AN INVESTIGATION ON CORE VULNERABILITES FOR WEB APPLICATION

Mrs M. SRIDEVI
Research Scholar, Assistant Professor
Laqshya Institute of Technology and Sciences, Khammam
Email Id: sreetech99@gmail.com

Dr. K. V. N. SUNITHA
Principal & amp; Professor
BVRIT College of Engineering for Women
Email Id: k.v.n.sunitha@gmail.com

ABSTRACT:

The modern tendency due to rapid growing of internet communications and accessibility of web based applications, in real world web is a stateless but web developers without worrying security challenges for websites they are building web applications they are mainly focused for fast and scalable access for different clients. Hence security for web application is major concerned and mostly is divided into two types one is client based or internal security and another one is server side or external security so in this paper we reviewed various types of security attacks and how they are threat to the web applications same way our review based on two categories static based and dynamic based review. And in order to aware or visible all kinds of vulnerabilities and security for web application an online community is developed called as Open Web Application Security Project (OWASP) using this developers and companies can make assessment about security for web applications.

KEYWORDS: *Security vulnerability, Web Application, XSS, OWASP.*

INTRODCUTION:

Open Web Application Security Project is listed some serious web application security flaws from that we discussing each one with detailed.

Cross Site Scripting (XSS): it is a client side and injection type attack in this a web site injected with malicious script, the client browser doesn't know whether the script came from trusted source or no and finally it will execute the script. XSS is one of major problem for many web applications according to White Heat security more than 50% of web applications are vulnerable to XSS, so before designing any web site any web developer must understand security issues with XSS.

Cross site scripting can be injected in the following scenarios

- TextBox (input controls)
- Query strings
- Cookies
- Session variables
- Application variables

Injection Flaws: injection flaws permit pass malicious code to one web application to another and it is severe vulnerable to web applications, injection flaws lead to web application weakness and failure to filter input text.

There are three types of injections attacks

System calls

Shell commands, Sq injection

Malicious File Execution: this type of vulnerable allow attacker to include file, this kind of attack mostly occurs at server side, so server can't decide whether to trust the file or not and it execute malicious file at a server side.

Insecure Direct Object Reference: The possible risk here is that aggressor could manipulate those references to access other objects without any permission. A direct object reference happen when a developer rendering a reference to an internal execution object, such as a file, directory, database record, or key, as a URL or form parameter.

Cross Site Request Forgery (CSRF): The potential threat from this flaw is that it would force a logged-on victim's browser to send a pre-authenticated request to a vulnerable net application that then forces the victim's browser to perform a hostile action to the good thing about the assaulter. CSRF are often as powerful because the net application that it attacks.

Information Leakage and Improper Error Handling: The potential threat from this flaw is that attackers will use this weakness to steal sensitive knowledge, or conduct a lot of serious attacks. Applications will accidentally leak data regarding their configuration, internal workings, or violate privacy through a spread of application issues.

Broken Authentication and Session Management: The possible risk here is that attackers may compromise passwords, keys, or authentication tokens so as to assume the identity of alternative users. This flaw is caused once account credentials and session tokens aren't properly protected.

Insecure Cryptographic Storage: This possible threat comes once attackers use poorly protected information to conduct fraud and different crimes, like mastercard fraud. This flaw is owing to net applications not creating correct user of science functions to safeguard information and credentials.

Insecure Communications: This defect originates from the conceivable spillage of delicate data over the system correspondence framework. This is caused by an inability to scramble arrange activity when it‟s important to secure delicate interchanges.

Failure to Restrict URL Access: This defect gives assailants the chance to get to and perform unapproved operations by getting to those URLs specifically. This imperfection is caused by applications that lone ensure touchy usefulness while keeping the show of connections or URLs to unapproved clients.

LITERATURE REVIEW:

For secure web applications many researches proposed different prevent mechanisms so in this part reviewing some of authors work. And this literature provide different issues which identified previously.

Gary Wassermann and Zhendong Su (2007) specifications are often avoided by queries that user input changes the meant grammar structure of the generated question as attacks. they need enforced the planned technique for PHP and this technique with

success discovered unknown and refined vulnerabilities in globe programs. Avik Chaudhuri and Jeffrey S. Foster (2010) given a symbolic security analysis on rails for net applications so as to observe cross website scripting attacks. they need thought-about cross website request forgery, too little authentication, leaks of secret info, too little access management and application specific security properties in their work.

Monga et al (2009) presented a hybrid analysis framework for detective work net applications vulnerabilities that blends along the strengths of static and dynamic approaches for the detection of vulnerabilities in net applications. In their work, a static analysis performed only once is employed to scale back the run time overhead of the dynamic observance part. Their system is capable of statically analyzing script language computer memory unit codes by finding out dangerous code statements. Moreover, it monitors solely the known statements throughout the dynamic analysis part.

Michelle Zhou and Prithvi Bisht (2010) proposed a way to deal with reinforce Cross Site Request Forgery (CSRF) protections for heritage web applications utilizing the white box investigation and change. Monica et al (2008) depicted a dialect called Program Query Language (PQL) that enables clients to give questions to extricate data stream designs adequately and definitively. For this reason, they built up a static setting touchy, however stream obtuse data stream following examination that can be utilized to discover every one of the vulnerabilities in a program. Their dialect is valuable for investigating the information vectors that uncover the powerlessness. Notwithstanding, most works consider just numeric imperatives. Then again, in the present web situation, emblematic controls are important to distinguish and avoid novel assaults. Keeping in mind the end goal to accomplish this, it is important to propose new calculations those utilization emblematic imperatives for viable location of assaults.

Mike Ter Louw et al (2007) proposed associate degree protrusive browser security model that uses code integrity checking techniques to regulate the extension installation and loading method. They conjointly projected methodologies for runtime watching of extension behavior that offer a foundation for defensive threats attributable to put in extensions. In another work, electro-acoustic transducer Ter Louw et al (2008b) conferred associate degree analysis on machine-readable text isolation techniques for XSS hindrance that focuses on security, browser compatibility and alternative necessary qualities.

Raymond Wu and Masayuki Hisada (2010) explained their work by exploitation macro and small views. The macro read oversees syntax structure and identification, whereas small read envisions information electronic communication and program automation. The coherence of macro and small views forms the online security framework for trailing and validation of the code. Their work is accustomed perform the safety services through fraud detection. It conjointly demonstrates information electronic communication for trailing and

code generation techniques for validation that bridges the gap between static and dynamic analysis. Viktoria Felmetsger (2010) projected a way towards automatic detection of application logic vulnerabilities in net applications. They used dynamic analysis to look at the conventional operation of an internet application and thus to infer a collection of behavioral specifications. Their system is capable of finding unknown logic vulnerabilities.

Sven Lachmund (2010) proposed auto producing access control arrangements for web applications utilizing static examination strategies. Utilizing these strategies and client input acknowledgment, it is conceivable to recognize asset gets to started by the application from those started by the client. In addition, it creates an application strategy which just contains get to rights that are not gotten from client cooperation. Their strategy show additionally fulfills the standard of slightest benefits. In their model, the client started gets to are taken care of independently at runtime which are resolved and separated in that framework. Andrea Avancini and Mariano Ceccato (2010) proposed a strategy towards security testing in view of spoil examination that performs preparatory examination on the combination of static investigation with hereditary calculations. Their approach recommends methods to separate competitor false positives announced by static investigation and gives input vectors that uncover genuine vulnerabilities.

Venkatakrishnan et al (2010) planned a framework for the bar of SQL injection and XSS attacks on net applications. Their framework incorporates techniques supported static and dynamic analysis, symbolic analysis and execution watching to retrofit existing net applications to be resilient to those attacks. All the dynamic tainting strategies according within the literature focus solely on securing the net applications from SQL injection and cross web site attacks by dynamic tainting analysis. However, most of those techniques area unit capable of solely familiar attack patterns and thence the attacker's area unit able to perform new attacks supported new attack patterns. Hence, it's necessary to propose and implement new and effective detection techniques that area unit capable of police investigation each familiar and novel attacks. For this purpose, a replacement detection and bar model that gives each static and dynamic analysis has been projected and enforced during this analysis work.

William Halfond et al (2006, 2008) built up the Web Application Security Provider (WASP) that ensures web applications utilizing positive spoiling and grammar mindful assessment. In addition, WASP secures a few web applications and subjected them to an extensive and fluctuated set of assaults and real gets to. Abdul Bashah Mat Ali et al (2011) talked about the advancement of another web examining instrument with improved highlights that can direct effective entrance test on PHP based sites to recognize SQL infusion vulnerabilities. This strategy mechanizes the entrance test process with a specific end goal to make it simple notwithstanding for the individuals who don't know natural about hacking methods.

Yao Wen Huang et al (2003) projected techniques for internet application security assessment mechanisms so as to spot poor committal to writing practices that render internet applications at risk of attacks like SQL injection and cross website scripting attacks. They used software system testing techniques (including dynamic analysis, recording equipment testing, fault injection, and behavior monitoring), for testing the protection in internet applications.

Yao Wen Huang et al (2004) described a sound and holistic approach to make sure internet application security that views internet application vulnerabilities as a secure info flow downside. They conjointly created a lattice based mostly static analysis algorithmic rule derived from kind systems likewise as kind state and self-addressed its soundness. city Wassermann and Zhendong (2008) mentioned the static detection ways planned by them for locating cross website scripting vulnerabilities. They provided a static analysis methodology for locating XSS vulnerabilities that directly address weak or absent input validation. Their approach combines the techniques for tainted info flow with string analysis. Moreover, they conjointly offer an in depth analysis that finds each celebrated and unknown vulnerabilities in world internet applications.

Yusuke Takamatsu et al (2010) projected a system for automatically discovers session complex vulnerabilities in web applications. They have experienced this method using an attack simulator that performs a real session complex attack and checks whether it is victorious or not. However, most of the workings either paying attention only on restraint management or security management. In this research work, symbolic restriction with constraint satisfaction and dissemination using rules has been projected and implemented for given that effective security.

Wei Xu et al (2005) introduced a bound together way to deal with counteract that endeavors some scope of programming vulnerabilities which are accounted for as of late. Their examination comes about show that 20% of the vulnerabilities were named Denial of Service (DoS) attacks, 30% are because of outline mistakes, and practically everything else is because of usage blunders. In any case, the majority of the current web attacks have been done either by utilizing cross website scripting or by SQL injection. This examination work gives new systems to distinguish and anticipate cross site scripting and SQL injection attack successfully by executing them at the program and in addition server side.

EXAMPLE OF XSS ATTACK AND HOW TO PREVENT:

In web applications XSS attack is a type of script or code injection attack. XSS attacks take place when an attacker uses a web application to send malicious script, generally in the form of a browser side script, to a different end user. In this paper taking example how to prevent XSS attack using ASP.NET MVC web application. Generally to handle XSS attack a security test required on web application instead of doing manual check web vulnerability scanner used and it is a automatic vulnerable

scanner and one of the vulnerability scanner is Acunetix web vulnerability scanner.

MVC application has inbuilt mechanism to keep away from XSS attack , when ever end user trying to insert script or malicious code into MVC application it will display annoying alert.

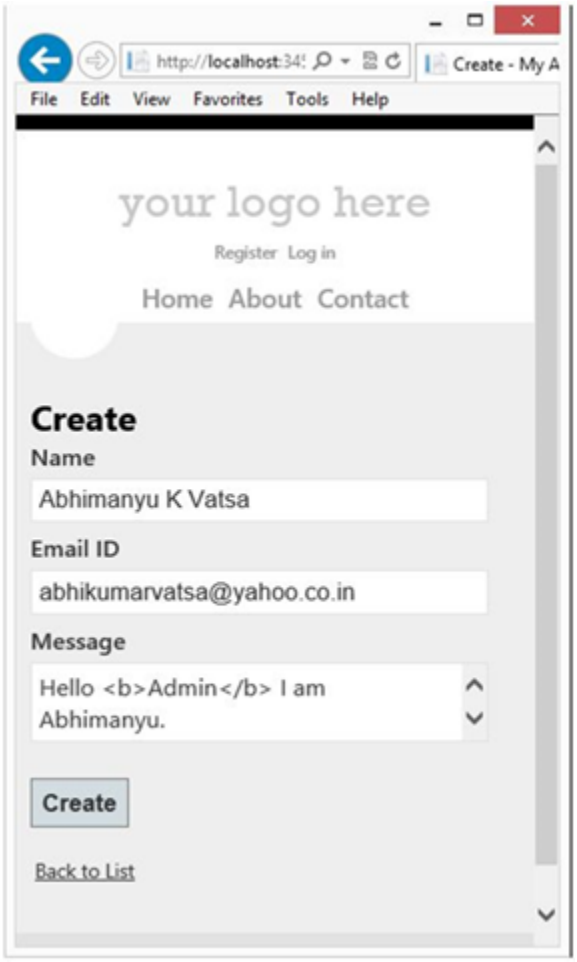
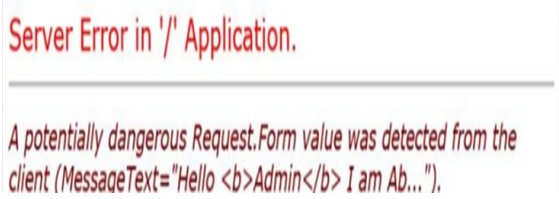


Fig1: Login page

In above web page when user inserting some text and along with he also inserting html markup but mvc application by default it ignores it so it shows below error message to end user.



RESULT AND DISCUSSTION

Here result is comparing between ASP.NET web application and ASP.NET MVC web application. In ASP.NET when developer designing web application in order to prevent script injection attacks such as XSS or HTML injection predefined validates or filters are presented. And whenever coming to ASP.NET MVC web application in built mechanism available for script injection attacks.

when am creating Fig1 using asp.net then it will show below message.

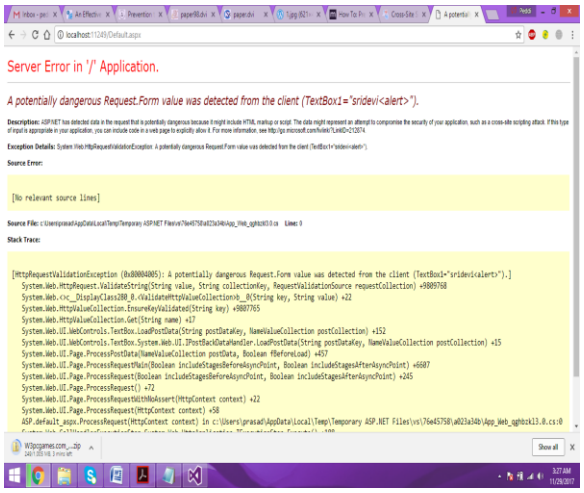


Fig 2; script injection attack.

Whenever comparing Fig1 & Fig 2 if using ASP.NET MVC has inbuilt mechanism to prevent basic XSS attacks where as in ASP.NET developer should add predefined filters.

In order to test web applications most of the technologies providing predefined filters or inbuilt mechanisms, but these mechanisms are limited to few and basic script injection attacks when ever coming to cookie theft attack, Sql injection attacks, Broken authentication and insecure direct object reference etc these kind of vulnerabilities there is no predefined filters or mechanisms are proposed. But some kinds of vulnerability scanners are proposed, for this further enhancement taking different scanners and their functionalities will discuss.

CONCLUSION:

In this paper studied about different security attacks to web applications and taken review from different authors and how they are prevented vulnerabilities and in era of web 2.0 most of the web applications have different major threats in order to understand and develop new mechanism for preventing vulnerabilities this review is helpful. And compared ASP.NET web application with ASP.NET MVC web applications.

REFERECES:

1. *Adam, Kieyzun., Philip J. Guo., Karthick, Jayaraman., Michael, D. Ernst. "Automatic Creation of SQL Injection and Cross Site Scripting Attacks", Proceedings of the IEEE 31st International Conference on Software Engineering, IEEE Computer Society, Washington, DC, USA, pp. 199-209, 2009.*
2. *Chengying, Mao. "Experiences in Security Testing for Web Based Applications", Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, ACM, Seoul, Korea, pp. 326-330, 2009.*
3. *Christopher, Kruegel., Giovanni, Vigna. "Anomaly Detection of Web Based Attacks", Proceedings of the 10th ACM Conference on Computer and Communications Security, ACM, Washington, DC, USA, pp. 251-261, 2003.*
4. *David, Scott., Richard, Sharp. "Developing Secure Web Applications", IEEE Internet Computing, Vol.6, No. 6, pp. 38-45, 2002.*
5. *Thorsten, Holz., Simon, Marechal., Frederic Raynal. "New Threats and Attacks on the World Wide Web", IEEE Security and Privacy, Vol. 4, No. 2, pp.72-75, 2006.*
6. *Yao, Wen Huang., Fang, Yu., Christian, Hang., Chung Hung Tsai., Der Tsai Lee, Sy Yen Kuo. "Securing Web Application Code by Static Analysis and Runtime Protection", Proceedings of the 13th International Conference on World Wide Web, New York, USA, pp. 40-52, 2004.*
7. *WhiteHat Security, "WhiteHat website security statistic report 2010."*
8. *T. Jim, N. Swamy, and M. Hicks, "Defeating script injection attacks with browser-enforced embedded policies," in WWW '07: Proceedings of the 16t*
9. *Y.-W. Huang, S.-K. Huang, T.-P. Lin, and C.-H. Tsai, "Web application security assessment by fault injection and behavior monitoring," in WWW'03: Proceedings of the 12th international conference on World Wide Web, 2003, pp. 148–159.*