International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

A Contemporary Guideline based Aegis Mechanism for Gateway

Rithik Sandron.Pa[1], Surendar.K[2], Vijay.J[3], Sugania.S.S[4]

[1,2,3]Final Year Student, Dept. of Computer Science and Engineering, Jeppiaar SRR Engineering College, Chennai.
[4]Assistant Professor, Dept. of Computer Science and Engineering, Jeppiaar SRR Engineering College, Chennai.

Abstract - *One of the basic services of the Military is to provide public services to citizens. Service delivery of this type requires public expenditure. Hence, military share data and confidential plans via phones and two-way radio transmitters but these ways are not secure. We overcome this by proposing a new way, a web application to transmit plans, messages and other updates. Using our web application, the transmissions can be done in instantly and also can be transmitted from top level to bottom level officers simultaneously. These transmissions are protected against cyber-attacks in our web application.*

***Key Words*: Secure transmission, JavaScript Content Security Policy(JSCSP), Ease of use, Instant Transmission, Simultaneous message transmission, Resilient design**

1. INTRODUCTION

In this project, we've proposed JSCSP to guard web applications against XSS attacks supported novel self-defined security policies. It's similar functions as CSP, like origin confinement upon both static and dynamic elements. There are several advantages of JSCSP: it's implemented in JavaScript, which enables it to figure on most browsers. JSCSP can defend against attacks that can bypass CSP e.g., UXSS and Code-Reuse attacks via Script Gadgets. Security policies of JSCSP are often generated automatically by analyzing web pages. Advanced features are supported by JSCSP, like cookie protection and JavaScript sandbox which may disable dangerous functions and objects. It has been verified that JSCSP is in a position to affect most real-world XSS threats and compatible with other popular JavaScript libraries.

Cyber security is the field of technologies, processes, and practices used to protect networks, devices and data from attack or unauthorized access. It can be also mentioned as information technology security. It is the protection of Internet-connected systems, including hardware, software, and data from cyber-attacks. It's made from two words one is cyber and other is security. Cyber is said to the technology which contains systems, network and programs or data. Security associated with the protection which includes systems, network security and application
And information security. It's the body of technologies, processes, and practices created to guard networks, devices, programs, and data attack, theft, damage, modification or unauthorized access. Cyber security is vital because government, military, corporate, collect, process, and store unprecedented amounts of knowledge on computers and other devices. a big portion of that data are frequent sensitive information, whether that be property , financial data, or other forms of knowledge that unauthorized access or exposure could have negative consequences. As soon as March 2013, the nation's top intelligence officials cautioned that cyber-attacks and digital spying are absolutely the best threat to national security, eclipsing even terrorism. Some of the elements of Cyber security are network security, application security, endpoint security, data security, etc.

In 1995, Java programing language was developed and released as the component by the Sun Microsystems. Initially, the language was called "Oak" and it had been renamed as "Java" in 1995. The primary motivation of this language was the necessity for a platform-independent language. Atlast, Java is for Internet Programming whereas C programming is for System Programming. Before 1991, JAVA was developed by Sun Microsystems Inc, later acquired by Oracle Corporation. It was designed by James Gosling and Patrick Naughton. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to make reusable code and modular programs. Java may be a class-based, object-oriented programing language and is meant to possess as few implementation dependencies as possible. A general-purpose programing language made for developers to write down once run anywhere that's compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code which will run on any Java Virtual Machine. The syntax of Java is similar to C/C++.

2. RELATED WORKS

G. Xu, X. Xie, S. Huang, J. Zhang, L. Pan, W. Lou and K. Liang, [1]. To mitigate cross-site scripting attacks (XSS), the W3C group recommends web service providers to employ a computer security standard called Content Security Policy (CSP). However, less than 3.7% of real-world websites are equipped with CSP according to Google's survey. The low scalability of CSP is incurred by the difficulty of deployment and non-compatibility for state-of-art browsers. To explore the scalability of CSP, in this paper, we propose JavaScript based CSP (JSCSP), which is able to support most of real-world browsers but also to generate security policies automatically. Specifically, JSCSP offers a

novel self-defined security policy which enforces essential confinements to related items, including JavaScript functions, DOM elements and data access. Meanwhile, JSCSP has an efficient algorithm to automatically generate the policy directives and enforce them in a cascading way, which is more fine-grained and practical than the functionalities provided by CSP. We further implement JSCSP on a Chrome extension, and our evaluation shows that the extension is compatible with popular JavaScript libraries. Our JSCSP extension can detect and block the tested attacking vectors extracted from the prevalent web applications. We state that JSCSP delivers better performance compared to other XSS defense solutions.

The MITRE Corporation. Common Vulnerabilities and Exposures [2]. Computer security is a matter of great interest. In the last decade there have been numerous cases of cybercrime based on the exploitation of software vulnerabilities. This fact has generated a great social concern and a greater importance of computer security as a discipline. In this work, the most important vulnerabilities of recent years are identified, classified, and categorized individually. A measure of the impact of each vulnerability is used to carry out this classification, considering the number of products affected by each vulnerability, as well as its severity. In addition, the categories of vulnerabilities that have the greatest presence are identified. Based on the results obtained in this study, we can understand the consequences of the most common vulnerabilities, which software products are affected, how to counteract these vulnerabilities, and what their current trend is.

M. Heiderich, J. Schwenk, T. Frosch, J. Magazinius, and E.Z. Yang [3]. Hasegawa discovered a novel Cross-Site Scripting (XSS) vector based on the mistreatment of the backtick character in a single browser implementation. This initially looked like an implementation error that could easily be fixed. Instead, as this paper shows, it was the first example of a new class of XSS vectors, the class of mutation-based XSS (mXSS) vectors, which may occur in innerHTML and related properties. mXSS affects all three major browser families: IE, Firefox, and Chrome. We were able to place stored mXSS vectors in high-profile applications like Yahoo! Mail, Rediff Mail, OpenExchange, Zimbra, Roundcube, and several commercial products. mXSS vectors bypassed widely deployed server-side XSS protection techniques (like HTML Purifier, kses, htmlLawed, Blueprint and Google Caja), client-side filters (XSS Auditor, IE XSS Filter), Web Application Firewall (WAF) systems, as well as Intrusion Detection and Intrusion Prevention Systems (IDS/IPS). We describe a scenario in which seemingly immune entities are being rendered prone to an attack based on the behavior of an involved party, in our case the browser. Moreover, it proves very difficult to mitigate these attacks: In browser implementations, mXSS is closely related to performance enhancements applied to the HTML code before rendering; in server side filters, strict filter rules would break many web applications since the mXSS vectors presented in this paper are harmless when sent to the browser. This paper introduces and discusses a set of seven different subclasses of mXSS attacks, among which only one was previously known. The work evaluates the attack surface, showcases examples of vulnerable high-profile applications, and provides a set of practicable and low-overhead solutions to defend against these kinds of attacks.

The Acunetix. Universal Cross-site Scripting [4]. Common cross-site scripting (XSS) attacks target websites or web applications that are vulnerable to XSS, because of inadequate development of client-side or server-side code. These attacks have the vulnerable web page as main prerequisite, and their effect is always revolving around the user session on the vulnerable web page itself. In other words, if a user had to browse to a website vulnerable to XSS while having a web-based CRM application open, the attacker will only gain access to the compromised session and cannot gain access to the CRM's session. This behavior is a result of security functionality implemented in browsers.

3. PROPOSED SYSTEM

In this project, we have proposed JSCSP to protect web applications against XSS attacks based on novel self-defined security policies. It has similar functions as CSP, such as origin confinement upon both static and dynamic elements. There are several advantages of JSCSP: It is implemented in JavaScript, which enables it to work on almost all browsers. JSCSP can defend against attacks that can bypass CSP e.g., UXSS and Code-Reuse attacks via Script Gadgets. Security policies of JSCSP can be generated automatically by analyzing web pages. Advanced features are supported by JSCSP, such as cookie protection and JavaScript sandbox which can disable dangerous functions and objects. In our evaluation, it has been verified that JSCSP is able to deal with most real-world XSS threats and compatible with other popular JavaScript libraries.

3.1 Advantages of Proposed System

1. By using JSCSP concept we can achieve hundred percent compatibility for security.
2. There will no hacking and viruses' threats.
3. Web page will be secure.
4. Military communications are easily achieved.
5. Undercover projects are achieved without any disturbance.

4. SYSTEM ARCHITECTURE

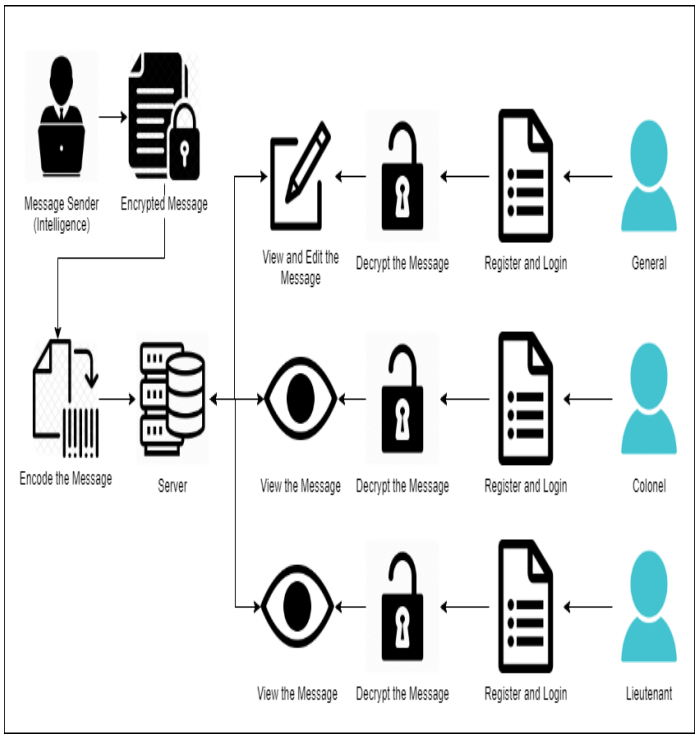

Fig -1: System Architecture

5. REQUIREMENTS SPECIFICATION

5.1 Hardware Requirements

Processor: Intel ® Pentium ®
RAM: 4 GB and above
Hard Disk: 120 GB
Speed: 1.6 GHz and above
Monitor: 15'' LED SVGA
Input Devices: Keyboard, Mouse

5.2 Software Requirements

Operating System: Windows 7/8/8.1/10
Coding Language: JAVA/J2EE
Java Version: jdk 8
IDE: Eclipse Oxygen
Database: MYSQL v5.1
Database Tool: HeidiSql v11.0/MySQL v3.5
Application Server: Apache Tomcat 8.X/9.X

6. MODULES AND DESCRIPTION

6.1 Intelligence

In this module, the Admin(Intelligence) is in a position to register and create an account using the Sign-Up form. during this form the fields 'Email', 'Password' are required as Email is shipped to the user. Once they register successfully, they're ready to compose and send messages. These messages can contain any number of alphabets, words, special characters, numbers etc. The Intelligence branch is brain of Military communications. Each and each order/message is conveyed to the intended recipient by Intelligence branch. In our application only Intelligence has the authorization to send orders/messages. This is most appropriate since the work of Communication Intelligence is to convey order/messages to the intended recipient. The message that's sent by Intelligence is Encrypted and uploaded to the server. The Decryption key's sent to the intended recipient's Email ID. they will use that key to decrypt the message and consider it.

6.2 General

In this module, the General(Top Order) is in a position to look at the message that was sent by Intelligence. they're also ready to monitor the message transmission history including the sender E-Mail ID, Message ID, Date, Time and the Encrypted file. this manner any suspicious message transmission or unauthorized transmission are often identified by the overall who is within the Top Order of the Military Hierarchy. The Chief of Army Staff(General) is that the top of military staff of the Indian Army they need the very best level of authorization. In our application, they're ready to monitor each and each message/order transmission. this manner they're ready to identify any/all unauthorized message transmission. Since the message history contains every details of the message transmission, the overall being the top of military staff will be ready to take swift action against the perpetuator.

6.3 Colonel

In this module, the Colonel(Middle Order) is in a position to look at messages that were sent by the Intelligence. They're also ready to send specific messages(orders) to the respective assault team(Team A/Team B/Team C). While sending the message and ID they have to specify to which team the message is shipped, whether A or B or C. By using this method, the Colonel is in a position to give precise orders to every team without overlapping or confusion. The Colonel may be a rank above Lieutenant and less than General. They serve as the captain of the brigade. Usually Colonels issue service orders to the Lieutenants. They micromanage the items that happen on the sector. In our application, Colonel is in a position to look at messages and send team specific messages. A Colonel controls variety of field teams. This needs them to send Team Specific orders which shouldn't overlap/conflict with the orders for the opposite teams. To prevent this, Colonel is in a position to send messages to specific teams(A,B,C). They can send separate team specific messages to stop conflicting of orders.

6.4 Lieutenant

In this module, the Lieutenant(Bottom Order) is in a position to read the messages that's sent to them. Lieutenants got to choose a Team(A/B/C) according to their orders while Signing Up. This Team Code is employed by the Colonel to send Team specific orders. This prevents overlapping of message for different Teams. The Lieutenant is additionally ready to utilize the Key which will be sent by Colonel to decrypt and skim the message that was sent to them by the Intelligence. They're also ready to update messages with reply. They will send replies to messages that were sent to them by the Colonel. The Lieutenant is that the bottom order of the Hierarchy. Usually they're those who work on the sector. Each Lieutenant are going to be assigned to a team by their Colonel. They have to register under the right team so as to urge the right messages. Multiple Lieutenants can register under an equivalent team should the scenario arise that demands multiple Lieutenants to figure under an equivalent team for efficient mission completion. Once the mission is completed or any information must be conveyed to the colonel, they're ready to update the message that were sent to them. They are also ready to view the first order that were sent to the Colonel by the Intelligence. To look at the first message, they have to urge Authorization from the Colonel. Once they get authorized, they will use the key sent by Colonel to Decrypt and consider the message.

6.5 Security

In this module, JSCSP and Email Encryption is implemented. According to a Google Survey last year, only 6% of the online applications have implemented CSP(Content Security Policy) due to its complexity and limited compatibility. Using JavaScript Content Security Policy(JSCSP) we overcome this vulnerability. The web's security model is rooted within the same-origin policy. Each origin is kept isolated from the remainder of the online, giving us a secure sandbox to develop and run our application. Since JavaScript is universal, CSP are often implemented using JavaScript to eliminated the complexity and compatibility issues.

7. RESULTS

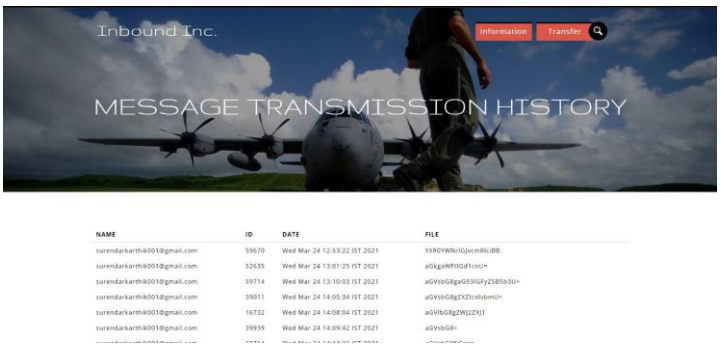

Fig -2: General Message Transmission

Here message are uploaded by Intelligence by encoding using JSCSP(JavaScript Content Security Policy). All the transmisson historys are saved along with the name, message id, time and encrypted message. Both the General and Colonel will be able to monitor the transmission. The lieutenant able to utilise the key that can be sent by Colonel to decrypt and read the message that was sent to them by the Intelligence.

8. CONCLUSION

We are able to prevent one of the major security vulnerabilities that can be exploited in around 94% of the web applications. Since we use JavaScript to implement CSP it can be easily adapted to various number of web applications without much changes. This would increase the overall security of the web application. For future developments, The JSCSP can be commercialized to be used on any web application. Here encode and decode concept is included with encryption and decryption. Messages are transferred instantly. More over hackers cannot hack any messages and viruses cannot be induced in web page.

REFERENCES

[1] G. Xu, X. Xie, S. Huang, J. Zhang, L. Pan, W. Lou and K. Liang, "JSCSP: a Novel Policy-Based XSS Defense Mechanism for Browsers", IEEE Transactions of Dependable and Secure Computing, China, 2020

[2] The MITRE Corporation. Common Vulnerabilities and Exposures: The Standard for Information Security Vulnerability Names[EB/OL].2017[2017-09-29]. http://cve.mitre.org.

[3] M. Heiderich, J. Schwenk, T. Frosch, J. Magazinius, and E.Z. Yang,“mxss attacks: attacking well-secured web-applications by using innerhtml mutations,” in CCS, Berlin, Germany, 2013, pp. 777-788.

[4] The Acunetix. Universal Cross-site Scripting (UXSS): The Making of a Vulnerability[EB/OL].2017[2017-10-2].https://www.acunetix.com/blog/articles/universal-cross-sitescripting-uxss.

[5] S. Stamm, B. Sterne, and G. Markham, “Reining in the web with content security policy,” in WWW, Raleigh, 2010, pp. 921-930.

[6] The W3C Working Draft. Content Security Policy Level 3[EB/OL].2017[2017-09-29]. https://www.w3.org/TR/CSP3/.

[7] Can I use. Content Security Policy Level 2[EB/OL]. 2017[2017-09-29].https://caniuse.com/#search=csp.

[8] S. Lekies, K. Kotowicz, S. Groß, E. Nava and M. Johns, "Code-Reuse Attacks for the Web: Breaking Cross-Site Scripting Mitigations via Script Gadgets," in CCS, Dallas, Texas, USA, 2017, pp. 1709-1723.