Constructing Secure Web Applications With Proper Data Validation

Authors
Dr. Anil Kumar and Krishna Reddy
Summarized by
Fariza Aqila Zulkifly
Faculty of Computing & Informatics
Multimedia University, Malaysia

Abstract

The rise of technologies and innovations around the world in the information technology field has made web applications prone to attacks if their security level is low. Web applications are programs that are supported by languages such as JavaScript, PHY, Phyton, HTML and many more that runs in a web browser. Among the benefits of web applications are easy to update and perceive to reduce cost. However, web applications with poor input validation are at risk to be exploit. One of the solution to create secure web applications is through proper data validations. Insertion of correct syntax, characters, digits, spell checks are part of the correct data validations.

1 Problem Solved

In this article, data validation vulnerabilities in web applications are the main problem. There are seven vulnerabilities mentioned which are Cross Site Scripting, Command Injection, XML Injection, Code Injection, Server Side Includes Injection, XPATH Injection and SQL Injection. The most frequent application attack is Cross Site Scripting or XSS. Victims browser will be attack. XSS will launch bad script into browsers. Modifying source code leads to major attack in XSS. Attackers are able to retrieve passwords from users as malicious scripts are being uploaded to service. Code Injection occurs when service request is changed. Attackers are able to inject code into HTML pages to perform remote code execution without having full-fledged connection with client or sever side scripting languages by just add a piece of dynamic code inside static HTML page. Access is granted in an unauthorized manner that will able to corrupt the whole document when XPATH is applied to data that stored in XML format. Data driven applications have been targets of the SQL injection where malicious SQL query is inserted to bypass authentication by passing true values to service. However, SQL injection is not a database vulnerability. The problem is how to produce secure web applications with proper data validations.

2 Claimed Contributions

This research introduces ways to insert proper data validations to provide secure application for users. Strict validations in server side must be maintained. Its a better solution to even sanitize. Sanitization is the best option in case of complexity. The advantage of having secure web applications is to provide reliable and secure communication, having lesser complexities to prevent attack patterns on a system.

3 Related Work

When security threats arise, a lot of consequences might be faced by the users. According to the vulnerabilities listed, each of them has their impacts on the users if not taken care of. It started by

the application data sends malicious codes via service request or along with user request headers. Not only security at the users computer breaches but data theft is also possible. For example, in an email sent by some random user, without even noticing the person receiving the email might have pressed something that may lead for information to leak. Information such as name, address, telephone number and many more may be obtained with just one single click. Service unavailability and service redirections can also occur. Some unsecured web application in browser may cause the web page to change after the user inputs their information to steal their credentials. Information are bound to leak by Server Side Include Injection. This is because SSI allows us to enter the code in HTML page hosted in web server. By allowing system command being executed, security threats will be face by the users. An attacker can retrieve large amount of data from database with SQL injections. Database details can be retrieved by passing authentication because the system did not ask for any username and password. Data validation vulnerabilities have cause a lot of effects to the users.

4 Methodology

The Method used is Theoretical. Web applications are mainly deals with request and response to process information with webserver. Security is highly needed in the web application business so that they won't loose integrity and maintain their confidentiality. Using principle associated and some knowledge, the Authors had systematically gave all the research details in this articles.

5 Conclusions

Vulnerabilities are major issue we often face these days. In this paper, vulnerabilities that are mentioned such as Cross Site Scripting (XSS), Command Injection, Code Injection, XML Injection, Server Site Injection (SSI), XPATH Injection and SQL Injection can be treated and mitigated. With proper precautions that have been proposed in this paper, users can protect their computer from being infected by unwanted threats. Data validations and input sanitizations are the main solution to secure a web application. A web application that has less complexity will likely be more secure. Applications that are more complex have to think about the secure communication to survive application in the web. Focusing in secure web application development strategies will be the future plan to protect the user and provide a trusted and secure communication.

6 Future Works

Web applications have become part of our daily lives. We use web applications to complete our tasks most of the time. Billing, presentation, file sharing, online education are just a few categories of web applications. For instance, PayPal application which is an online payment app that allows users to make payment using their PayPal account or credit cards for any service or items users bought online. The reason why web apps are so popular is because it is for business mobility. Increasing complexity of an application will make the developer consider the secure communication for an app to survive. Precaution steps must be taken before developing an app. Firstly, security awareness must be integrated to the development team at early stages so secure software development life cycle (SSDLC) will fulfil. Next, system must be updated and log analysis is conducted to prevent malicious users. The network status of system should be monitored occasionally to analyse request patterns. To prevent system crashes, load and stress on system must be controlled. If validation threats continue to arise, data sanitization is preferred.

7 References

Gary McGraw and John Viega, "Building Secure Software: How to Avoid Security Problems the Right Way", Addison-Wesley Pub Co, ISBN 020172152X.

Dafydd Stuttard, Marcus Pinto, "The Web Application's Handbook Discovering and Exploiting Security Flaws", 2008, Wiley, ISBN 978-0470-17077-9.

Mike Howard and David LeBlanc, "Writing Secure Code", Microsoft Press, ISBN 0735617228. Gary McGraw and Greg Hoglund, "Exploiting Software: How to Break Code", Addison-Wesley Pub Co, ISBN 0201786958.

Asmawi, A ; Aflendey, L.S. ; Udzir, N.r. ; Mahmod, R., "Model-based system architecture for preventing XPath injection in database-centric web services environment", Computing and Convergence Technology (ICCCT), 2012, Page(s): 621-625.

Johari, R. ; Sharma, P., "A Survey on Web Application Vulnerabilities (SQLIA, XSS) Exploitation and Security Engine for SQL Injection", Communication Systems and Network Technologies (CSNT), 10.11 09/CSNT.2012.10 4, Page(s): 453-458.

Jeremiah Grossman, Robert "RSnake" Hansen, Petko "pdp" D. Petkov, Anton Rager, Seth Fogie, "Cross Site Scripting Attacks: XSS Exploits and Defense", 2007, Syngress, ISBN-IO: 1-59749-154-3. Joel Scambray, Mike Shema, Caleb Sima, "Hacking Exposed Web Applications", Second Edition, McGraw-Hili, 2006 - ISBN 0-07226229-0.

Atashzar, H. ; Torkaman, A ; Bahrololum, M. ; Tadayon, M.H., "A survey on web application vulnerabilities and countermeasures", Computer Sciences and Convergence Information Technology (ICCIT), 2011, Page(s): 647-652.

James S. Tiller, "The Ethical Hack: A Framework for Business Value Penetration Testing", Auerbach, ISBN 084931609X.