NFC-IEFR Journal of Engineering & Scientific Research
Journal URL: www.nijesr.iefr.edu.pk/

Analysis of Risks against Web Applications in MVC

Pariwish Touseef , Moeez Ameer Ashraf, Ammar Rafiq*

Department of Computer Science, NFC Institute of Engineering and Fertilizer Research, Faisalabad, Pakistan

ARTICLE INFO

***Corresponding Author:**
ammar.rafiq@iefr.edu.pk
DOI:
10.24081/nijesr.2017.1.0005
Keywords:
MVC, Risks, Security, Architecture

ABSTRACT

In this era of internet web security is the main issue. The internet growth led the web crimes to increase. The enhancement of new architectures makes the web application prone to security risks. Now a day's Model View Controller Architecture is very well known name for web application development. It could be a very good solution for rapid application development. This model can be implemented in PHP and ASP.Net. With comparison of both languages regarding their performance, time, and memory our findings show that ASP.Net is better than PHP framework. But there are several security risks that exploit the Model View Controller Architecture. The Open Web Application Security Project identifies common types of risks a typical architecture can expect. Researchers are paramount to secure the websites. Hence security have become major concern and researchers have developed various security detection and prevention approaches. This paper provides the overview of the web applications developed in Model View Controller Architecture using ASP.Net or PHP with common web applications risks that are used to tamper systems assets and possible solutions to secure them. It also proposes the further possibilities to develop a security model in Model View Controller ASP.Net for the protection of sensitive data exposure risk.

I. INTRODUCTION

With the development of Internet, Web Applications have observed overwhelming growth, which increases the number of architectures such as MVC .The Model View Controller (MVC) is architecture to develop ASP.Net and PHP Applications [1].

The Implementation of MVC architecture with ASP.Net Framework is more easy and higher in performance then PHP Framework [2]. But the security of applications becomes more critical due to increase in number of technologies making data vulnerable to attacks. The Open Web Application Security Project (OWASP) reviewed the popular and common web applications risks. The categories specified are: [3]

1) Injection
2) Broken Authentication and Session Management
3) Cross Site Scripting (XSS)
4) Insecure Direct Object References
5) Security Misconfiguration
6) Sensitive Data Exposure
7) Missing Function Level Access Control
8) Cross-Site Request Forgery (CSRF)
9) Unvalidated Redirects and Forwards

It is very difficult to achieve entire security of the web; there are flaws that distinct the application, especially in website developed in MVC using ASP.Net [4]. The Sensitive Data Exposure is generally believed to be one of the most common risks in web applications. It led unlawful activities on web information related crimes so it is important to secure it from attacks.

Attackers usually capture the sensitive information by doing a number of attacks on the data. Many studies conducted by researchers against the types of attacks which concluded various types like client based attack, Data Base attacks, Communication Channel Attacks, Server Attacks and many more which exploits the data. Therefore, there is need to investigate the security measures [5], to determine most appropriate approach for securing the data against attacks.

II. LITERATURE REVIEW

In this section we look at the previous literature behind the architecture of MVC. In paper "Designing a MVC model for rapid web application development" by Dragos-Paul Pop and Adam Altar has provided an overview of the model view controller architecture and explained the components of security. The MVC architecture maintain application domain model, the presentation of that model with interaction and show that MVC divide the web into three components the

model of domain application, the presentation of data model and user interaction [6].

The three layer MVC architecture (discussed in MVC structure below) identifies that input, process and output separately .While Model defines entities, View defines user interface and Controller receive information from view, perform logic and pass data backup. There are different languages frameworks for creation of MVC architecture web application such as Analysis and Implementation of ASP.Net and PHP frameworks based on MVC architecture by Xiaokang Liu and Gengguo Cheng has highlighted the MVC architecture and its performance with both ASP.Net framework and PHP Framework and from many papers it has been concluded that MVC architecture in ASP.Net [7] framework is easy to implement than PHP framework due to its program complexity, good performance discussed in comparative analysis [2] .The ASP.Net is good way to implement MVC architecture but the security risks are being found due to the attacks by the hackers [8].

The OWASP lists [9] the top ten biggest risks faced by web applications. Previous literatures evaluated different risks on MVC websites are facilitated by exploitable attacks. The possible risks a web application includes with its security measures are listed below:

A. Injection

Injection flaws includes SQL [10], OS ,inferences [11] it occurs when un trusted data is sent .Attackers retrieve crucial information from database server. And retrieve information by unauthorized means.

B. Broken Authentication and Session Management

External Hackers with their own account try to hack others accounts [9]. Allow attackers to compromise sessions, keys, passwords and assumes the identities of users. Different types of cryptographic algorithms are there with session management tokens for these types of risks. It is attacked if least password strength is there, password change with control and session id protection [11].

C. Cross Site Scripting (XSS)

It takes place when JavaScript or HTML code is added to the database. It flaws application when fake data is sent over browser page without authentication and validation .Authentication includes brute force attack, network sniffing, identity thefts and cookie faking etc [6].

D. Insecure Direct Object References

It occurs when reference to internal implementations are exposed such as database key s, files .Without access hackers manipulate the data and insecure the sensitive information [12].

E. Security Misconfiguration

A good security involves secure configuration for database, frameworks and applications, The settings defined, features defined are sometimes not secure and led to risks.

F. Sensitive Data Exposure

Attackers steal and change the sensitive data [13] that includes database security [14] ,credit card number, taxes, passwords, identity thefts, Token Cracking, Add frauds ,Brute force ,Parameter tempering etc [15].

A literature review Security in Electronic Payment Systems by Roxana Turcu describe the security of a sensitive data in web develop in MVC Architecture using ASP.Net Framework. Payment system introduced that has three parts Test Shop, Payment Gateway and Payment provider as MVC. For online payment system the security against whole data explained in it with types of e commerce attacks on that data e.g.: Intellectual Property Threats that is using of assets without permission of owner e.g. Software Pirating, Client Computer Threats in which the computer of client is infected by Active Viruses and Contents, Communication Channel Threats includes Brute Force Attack, Copy of websites, Phishing Attack against sensitive data like credit card numbers etc, server threats which include spamming, file transfer in this server is not secure and changing database data. This literature stated the consents of risks that cause sensitive data exposure. [16].

G. Missing Function Level Access Control

Attacker changes the URL parameters and anonymous functions are being accessed. This can be difficult to provide security at missing function level control [17].

H. Cross-Site Request Forgery (CSRF)

It makes fake logging on the victim's browser and sends forged HTTP requests [18] including cookies and Validation and authentication information. Attackers make browser feel request is from user and exploits data.

I. Using Components with Known Vulnerabilities

Components include libraries, frameworks and software modules are vulnerable. If exploited by the hackers led to swear server takeovers with whole system data.

J. Unvalidated Redirects and Forwards

Websites redirect to more pages and use data i.e. un trusted to determine the page of destination. Without proper validation attackers can make the user to phishing or malware pages to access the browser page [19]. The attacks led the risks which affect the security. Authors from different literatures mention the counter measures against those risks.

The risks are mostly categorized into Server Side Attacks which is consider as web server attack, Client Side Attacks and Network based attacks. The common countermeasures
To stop them are using web frameworks, Mozilla's contents security and using load balance systems [20].

The ORM (Object Oriented Mapping Model) with encryption is used for safe data storage in database against Injection, data risks against sniffing, brute force cookie faking and network attacks etc. The Vulnerabilities of attacks in MVC Architecture is secured by ORM [6].

The Client Computer Protection by Digital Certificates, Browser Protection and Cookies Protection, Communication Channel Protection use SSL-Protocol and Digital Signature and Server Protection by giving control only to users, authentication of user and firewalls. To avoid loss of sensitive data between payment gateway and shop module encryption done using RSA Algorithm and for decryption is done in payment gateway and MDS Hash Algorithm use for secure payments to shop and database sensitive data protected by user name and password [16] .

Muhammad Imran Hussain and Naveed Dilber has provided an overview of the security areas and techniques to secure the

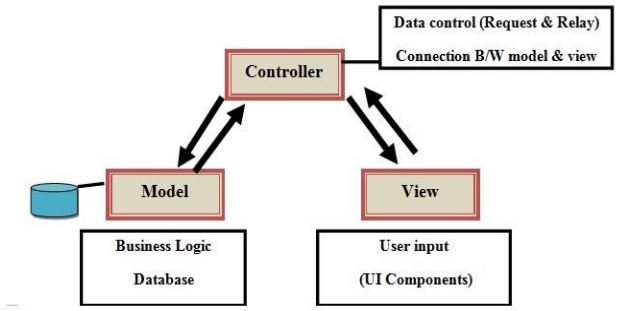

Figure 1: MVC Structure

data delivery at its destination basically to secure RESTAPI by MVC Web API Architecture using ASP.Net by using security techniques e.g. OAuth2 (Claimed Based), Web token (Secure using SSL) and delegation against attacks of JSON Injection, Cross Site Scripting etc [18]

An overview of the dynamic security policy against sensitive data exposure risk defines by OWASP. A model is developed using Oracle. In this Author Propose a model of SDF (Sensitive Data Filter) at database level on oracle database creating custom security functions. The SDF have tables to store entities i.e. objects. There are three objects identified: SDF User Master, SDF Application Master and SDF Policy Master [13].

To store identified sensitive data and stored information into db objects created. Any query executed fine grain policy will check whether any security policy is set on objects used in the query ,If yes, it will modify the query and run else run query as it If no security it will execute the query as it is. Water mark technique, Encryption, Decryption, Stenography is used for prevention of unauthorized access to sensitive data [13]

Security policies are design, define and implement against attack on database. The numbers of attacks with their prevention measures listed are: Excessive privilege attack and countermeasure of access control policy, SQL Injection its security is provided by means of using stored procedures instead of using direct queries and Implementing MVC Architecture, Mal ware Attacks can be secured by Firewall Protection, Unmanaged Sensitive data and secured by encryption, stenography of data . [10]

Nelly Delessy-Gassant and Eduardo B. Fernandez have provided an overview of the MVC Structure with countermeasures to secure it. Mention some security patterns such as Authenticator; Role based access, Secure Channels and Secure Logging's by adding all these to components of MVC. The Secure MVC patterns allow the user to modify after accessing the sensitive information in the MVC components. The sensitization of the data is done to prevent malicious attacks e.g. Cross Site Scripting and SQL Injection. Countermeasures such as Authentication, Authorization and Secure Logging are done to secure MVC Patterns [21].

III. MVC STRUCTURE

The Model View Controller (MVC) understands and modifies each particular unit without having to know everything about the other units. An application is divided into three main categories: the model of the main application domain, the presentation of data in that model and user interaction [6].The Figure 1 below shows the MVC structure and Figure 2 shows the Functions of MVC.

MVC Architecture	
Model	Defines entities
View	User Interface
Controller	Receive input from view, Perform logic/updates & Pass Data back

Figure 2: MVC Functions

IV. COMPARITIVE ANALYSIS

The web security is big challenge for researchers as there are several new technologies and architectures exist for web application development. One of them is MVC. It can be

implemented by using different coding languages like C, Java, PHP and ASP.Net etc.

Web developers mostly prefer the use of PHP and ASP.Net. We reviewed literature of different authors to check compatibility of both languages with the MVC architecture to counter the OWASP risks. The comparison of the PHP with ASP.Net using MVC Architecture regarding their response time, Program Complexity, Memory Usage and implementation is shown below in the Figure 3. Which shows that ASP.Net is better than PHP Framework to add security to MVC Architecture.

The web is not a secure world which led to face pressing issue to the organizations. The increasing attacks lead the OWASP organization to identify the risks that organizations are facing. We depict those vulnerabilities identified by OWASP in web applications. The mostly occurring risks include the Injection, Sensitive data exposure; Cross Site Request Forgery and Cross Site Scripting (XSS). The Figure 4 below indicates the percentage of vulnerabilities

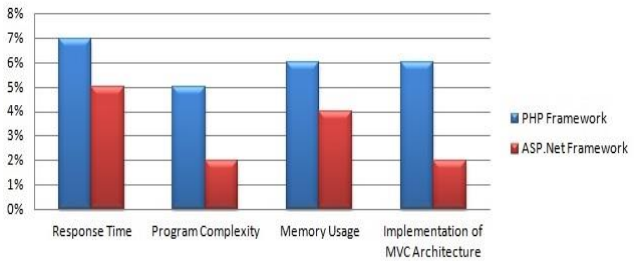

Figure 3: Comparison between PHP and ASP.Net Framework

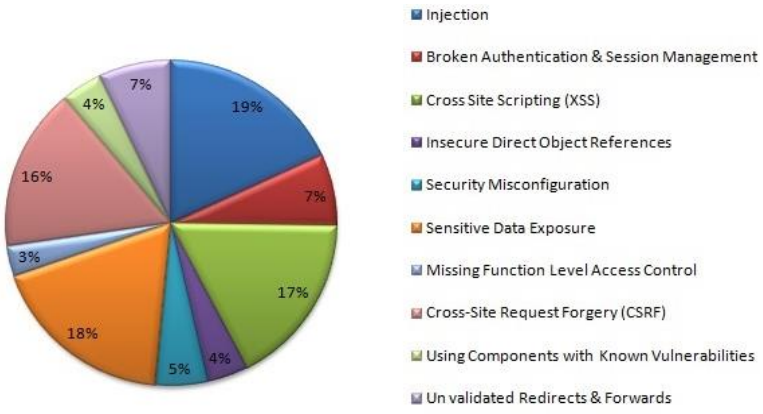

Fig. 4: Web Risks

The OWASP top 10 risks occur due to different types of attacks like client based attacks, server based attacks, application attacks and web service attacks etc .In Table 1 below we mention the types of risks occur in web applications the characteristics and the impact of the attacks which led to the top 10 OWASP risks with the response by the researchers to provide security measures for attacks and a brief description on the basis of their performance of their security measures.

In Table 1 we have shown that the hacker's attacks mostly led the Injection, Sensitive data exposure, and Cross Site Request Forgery and Cross Site Scripting (XSS) risks to increase in web applications. We depict that web applications designed in MVC Architecture have gone through these risks but no one provide efficient model of security. Therefore, this may suggest for future research that the sensitive data exposure is one of the highly occurring risks, but there is a lack of study in MVC ASP.Net security.

V. CONCLUSION

Through the comparison of different authors we concluded that ASP.Net is better than PHP and easier to implement the MVC architecture and has realistic significance for identifying the OWASP risks in web applications. The result of this emphasize that the mostly occurring web risks are Injection, Sensitive data exposure, Cross Site Request Forgery and Cross Site Scripting (XSS). Despite the rising importance of security practices are still ignored and there is a lack of study to implement security of ASP.Net MVC architecture.

VI. FURTHER RESEARCH

The focus of our research in this area suggests that there is need to develop a security model in MVC ASP.Net for the protection of sensitive data exposure risk. So a solid and safe impression of risk against various attacks could be achieved in MVC ASP.Net.

Table 1: Web Risks with its impact and security measures

Risks Names	Impact of Attacks which lead to risks in web applications	Response to provide Security Measures	Brief Description
Injection	High	Good	High impact of attacks led injection risks to increase and more security measures for it.
Broken Authentication & Session Management	Medium	Average	Medium impact of attacks led this risk to increase on normal bases and normal security measures for it.
Cross Site Scripting (XSS)	High	Good	High impact of attacks led this risk to increase and more security measures for it.
Insecure Direct Object References	Low	Poor	Low impact of attacks led this risk not to increase and very less security measures for it.
Security Misconfiguration	Medium	Average	Medium impact of attacks led this risk to increase on normal bases and normal security measures for it.
Sensitive Data Exposure	High	Good	High impact of attacks led this risk to increase and more security measures for it.
Missing Function Level Access Control	Low	Poor	Low impact of attacks led this risk not to increase and very less security measures for it.
Cross-Site-Request Forgery (CSRF)	High	Good	High impact of attacks led this risk to increase and more security measures for it.
Using Components with Known Vulnerabilities	Low	Poor	Low impact of attacks led this risk not to increase and very less security measures for it.
Un validates Redirects & Forwards	Medium	Average	Medium impact of attacks led this risk to increase on normal bases and normal security measures for it.

VII. REFERENCES

[1] M. Vermani, "Review of some comparisons between php and asp.net from the viewpoint of a novice programmer," *international journal of engineering sciences & research technology*, vol. 5(3), p. 8, March, 2016].

[2] X. L. &. G. Cheng, "Analysis and Implementation of ASP.Net and PHP Frameworks Based," *Trans Tech Publications*, p. 4, 2013.

[3] R.Dunne, "OWASP Top 10 – 2013," 23 May 2013. [Online]. Available: https://dunnesec.com/category/test-standards/owasp-test-standards/owasp-top-10-2013-owasp/.

[4] G. K. &. Pannu, "A Survey on Web Application Attacks," *International Journal of Computer Science and Information Tech*, vol. 5(3), p. 5, 2014.

[5] S. Bageri, "10 Points to Secure Your ASP.NET MVC Applications".

[6] D.-P. P. &. A. Altar, "Designing an MVC Model for Rapid Web Application Development," *International Symposium on Intelligent Manufacturing and Automation*, vol. 69, p. 8, 2014.

[7] D. M. Mistry, "Asp.net Mvc With Derivative Patterns And Its," *International Journal of Information and Computing Technology (Research@ICT)*, vol. 4 , no. Issue II, p. 4, 2014.

[8] MRC, "Solving the Top 10 Application Security Threats," [Online]. Available: http://www.mrc-productivity.com..

[9] R. Motwani, "A list of the 10 Most Critical Web Application Security Risks : OWASP Top 10," 21 Sep 2015. [Online].

Available: https://ramamotwani.wordpress.com/2015/09/21/a-list-of-the-10-most-critical-web-application-security-risks-owasp-top-10/.

[10] A. G. &. S. Singh, "A Review on Web Application Security Vulnerabilities," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 3, p. 5, January 2013.

[11] G. K. Pannu, "A Survey on Web Application Attacks," *International Journal of Computer Science and Information Tech,* vol. 5(3), p. 5, 2014.

[12] Arbin, "Top 10 2010-A4-Insecure Direct Object References," 1 May 2010. [Online]. Available: https://www.owasp.org/index.php/Top_10_2010-A4-Insecure_Direct_Object_References.

[13] J. D. &. B. Trivedi, "Sensitive Data Exposure Prevention using Dynamic Database Security Policy," *International Journal of Computer Applications,* Vols. 106 – No. 15,, p. 5, November 2014.

[14] M. M. a. T. Patel, "DATABASE SECURITY - ATTACKS AND CONTROL METHODS," *International Journal of Information Sciences and Techniques,* Vols. 6, No.1/2, p. 9, March 2016.

[15] C. Watson, "Automated Threats Web Applications," *International Journal of Information Sciences and Techniques,* vol. 4, p. 2, July 2015.

[16] R. TURCU, "Security in Electronic Payment Systems," *Journal of Mobile, Embedded and Distributed Systems,* vol. 4, p. 6, 2014.

[17] N. Biller, "OWASP Top 10: Application Security Risks," 15 July 2016. [Online]. Available: http://blog.blackducksoftware.com/owasp-top-10-application-security/.

[18] M. I. H. a. N. Dilber, "Restful web services security by using ASP.NET web API MVC based," *Journal of Independent Studies and Research,* vol. 12, p. 7, 1 January 2014.

[19] P. J. a. J. Fontana, "10 STEPS TO SECURING YOUR WEB APPLICATIONS," *Unisys Corporation,* vol. 2, p. 10, 2015.

[20] M. Friederichs, "Web System Attacks," *SE411,* vol. 3, p. 11, April 18, 2013.

[21] N. D.-G. &. E. B. Fernandez, "The Secure MVC pattern," *International Symposium on Software Architecture and Patterns,* vol. 2, p. 6, 2012.