintegrated detection and mitigation method.
- unable to protect multi-websites.
- there is no detailed specification of target website, such as description about security level, provided vulnerabilities, other web vulnerabilities interference.
- the use of single layer security technique.
- not all XSS attack types are solved.

III. CROSS SITE SCRIPTING

XSS attacks exploit the user's browser or frontend. To perform this attack, the attacker will use JavaScript code which run on client browser. If the website has an XSS vulnerability, the JavaScript code can be executed, against the business process of the application [15]. The code can be submitted via search box, form value, and DOM. To check whether the website has XSS vulnerability or not, the attacker simply does some experiments or trial and error. When the test XSS attack is successful, the attacker will devise a scenario to carry out further attacks. The motivation for the attack will certainly vary, depending on the attacker's motives. In more advanced method and techniques, XSS vulnerabilities can be scanned by certain software.

XSS attack has three types of methods, they are (1) stored XSS, (2) reflected XSS, and (3) Document Object Manipulation or DOM based XSS [16]. The essential difference between these types is in how attacker implement the attack procedures and technique. In case of real attack implementation, although there are only three types of XSS attack, there are many techniques that can be used to exploit XSS vulnerabilities. That's why it is urgent and important to develop detection system to detect varied XSS attack pattern accurately.

Stored XSS occurs when the attacker sends and saves malicious JavaScript code to the database, file, or filename. The system saves the code because of a lack of sanitation and validation. When the user or victim accesses the page containing the code, the browser will execute XSS code. In contrast to the stored, reflected XSS can be executed without saving the code into the database. It acts like a mirror or reflection. Most type of this XSS attack occurs in the search box, filter widgets, or URL. Reflected XSS is more frequently used by attackers because they can see the result faster than stored XSS. DOM XSS can be implemented by modifying DOM in the victim's browser. DOM XSS is more difficult to detect, but also difficult to implement. To see the clear differences between each type of XSS attack, see the following XSS code example in table 2.

TABLE 2. THE DIFFERENCES BETWEEN TYPE OF XSS ATTACK

	XSS Type	Code Explanation and Example
1	Stored XSS	The malicious XSS code is saved on the database and executed when user or victim access the page contained that saved malicious code. ——————————————— <!DOCTYPE html> <html lang="en"> <head> <title></title> <link rel="stylesheet" href="css/main.css" /> </head> <body> **<p><script> malicious XSS code </script></p>** </body> </html>
2	Reflected XSS	Most of search box or page will show the search keyword on the result page. The attacker can use the search box to send malicious XSS code. ——————————————— Sample of malicious XSS code: **<script>alert(document.cookie)</script>** ——————————————— When the keyword submitted, the URL will be the following pattern: http://www.victim.site/search.php?keyword**=<script>alert(document.cookie)</script>**
3	DOM XSS	This type of XSS can be done by modifying DOM element in a web page. Modifying URL parameter on navigation menu for example. http://www.victim.site/page.html?default**=<script>alert(document.cookie)</script>**

To describe why XSS can be one of dangerous attack method and how this attack method is implemented, it is needed to explain simple XSS attack scenario. This is how attacker do account takeover using stored XSS technique. Before explaining the scenario, it is important to know that this is very simple and common scenario of XSS attack procedure. Those scenarios are (1) attacker send XSS malicious code and save it to the victims website, (2) the malicious code will redirect user or web visitor to the phishing page prepared by attacker, (3) the phishing is a login page and the victim is asked to login to the page, (4) because the attacker makes the page similar to original one, the victim login to the phishing page, (5) attacker steal the credential or account of victim (username/e-mail and password), (6) attacker login to the original page using victim's account and change the password and other identification, and (7) attacker uses victim authority to do further illegal actions. See figure 1 to check overall scenario.

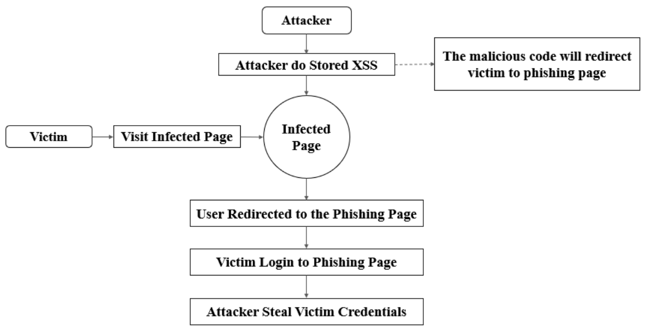

Attacker
Attacker do Stored XSS
The malicious code will redirect victim to phishing page
Victim
Visit Infected Page
Infected Page
User Redirected to the Phishing Page
Victim Login to Phishing Page
Attacker Steal Victim Credentials

Figure 1. Simple XSS Attack Scenario in Account Takeover

IV. METHODS

The method proposed in this study is divided into 2 parts. First, the machine learning method to detect XSS attacks. To detect more accurately, the detection uses machine learning model. The second proposed method is a multi-layer security to mitigate or prevent XSS attack. Therefore, to ease the explanation, the proposed method will be divided into two separated explanations. To get an idea of the overall method proposed in this paper, see Figure 2.

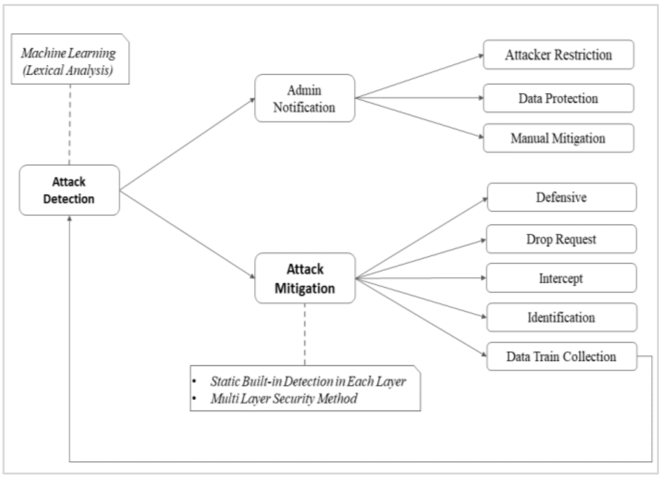


Figure 2. XSS Attack Detection and Mitigation Scenario

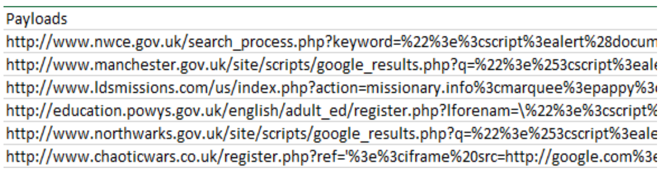
Both detection and mitigation method have mutualistic relationship. When the detection engine detects XSS attack, the system can notify the web administrator and activate or trigger the mitigation engine. In addition to build wall system defense, mitigation engine collects XSS attack pattern to NoSQL database (JSON) as feedback. The detection engine can use the database for optimizing further detection result. Meanwhile, in case of web administrators' action, they might take certain action to avoid the attack, such as restrict attacker IP or Mac Address (if identified), take data protection acts, or carry out manual mitigation. With these two forms of notification, the web application gets more option to customize security level and action.

A. *Machine Learning Detection*

The proposed method used to detect XSS attack in this study is machine learning model. XSS attack patterns are string-based attacks packaged in the form of JavaScript code. The composition string of XSS code can be written very complex and varied, so the application or system must have a comprehensive attack code detection method. It is a must to have complete dataset to develop comprehensive detection. The use of machine learning is to ease the detection of complex attack string pattern. In addition, the detection also can be carried out independently, without involving multiple administrator actions. Therefore, the detection can be carried out more effectively.

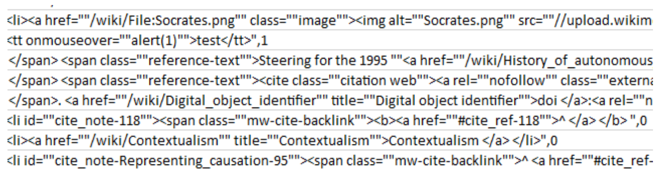
In case of machine learning implementation, XSS dataset can be taken from GitHub and Kaggle. Figures 2 and 3 show two datasets that have different scope or source. XSS code in Github dataset taken from URL or request parameters, and Kaggle taken from HTML source. To get more accurate machine learning model, the dataset is separated into two datasets, for training and testing purpose. To ease the process of detection implementation, Turicreate can be used as engine. This library can ease the implementation of detection using machine learning. Turicreate has complete and powerful machine learning or even AI features. Related to XSS pattern detection, it has a text classifier feature. The feature can be used to classify the text or XXS codes into recognized pattern with the use of lexical analysis. Thus, it is assumed as an effective engine to build the method of XSS attack code detection [17]. This Apple machine learning engine can do lexical analysis or computation so the detection method can accurately predict whether the query or code is malicious or not.

Meanwhile, to check the effectiveness and the stability of the detection method, Zap or OWASP Zed Attack Proxy application could be used because it has XSS attack features. Zap is used to send the payload, so the machine can test and learn the attack string pattern used during the payload. To get more description about dataset, see figure 3 and 4 to see dataset sample.



```
Payloads
http://www.nwce.gov.uk/search_process.php?keyword=%22%3e%3cscript%3ealert%28docum
http://www.manchester.gov.uk/site/scripts/google_results.php?q=%22%3e%253cscript%3eal
http://www.ldsmissions.com/us/index.php?action=missionary.info%3cmarquee%3epappy%3
http://education.powys.gov.uk/english/adult_ed/register.php?lforenam=\%22%3e%3cscript
http://www.northwarks.gov.uk/site/scripts/google_results.php?q=%22%3e%253cscript%3eale
http://www.chaoticwars.co.uk/register.php?ref='%3e%3ciframe%20src=http://google.com%3
```

Figure 3. XSS Dataset From Requests Parameters (Github)



```
<li><a href=""/wiki/File:Socrates.png"" class=""image""><img alt=""Socrates.png"" src=""//upload.wikim
<tt onmouseover=""alert(1)"">test</tt>",1
</span> <span class=""reference-text"">Steering for the 1995 ""<a href=""/wiki/History_of_autonomous
</span> <span class=""reference-text""><cite class=""citation web""><a rel=""nofollow"" class=""extern
</span>. <a href=""/wiki/Digital_object_identifier"" title=""Digital object identifier"">doi </a>:<a rel=""n
<li id=""cite_note-118""><span class=""mw-cite-backlink""><b><a href=""#cite_ref-118"">^ </a> </b> ",0
<li><a href=""/wiki/Contextualism"" title=""Contextualism"">Contextualism </a> </li>",0
<li id=""cite_note-Representing_causation-95""><span class=""mw-cite-backlink"">^ <a href=""#cite_ref-
```

Figure 4. XSS Dataset From HTML/Page Source (Kaggle)

The establishment procedures of XSS detection in this study are divided into three stages: machine learning preparation, detection integration, and detection implementation. The machine learning preparation has seven steps as follows: (1) data preparation; (2) data pre-processing; (3) data modeling; (4) data training (5) data testing; (6) performance evaluation; and (7) performance optimization. To optimize the performance, n-gram value set to the text analytic. It is done to see the accuracy level based on n-gram value. N-gram is a term in Natural Language Processing (NLP). It can be defined as continuous sequences of words, symbols, tokens in a sentence or document. In more technical terms, n-gram can be also assumed as neighboring sequences of items in a sentence. When optimized n-gram has been formulated, model can detect the XSS code patterns, and the interpretation can be the table

TABLE 3. XSS PATTERN INTERPRETATION BASED ON N-GRAM

XSS Pattern	n-gram 1	n-gram 2	n-gram 3
Pattern 1	False	True	True
Pattern 2	True	True	False
Pattern 3	True	True	True
Pattern 4	True	True	False
Pattern 5	True	True	True

True means the detection is valid, and false is invalid. It means that n-gram value can affect to the accuracy of detection. In addition, there are some procedures taken to get more accurate result, such as separating data modelling into training and testing, and evaluating performance of machine learning model. It is also important to use a comprehensive XSS dataset with very complete data features, so the detection engine can be more accurate. Related to machine learning detection procedures, see figure 5.

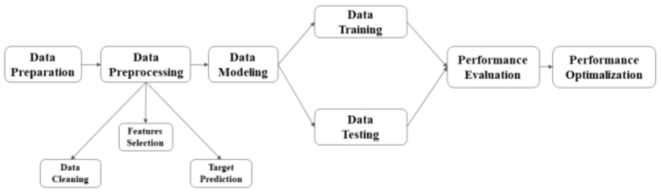


Figure 5. Machine Learning Detection Procedures

As shown on the figure 5, data processing has three subprocess, as follows data cleaning, features selection, and target description. After machine learning is ready to be implemented in real payload, the detection and mitigation methods need to be integrated. In this case, the detection become a trigger to create an admin notification and activate the mitigation. The detection engine do prediction XSS cyber threat [18], and the mitigation engine build secured defense system. To clearly get description, see the figure 6.

To develop integrated and synchronized detection and multi-layer security engine, the detection engine can be embedded in web application, as a service or task. In addition, it can also be placed on web server mod security or HTTP layer. It is also important to know that each security layer has built-in attack detection, although it has not built with machine learning mechanism. See figure 6 below to check the integration of detection and mitigation mechanism.

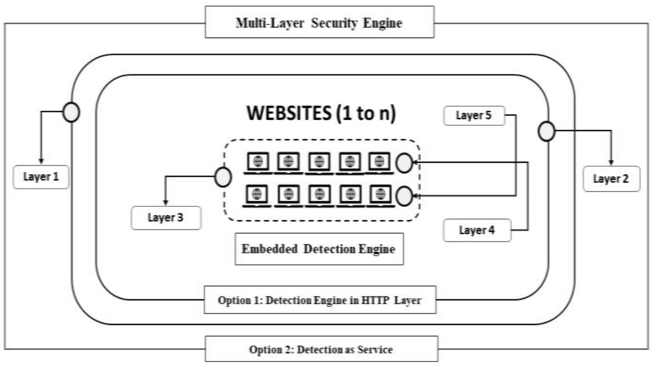

Multi-Layer Security Engine
WEBSITES (1 to n)
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Embedded Detection Engine
Option 1: Detection Engine in HTTP Layer
Option 2: Detection as Service

Figure 6. Integration of Detection and Mitigation Mechanism

After machine learning model is embedded as detection engine, Zap and Arachni is implemented to establish XSS attack simulation to sample websites run on web server. Those application have very comprehensive XSS attack features so the pentester, researcher, or web administrator can evaluate web security level, especially for evaluating XSS attack pattern. Those applications can also run multi-threaded tasking and service, so the process of XSS attack can be rapidly fast and powerful. Payload attack implemented by Zap and Arachni can produce detailed attack statistic or report. Since the mechanism is supported by attack report, it can be used to compare security level effectiveness between the propose and previous methods. In addition, it can also be used as proof that the proposed method has accurate detection and reliable defensive system that can solve security problems found. It is also to measure the effectiveness level of proposed method. See figure 7 to see the implementation of XSS attack using Zap and Arachni.

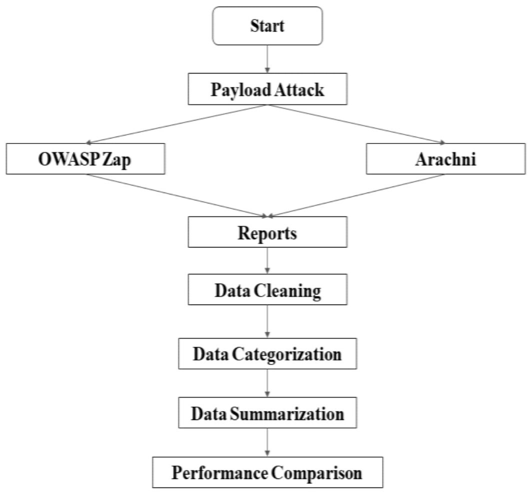


Figure 7. Attack Simulation Detection and Mitigation Zap and Arachni

B. Lexical Analysis or Computation Technique

In context of this study, lexical analysis is the method to extract and classify sentences into structured-identified value [19]. This analysis technique is used to ensure whether the input (payload) is XSS malicious code or not. Therefore, sentences in lexical analysis assumed as codes or URL used by attacker to insert XSS code. Lexical analysis is used to extract features from XSS dataset. This type of analysis extracts XSS codes into several parameters or characteristics. The text will be extracted as ASCI so the machine learning engine can classify whether the string is XSS codes or not. In case of detecting XSS code pattern, lexical analysis is simple but powerful method. With this analysis model, the process of detecting XSS malicious code can be easily done, without consuming many computation resources. Since the analysis model only take a few resources, it can be embedded in several layers. As seen on figure 8, the XSS attack detection will use lexical analysis.

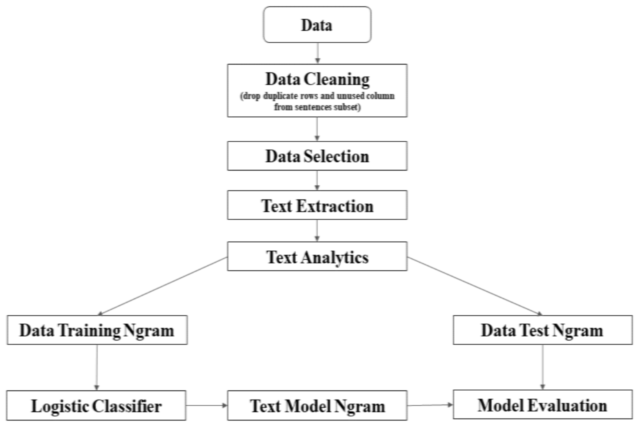


Figure 8. Lexical Analysis for Feature Extraction and Analysis

C. Multi-Layer Security for Mitigation

Multi-layer security is a mechanism to comprehensively secure a website or application. This security is called comprehensive because it can handle all websites that are on a web server. This security is built to meet the various characteristics and security needs of each level of website security. This mechanism can also cover various web architectures if it is on the same web server. This mechanism is compiled from various research results on cyber security. Multi-layer security consists of five security layers, they are (1) OWASP ModSecurity; (2) Framework/CMS Default Security Features; (3) HTTP Middleware; (4) Templating Engine; and (5) Data Sanitizer or filter. See figure 9 to get more clearly description.

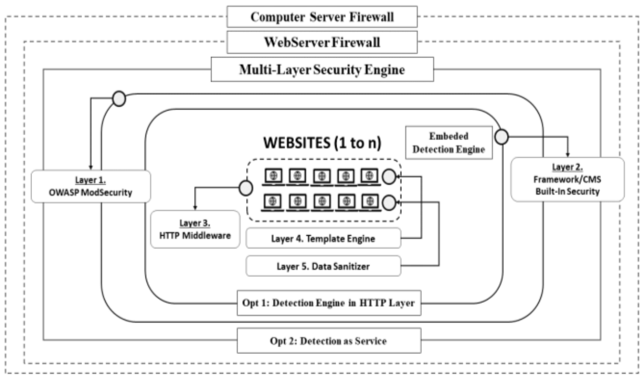

Computer Server Firewall
WebServer Firewall
Multi-Layer Security Engine
WEBSITES (1 to n)
Embeded Detection Engine
Layer 1. OWASP ModSecurity
Layer 2. Framework/CMS Built-In Security
Layer 3. HTTP Middleware
Layer 4. Template Engine
Layer 5. Data Sanitizer
Opt 1: Detection Engine in HTTP Layer
Opt 2: Detection as Service

Figure 9. Multi-Layer Security Method to Mitigate Multi-Website

Layer 1 or OWASP ModSecurity or OWASP Web Application Firewall Mod Security is a service, contains a set of rules run on a web server and act as a firewall [20]. This service supports the top-level firewall provided by the web server, computer server, and network gears. OWASP ModSecurity acts based on the rule set by default or customized by the web server administrator. Since it runs on a web server, OWASP ModSecurity can guarantee protection for all websites available on the web server. Not only XSS attack, this ModSecurity provides comprehensive protection against several common attack types, they are:

- SQL Injection
- Local File Inclusion
- Remote File Inclusion
- PHP Code Injection
- PHP Code Injection
- Java Code Injection
- HTTProxy
- Shellshock
- OS Shell Injection
- Session Fixation
- Bot Detection
- Metadata/Error Leakages

Layer 2 or Framework/CMS default security features is a term to refer to the default security features of a web framework or content management system used by web developers. In this case, the common security features provided by web framework or CMS (Content Management System) are filtering and validation, captcha challenges, and CSRF protection. The list of popular and commonly used web framework and CMS can be seen on table 4 and 5.

TABLE 4. TOP-7 COMMON USED WEB FRAMEWORK [21-22]

	Web Framework	Language	Official Website
1	Django	Python	https://www.djangoproject.com/
2	Laravel	PHP	https://laravel.com/
3	Ruby of Rails	Ruby	https://rubyonrails.org/
4	ASP.NET	ASP	https://dotnet.microsoft.com/apps/aspnet
5	CodeIgniter	PHP	https://codeigniter.com/
6	Yii	PHP	https://www.yiiframework.com/
7	Express	JavaScript	https://expressjs.com/

TABLE 5. TOP-5 MOST USED CMS [23]

	Web Framework	Language	Percentage	Domains
1	Wordpress	PHP	77,9%	691,237
2	Drupal	PHP	5,6%	49,834
3	Joomla	PHP	3,7%	33,029
4	Squarespace	PHP	2.6	22,694
5	Moodle	PHP	-	-

HTTP middleware can be defined differently, depend on web framework concept. In Java for example, middleware is called filter or C# calls it delegate handler. Basically, HTTP middleware can be assumed as a function to be used as a controller, watcher, sanitizer, or manipulator in HTTP transportation, such as request and response. See figure 10 to see the concept of HTTP middleware. Templating engine is a parser or converter used to provide readable templating system and to sanitize data output. Before showing data to the user, templating engine sanitize the data. The last layer is data sanitizer. This function is used to sanitize data from or to user. This function is called on form processing, database pre-save, etc.

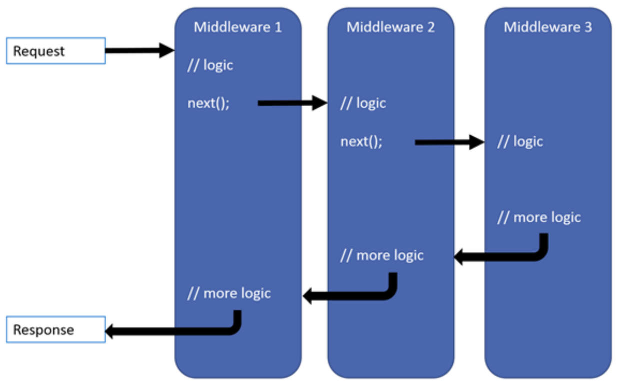


Figure 10. HTTP Middleware Concept in ASP [24]

D. Method Evaluation and Comparison

The most appropriate way to measure and evaluate the accuracy of XSS attack detection and the effectiveness of attack mitigation method is by implementing the XSS attack simulation with real attack codes and scenario. In this case, Zap and Arachni is powerful application for simulating and researching the demand of XSS attack. In addition, to get emphasized method, the proposed method should be compared with previous methods used by other researchers.

E. Complementary Layers in Protecting Multi-Website

It is important to comprehend that the biggest multi-layer security scope is a web server, and the smallest is web application or even micro service application. In this case, every layer has its' own task to protect web application or components. It also means that to protect a web application, multi-layer security is not always in the form of a complete layer, especially for web application built with very secure web framework or code by experienced programmer. Some of security layers may not need to work, because web application can handle the security problem. In protecting web application, this multi-layer security method has several scenarios, see table 6 below.

TABLE 6. THE SECURITY LAYERS IN PROTECTING MULTI-WEBSITES

	Layers and Web Application Capability	Web/App Security Level	Multi-Layer Actions
1	Web application and server has activated all security layers	high	XSS attack can be handled by all security layers simultaneously.
2	Web application has secured HTTP middleware, data sanitation, and powerful templating engine.	high	XSS attack can be handled by web application itself.
3	Web application has no secured HTTP middleware/data sanitation/powerful templating engine features	medium	XSS attack can be handled by OWASP ModSecurity, data sanitizer, and templating engine
4	Web application does not provide secured HTTP middleware, data sanitation, and templating engine.	low	XSS attack solved by OWASP ModSecurity
5	Web application is not protected by OWASP ModSecurity	low	XSS attack can be handled by web application itself.

V. CONCLUSION

The proposed method is divided into detection and mitigation method. XSS attack patterns and combinations vary widely. This attack can also evolve, along with the attacker's abilities. To overcome the diversity of XSS attacks pattern, the proposed method for detecting XSS attacks involves machine learning model. Model analysis used is lexical analysis or computation that can classify text into several classified. The classified text can be easily identified, so the detection process can be effectively done.

In addition to the use of machine learning with lexical analysis, XSS attack mitigation method is implemented with multi-layer security mechanism. This method utilizes five layers of security so that XSS attacks are not easy to be implemented. This security method is carefully structured, so the probability of a successful XSS attack, both stored, reflected, and DOM, is very difficult to achieve.

Based on considerations mentioned in previous explanation and research, the integration between detection and mitigation method to overcome XSS attack is assumed and believed as effective method. This is reasonable statement because the detection and layer items have been measured and researched. In other word, the detection and mitigation method proposed here have been implemented by some researchers. Related to previous research, this proposed method is to develop previous segmented-existing methods with more reliable and comprehensive configuration and advanced customization. The-refore, these two methods are eligible to be proposed.

VI. REFERENCES

[1] B. B. Gupta, S. Gupta, S. Gangwar, M. Kumar, and P. K. Meena, 'Cross-Site Scripting (XSS) Abuse and Defense: Exploitation on Several Testing Bed Environments and Its Defense', *J. Inf. Priv. Secur.*, vol. 11, no. 2, pp. 118–136, Apr. 2015, doi: 10.1080/15536548.2015.1044865.

[2] S. Gupta and B. B. Gupta, 'Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art', *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, no. 1, pp. 512–530, Jan. 2017, doi: 10.1007/s13198-015-0376-0.

[3] D. Wichers and J. Williams, 'Owasp top-10 2017', *OWASP Found.*, 2017.

[4] 'OWASP Top-10 2021, statistically calculated proposal.', *Wallarm Blog*. https://lab.wallarm.com/owasp-top-10-2021-proposal-based-on-a-statistical-data/ (accessed Jun. 19, 2021).

[5] 'Rekap Serangan Siber (Januari – April 2020) | bssn.go.id'. https://bssn.go.id/rekap-serangan-siber-januari-april-2020/ (accessed Jul. 19, 2021).

[6] H. M. Søhoel, 'OWASP top ten - What is the state of practice among start-ups?', p. 94.

[7] 'Types of Cyber Attacks | Hacking Attacks & Techniques', *Rapid7*. https://www.rapid7.com/fundamentals/types-of-attacks/ (accessed Oct. 29, 2021).

[8] 'Top 10 Most Common Types of Cyber Attacks', *https://blog.netwrix.com/*. https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/ (accessed Oct. 29, 2021).

[9] D. A. Prasetio, K. Kusrini, and M. R. Arief, 'Cross-site Scripting Attack Detection Using Machine Learning with Hybrid Features', *J. INFOTEL*, vol. 13, no. 1, pp. 1–6, Feb. 2021, doi: 10.20895/infotel.v13i1.606.

[10] F. M. M. Mokbal, W. Dan, A. Imran, L. Jiuchuan, F. Akhtar, and W. Xiaoxi, 'MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique', *IEEE Access*, vol. 7, pp. 100567–100580, 2019, doi: 10.1109/ACCESS.2019.2927417.

[11] Y. Putra, Y. Yunus, and S. Sumijan, 'Meningkatkan Keamanan Web Menggunakan Algoritma Advanced Encryption Standard (AES) Terhadap Seragan Cross Site Scripting', *J. Sistim Inf. Dan Teknol.*, Sep. 2020, doi: 10.37034/jsisfotek.v3i2.110.

[12] M. F. Kurniawan and W. Setianto, 'OPTIMASI METODE OTOMATISASI PENGHILANGAN KERENTANAN TERHADAP SERANGAN XSS PADA APLIKASI WEB', no. 2, p. 8, 2020.

[13] A. Anggara and R. Somya, 'Pengembangan Sistem Informasi Manajemen Persediaan Barang Dagang Berbasis Web menggunakan Library XSS Filtering', p. 7, 2021.

[14] R. M. Wibowo and A. Sulaksono, 'Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd', *Indones. J. Inf. Syst.*, vol. 3, no. 2, p. 149, Feb. 2021, doi: 10.24002/ijis.v3i2.4192.

[15] M. Shema, *Hacking web apps: detecting and preventing web application security problems*. Amsterdam ; Boston: Syngress, 2012.

[16] 'XSS Attacks Cross Site Scripting Exploits and Defense', p. 482.

[17] *Turi Create*. Apple, 2021. Accessed: Oct. 29, 2021. [Online]. Available: https://github.com/apple/turicreate

[18] A. Kok, I. Ilic Mestric, G. Valiyev, and M. Street, 'Cyber Threat Prediction with Machine Learning', *Inf. Secur. Int. J.*, vol. 47, no. 2, pp. 203–220, 2020, doi: 10.11610/isij.4714.

[19] J. Thanaki, *Python Natural Language Processing*. Packt Publishing Ltd, 2017.

[20] M. Akbar and M. A. F. Ridha, 'SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall', p. 7.

[21] '8 Framework PHP Terbaik untuk Developer', *Hostinger Tutorial*, Mar. 03, 2019. https://www.hostinger.co.id/tutorial/framework-php (accessed Oct. 30, 2021).

[22] A. C. Nisa, '10 Web Development Framework Terbaik Yang Mesti Kamu Pelajari Di 2021', Feb. 16, 2021. https://www.exabytes.co.id/blog/web-development-framework-terbaik-2021/ (accessed Oct. 30, 2021).

[23] 'CMS Comparison - Most Popular CMS 2021 (Statistic)'. https://www.experte.com/website/cms-software (accessed Oct. 30, 2021).

[24] Rick-Anderson, 'ASP.NET Core Middleware'. https://docs.microsoft.com/en-us/aspnet/core/fundamentals/middleware/ (accessed Oct. 30, 2021).