iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** VI **Month of publication:** June 2019

DOI: http://doi.org/10.22214/ijraset.2019.6066

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

Web Application Vulnerability Scanner

Vijay Bhagwan Mahajan[1], Mayank Garg[2], Mayank Garg[3], Mazhar Momin[4], Prof. Asmita R. Kamble[5]
[1, 2, 3, 4]*Student,* [5]*Professor, Department of Computer Engineering, Sinhgad Institute of Technology and Science, Narhe, Pune, India*

Abstract: ***In today's world the field of cyber security is a hotline. As the digital world is growing to its mass extend the security related to it is also on the peak verge of exploitation and vulnerable. So the domain of Computer security, cyber security has become a major topic to be concerned about.***
The mass cyber attacks have been detected and has done a great loss of money , capital and various type of resources in that domain .There are very few prevention techniques that are implemented when the model is designed on a big platform and these are not maintained at all. The paper describes a technique that is used for detecting vulnerability (XSS) in the web applications that are poorly constructed and can be easily targeted to a system attack and possible exploitation.
Keywords: XSS –Cross Site Scripting, exploitation, Automation, OWASP

I. INTRODUCTION

Computer security , cyber security , or IT security is the wall of computer systems from any type of vulnerabilities or possible exploitations. But is it that much strong to safeguard the domain is the main question. The field is of growing importance due to increasing reliance on computer systems, the Internet and wireless networks such as Bluetooth and Wi-Fi, and due to the growth of "smart" devices, including smart phones, televisions. Due to its complexity, both in terms of politics and technology, it is also one of the major challenges of contemporary world.The paper discusses vulnerability scanner that will find vulnerabilities in web applications.

The web application now a days are more poorly maintained and are more vulnerable to basic vulnerabilities that can cause a high amount damage to the system .To prevent data leaks through the vulnerability and prevent data misuse the application will provide a safety measures through the scanner .

To ensure safety of the system and web applications from such vulnerabilities It is beneficial to the systems to scan through all the vulnerable blocks that are present. The basic idea is to find vulnerability so that further developing bodies can correct the vulnerability so no data misuse and mislead can occur.

In this paper studies discusses in second section which is literature survey then in third section we thoroughly go into system requirement specification in which we discussed all aspects of the proposed system.

II. LITRATURE REVIEW

A. Extenuating Web Vulnerability with a Detection and Protection Mechanism for a Secure Web Access

In this paper it is described that Web Application Security is a serious issue like network security and it cannot be neglected. It also deals with the topic of technological growth , the threats and the awareness and readiness to deal with them . According to the latest revision of OWASP on July 15, 2016, the top most three web attacks are Injection , Broken authentication and session management, XSS Attacks i.e., Cross-site scripting attacks. Cross-site scripting attacks are a leading online threat. The aim of this attack is to exploit vulnerabilities in the websites which the victim visits. By compromising legitimate websites with malicious content that can capture keystrokes and record user's login information and password. If the login information and password are captured, then the personal data could be compromised.

It mainly informational on three type of XSS attacks that are as follows . Cross-site Scripting can be classified into three major categories :

1) *XSS Attack - Stored (Type I):* The most vulnerable type is Stored XSS or Type 1 XSS ie. Persistent XSS. This type of XSS attacks involves an attacker who inject a script which can be mentioned as the payload that is eternally stored on the web application that is within a database. The classic example of stored XSS is a malicious script inserted by an attacker in a blog or forum. Whenever a victim goes for a surf to the affected web page in a browser the XSS payload will be served as part of the web page. This means once the page is viewed in a browser the victims will inadvertently end-up executing the malicious script.

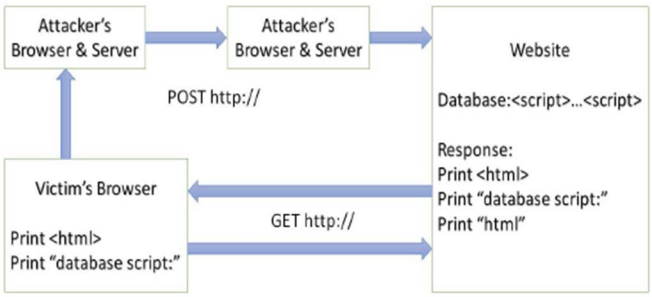


Figure 1 : Scenario for Stored XSS Attack

2) *Reflected XSS Attack (Type II):* The second type of XSS attack is also known as Type II attack is the Reflected XSS attack. In Reflected Cross-site Scripting, the invader's script is a part of the HTTP request is sent to the server and reflects back the response that includes the payload from the request. Using malicious emails and other social engineering methods, the invader traps the user to unintentionally make a request to the server which includes the payload and results in executing the script that gets reflected and executed in the browser. The Reflected XSS is not a like the persistent attack hence the attacker has to carry the payload to each user's social networks.

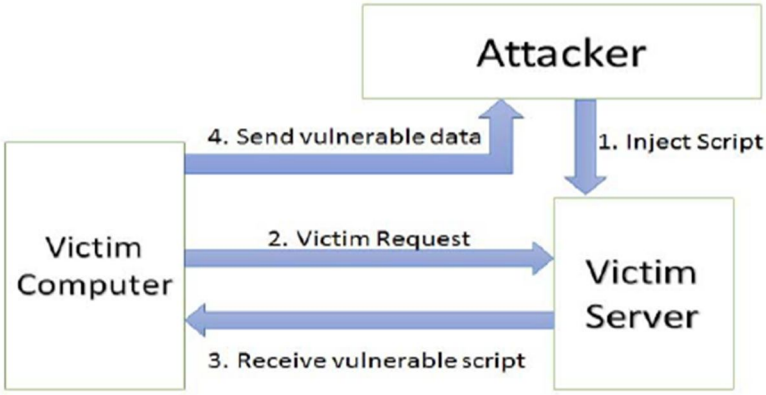


Figure 2 : Scenario for Reflected XSS Attack

3) *DOM Based XSS Attack (Type 0):* DOM-based XSS is also called as Type 0 of XSS attack, DOM-based XSS attacks are executed on the client side. Attackers are able to collect sensitive information from the customer's computer. The data can be breached from the DOM by the web applications and sent to the browser. If suppose the data is not handled properly, an attacker can inject a malicious code, which will be deposited as a fragment of the DOM and executed from the DOM[5], when the user make an attempt to read. In this way DOM based XSS makes the detection more difficult.

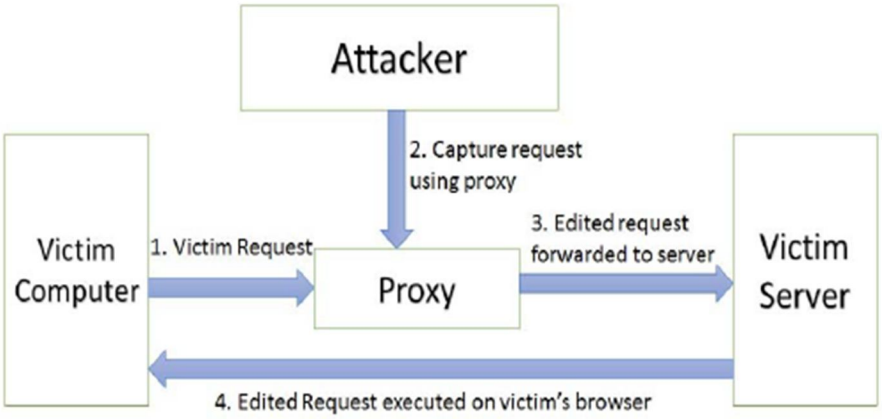


Figure 3 : Scenario for DOM based XSS Attack

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

The next main it deals with the security proxy systems which is discussed as follows.

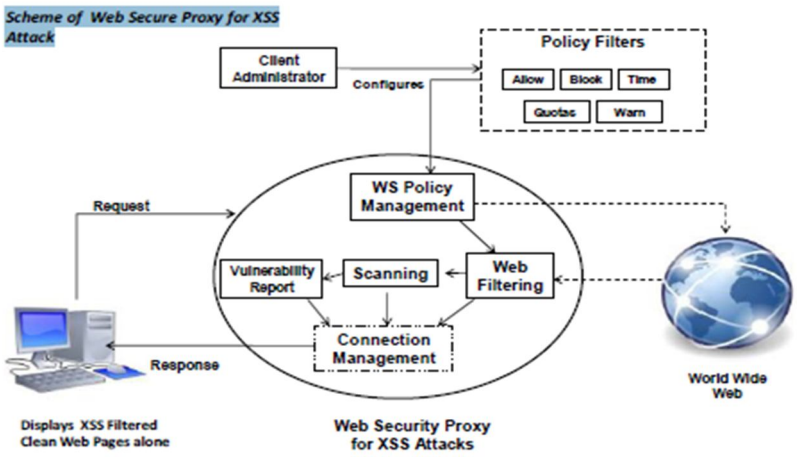


Figure 4 : Web Security Proxy for XSS Attack

The paper also explains about the web security Proxy for XSS attack and how can it be vulnerable . It also talks about the WSP(Web Security Proxy) , Policy filters , WS policy management , malevolent reports etc .

B. XSS Vulnerability Assessment and Prevention in Web Application

In this paper there is more detailed information on three types of XSS . There are three types of Cross-site Scripting attacks: persistent or stored, non-persistent or reflected and DOM-based. In
DOM based and Non-persistent attack the attackers require a user who visits their malicious web page or click on a malicious link. The HTTP request posts automatically without the knowledge of the victim if the application sends the request using POST method only. In this case, vulnerable code will be submitted easily through request . Upon submitting the malicious form or clicking on the malicious link, the XSS code will get executed again and it will get displayed on user's browser form. In Persistent attacks attacker injects when malicious code using the web form and it is stored in the database. Example message board, review form, webmail messages, web chat software and any user input form. The unsuspicious user is not necessarily need to interact with any additional link, he can simply view the web page containing code.
The paper assessment and result are based on the study of previous research and their implementation in the real scenario. They have tested many web applications and found high priority cross site scripting issues. Although they have many approaches available now to prevent XSS, but many web applications are still vulnerable.
In existing approach normally developers user java script validation or user input filter to prevent client side malicious inputs, for the output they use escape method and sanitation method. Some more secure web applications use application level firewall to filter user request. But still, not able to stop XSS attacks.
In the paper it is mentioned that it is not sufficient in the prevention of more dangerous XSS payloads. The use of one or two existing approaches can stop the direct malicious inputs from the web browser, but not strong enough to handle middleware attacks.
It suggests the use of strong application level firewall, use of HTTPOnly cookie flag, use of proper security testing and also the use of scanner tool to detect XSS payloads in all parameters, headers and path, use of strong SPRING tool to develop java application and use of updated web browser.
The proposed system help developer in the handling of XSS issues and it also help tester in detection. We suggested this hierarchical approach because a single level of security is not sufficient in the prevention of cross site scripting issues.

C. WebGuardia – An Integrated Penetration Testing System to Detect Web Application Vulnerabilities

In this paper, it presents WebGuardia which crawls through a given target web application and detects web application vulnerabilities identified by OWASP . The types of vulnerabilities that WebGuardia is able to detect are SQLI, XSS, Invalidated Redirects and Forwards, Insecure Direct Object References and Security Misconfigurations. This tool uses a simple architecture which involves crawling, attacking, analysing and reporting modules.

1) *SQL Injection Vulnerability:* If an attacker is able to insert unauthorized SQL statements into an existing SQL query of a web application (while injecting malicious input into user fields that are used to generate the query), that web application is vulnerable to SQL injection attacks. Detailed discussion regarding these types of SQLI vulnerabilities has been conducted .
2) *Cross-site Scripting (XSS) Vulnerability:* Apart from malicious JavaScript codes, malicious VBScripts, ActiveX, HTML or even Flash can be embedded into vulnerable web applications by exploiting XSS vulnerabilities . Two types of XSS attacks can be observed in modern day exploitations and they are namely, reflected XSS attacks and stored XSS attacks . During reflected XSS attacks, malicious script is immediately reflected to the victim by the server and during stored XSS attacks, the malicious script is stored inside the vulnerable application for later retrieval . The proposed tool focuses on detecting reflected XSS vulnerabilities in the target web applications.
3) *Vulnerabilities due to Invalidated Redirects and Forwards:* This type of vulnerabilities occur when untrusted input that is capable of redirecting the request to a URL, is accepted in a web application and this type of vulnerabilities can be exploited to launch a successful phishing scam .
4) *Vulnerabilities due to Insecure Direct Object References:* These types of vulnerabilities occur when direct access to information is provided to the end user and by exploiting this type of vulnerability, the authorization of information can be violated.
5) *Vulnerabilities due to Security Misconfigurations:* Security misconfigurations can occur at platform, web server, application server, database, framework and custom code . Vulnerabilities due to security misconfigurations can be exploited by accessing default accounts, unused pages, unpatched flaws, unprotected files and directories to gain unauthorized access and knowledge of the system being used . In this paper, they have presented WebGuardia, an integrated penetration testing system to detect five of the top ten web app vulnerabilities that exist in modern day web applications. Two new approaches have been proposed in this paper to detect two types of vulnerabilities and already existing approaches have been highly modified in order to enhance the performance of the system. On completion of the tests, a report is generated along with a user friendly graphical overview of the detected vulnerabilities which can accordingly be used to find in detailed information regarding the executed attacks. WebGuardia has been tested on several web applications and against over 700 web pages. The results indicate that, even though it is technically infeasible to completely avoid generation of false positives and false negatives, by using WebGuardia and proposed approaches, generation of false positives and negatives can be kept at a minimum level.

D. Detecting Cross Site Scripting Vulnerabilities Through Automated Unit Testing

In summary, the paper proposes a unit testing approach to detect cross-site scripting vulnerabilities caused by incorrect encoder usage. This approach can be easily integrated into existing software development practices and can pinpoint the location of a vulnerability in the source code. It can help developers find and fix XSS vulnerabilities early in the development cycle, when they unit test their code, without involving security experts. The grammar-based attack generation is a structured way to generate XSS attack strings. We were able to generate tests for vulnerabilities missed by popular attack repositories. More importantly, our grammar models can be modified to cover unknown or new attack scenarios. For example, a new version of a browser may offer new ways for attackers. Our approach also has low false positive rates. Finally, our evaluation demonstrates that our approach is computationally efficient and can detect vulnerabilities cannot be found using black-box fuzzing systems. This work can be extended in a number of ways. We are currently evaluating our approach with more open source applications. We also plan to look at handling recursive structures in the attack grammar as well as improve the efficiency of attack evaluation.

III. PROPOSED SYSYTEM

A. System Architecture

The basic web application vulnerability scanner architecture in that implemented is a web application where the user first registers if a new user or logins with the credentials .The credentials are for proper authentication. After that the user provides input to the scanner in the form of a website URL which is to be checked for any vulnerability . Firstly URL is taken as a input from user. Then that URL is crawled for all its respective pages. Now the pages collected are of various types some may have parameters some may not. So the cleaning of pages is done, only the pages falling under certain criteria are taken and rest are discarded. These are some simple criteria like if the page has parameter or not, redundant pages, if the pages are different but have similar parameters only one copy of page is taken and rest are discarded.

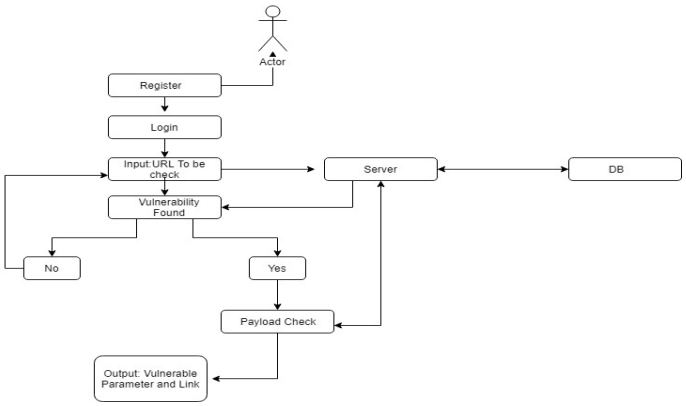


Figure 5: System Architecture

Then a list of payloads is tested on every parameter of every page selected. In this payload testing through an automated popup detection every page/parameter with a payload is executed on browser. And page which give popup stored separately for next step of process.

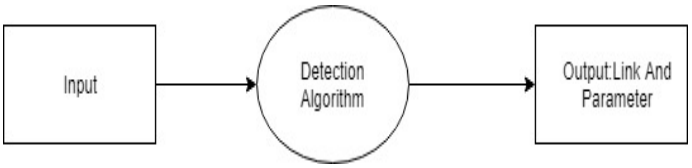


Figure 6: Data Flow diagram

B. Hardware Requirements

Basic Hardware Requirement are i5 6 Generation processor ,8 GB RAM , 1 TB internal Storage device compatible computer which can be used for such scanner . The basic Software Requirement is knowledge about Web application vulnerability Scanner, Any Browser , Burp Suite.The best search results will be opted on faster computer therefore it needs high end hardware specification .

C. Web Applications Scanner

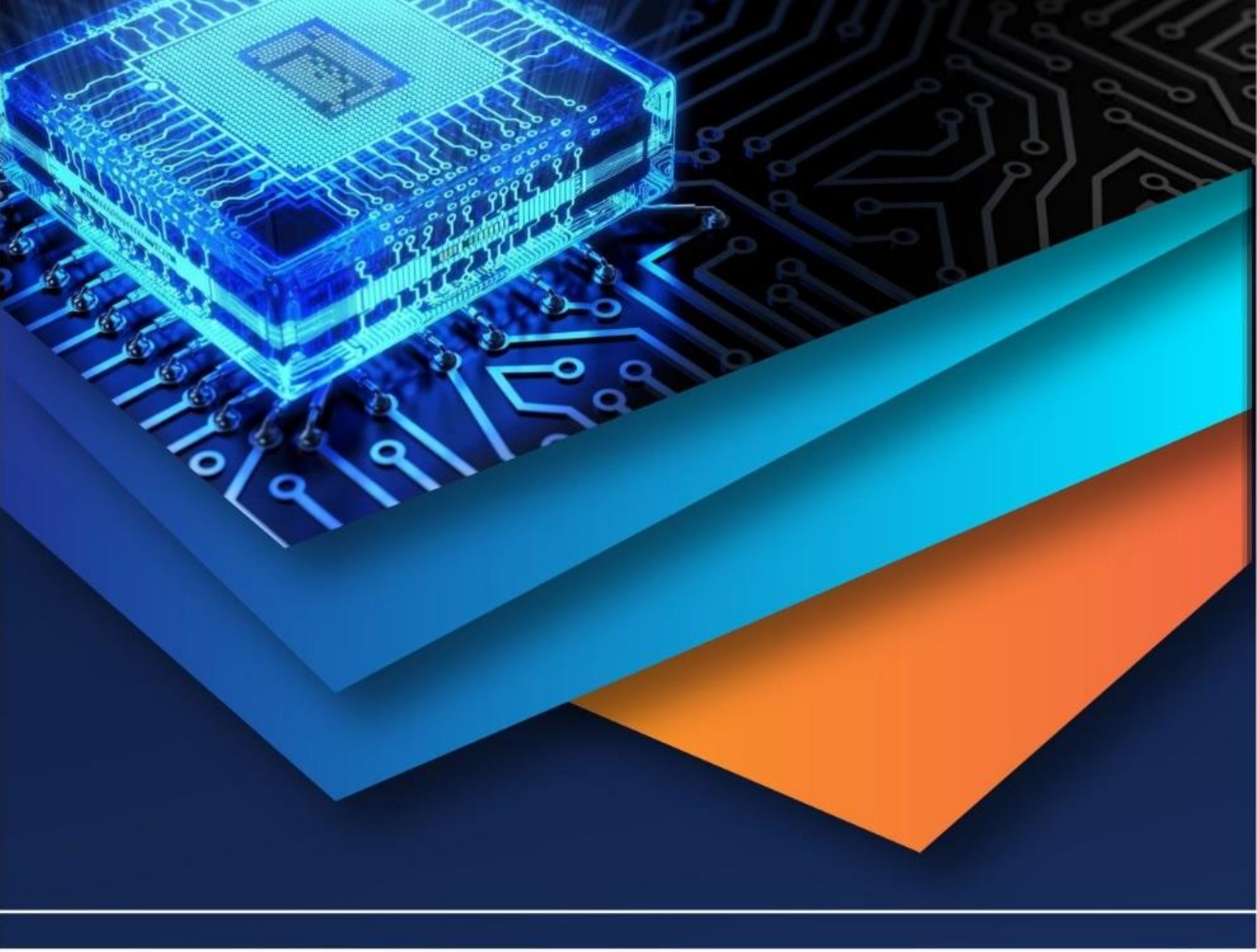
The basic idea for development of scanner is to see to it that the vulnerability that affects the system most must be detected with less efforts of the user .All the work is done on backend so there is not much for the user to do .It may be seen that the process is time consuming but distributed modular platform makes it faster and more prone to find vulnerabilities in less amount of time.

IV. CONCLUSION

In this paper we can conclude that a robust , modular web application vulnerability scanner can be developed that crawl and test at the same time and can find vulnerabilities in different web applications implicating the wide usage for safety and security .

REFERENCE

[1] K. Vijayalakshmi and Dr. A. Anny Leema, “Extenuating Web Vulnerability with a Detection and Protection Mechanism for a Secure Web Access,” IEEE conference (ICSCN -2017), March 16 – 18, 2017, Chennai, INDIA
[2] Ankit Shrivastava and Ashish Kumar, “XSS Vulnerability Assessment and Prevention in Web Application” 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16 October 2016.
[3] Nisal Madhushan Vithanage and Neera Jeyamohan, “WebGuardia – An Integrated Penetration Testing System to Detect Web Application Vulnerabilities” IEEE WiSPNET 2016 conference.
[4] Mahmoud Mohammadi , Bill Chu and Heather Richter Lipford , “Detecting Cross-Site Scripting Vulnerabilities through Automated Unit Testing” IEEE WiSPNET 2014 conference.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY