International Journal of Computer Applications (0975 – 8887)
Volume 173 – No.4, September 2017

Some Basic Steps to Secure Web Application from Unauthorized Access

H. M. Mohidul Islam
Department of Computer Science
American International University-Bangladesh
Banani, Dhaka 1213, Bangladesh

ABSTRACT

Nowadays people are being more dependent on web application because of its accessibility. The web application is expected to play a significant role in internet applications. A web application can be accessed from anywhere by any person of the world through internet. People use web application for their own purpose through a web system. But it has some probability to be hacked or data stolen by an unauthorized person. There are some common vulnerabilities of a web application. If it is possible to prevent the attack and then web application will be secure to protect user's information. And then the user also will get the best productivity of web application and they will increase interaction with the web based system. So it is very important to provide a secure web platform. In this paper, some basic steps have been proposed to secure web application form unauthorized access.

General Terms

Web Technology, Information Security.

Keywords

Web application Security, Prevent unauthorized access, Access Control.

1. INTRODUCTION

The web application needs data from the user to provide service. The application developer needs to make sure that user data will be secure and well protected. For that reason, it is important to provide a security for authorized access in their web application. If an unauthorized person gets access to an authorized person data this will be unethical and important information will be stolen, which can be a risk for a valid user of the application. If this happens then the developer will be responsible for that. So it is a developer's deeds to make sure a secure application. If the user gets full safe cloud storage of data, they can get maximum utilization of web and they will be safe from cybercrime.

2. BACKGROUND STUDY

According to a Symantec report released earlier, 2003 reported that the total number of Web application vulnerabilities discovered in 2002 was 178% higher than in 2001 that 95 percent of these were remotely exploitable, and that was considered highly or moderately severe [1]. But a developer can prevent them by securing code at the time of developing application. David Scott & Richard Sharp [2] have asserted that types of attacks are form modification, SQL Attacks and cross site scripting. Flow-insensitive information flow tracking analysis developed by Monics S.Lam, Michael Martin[3] to find all the vulnerabilities of a web application. Form-modification attacks are possible when Web developer absolutely trust assertions checked only on the client side. SQL attack occurs in back end database when a query will work to manipulate database from client server. XSS comes from the dynamic web site itself through submitting malicious HTML possibly include JavaScript Code. Web hacker mostly tries to inject SQL to get access to a user account. a static context-sensitive. Monics S.Lam [3] suggested Security-aware API, controlling access and database abstraction to secure web application. To prevent SQL attack database abstraction is the security concern. Where many frameworks for web development provide application level security with API and ensure security [2]. On the other research by Yao-Wen Huang, Shih-kun Huang [4], they tried to detect SQL Injection through black box testing, white box testing. Analyzing their detection process, it can be said that through application code SQL injection can be blocked rather than database abstraction. Because with the coding condition or using object variable vulnerability can be easily reduced. Only effective condition need to apply through code. Preventing SQL injection can prevent unauthorized data manipulation and access. Other research paper of Yao-Wen Huang [5] had added general script injection as web application vulnerabilities. Their paper uttered about some web application Security. Scott & Sharp [2] used user input proxy to abstract web application Security.

Securing from unauthorized access session is one of the most important parts of a web application. Session holds information by which application can give an authorized access. But this session also can be hijacked by a network eavesdropper. Ben Adida[7] proposed a method to improve the security which method is known as SessionLock. Over SSL (Secure Sockets Layer) at user login time the web browser get a session secret delivered by a web server. The web browser can use this secret token to authenticate. An attacker cannot generate a valid HTTP request on behalf of an authorized user as attacker limited to capabilities of eavesdropping. Some other research papers proposed the different technique to prevent unauthorized access. Like Shan Jiang,Sean Simth[8] mentioned about SSL which can work over the network to a server through encrypting login information. RA. K. Saravanaguru, George Abraham [9] focused on XML signature and XML encryption. The main function of XML signature is including partial signature, which allows signing of specific tags contained in XML data. But XML encryption is might be better than SSL, Because SSL work through cloud over the network to a server, client and server carry exchange protocol to ensure client and a party which knows the private key then share a semantic key – that will not be possible to know by others [8]. Raluca Ada Popa, Emily Stark [16] presented Mylera which is a platform to build web application on top of encrypted data. Mylar's goal is to help the application admin to protect the confidential data of users Attack usually client via server path [18] through HTTP request. But if an encryption is provided then path become secure.

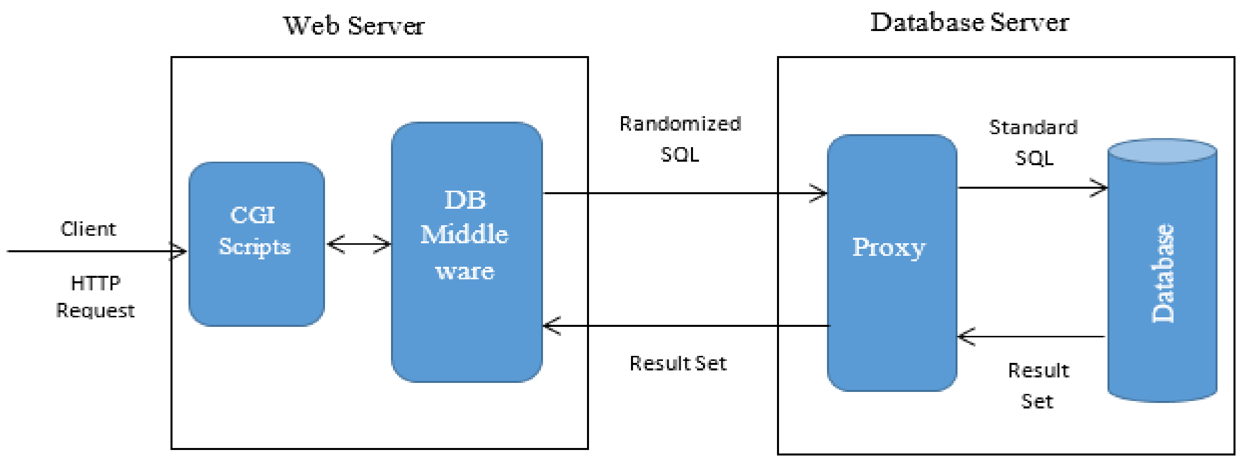

Figure 1: SQLrand System Architecture

3. PROPOSED STEPS

The main objective is to find some way by which a web application will be secure from unauthorized access of data. To find the secure way first we need to get every insecure or weak point of a web application and do action against them. Authentication process also needs to make more strong so that no other can enter in an authorize account without proper verification. If common harmful action against web application is prevented by application developer providing proper action with code that application will be much secure from unauthorized access.

3.1 Secure User Input

The most common element in a web application is input box. Where user provides some data and submits the form for the different purpose. If the user gives the regular data in the input box this will not do any harm in a web application generally. But if the user can give some unwanted data and can submit the form, this may generate any unexpected result. Harmful cross site scripting (XSS) also can implement by input box. To make a secure web application developer must need to make sure system will not take any unwanted data. For that client side validation and server side validating is required. Client side validation can easily be destroyed by a user, so server side validation is the most important for data validating. Web applications security will be increased depending on server side validation. Griffith [6] have asserted that filter data, validate data & escape data before submitting a form are best practice when accepting any kind type of data into web application

3.2 Prevent SQL Injection

Structured Query Language is used to create the relationship with the database. Generally, all data are kept in the database in all web application. So if anyone can inject SQL, he will be able to steal data from the database. SQL inject can be done by SQL and CGI Script [10]. Client request to a web server and web server send randomized SQL to database server [Figure 1]. A developer needs to make sure that randomized SQL will not carry injected SQL. If proxy of database server receives safe SQL, then database server will return expected result. So it is the responsibilities for the developer to prevent SQL injection so that database server returns safe result set. Nowadays in modern web framework [12] [13] use different kinds of query builder to ensure prevention of SQL injection. And there are many different mechanisms available to prevent SQL injection. Almost every web framework provides SQL processing or building injection free query [15].

3.3 Strong Authentication Process

Securing web application from unauthorized access strongly depends on the authentication process. Normally to authenticate every application follow username and password matching with the database. If matching returns true, then authentication become successful and the user can access data of authentic account. Web hacker commonly try to recover password to get an access of an account. For a strong authentication, process developer needs to use a secure method, adding extra conditions to the authenticating query, validate a user's credentials, protecting the route, providing csrf protection etc[17]. The developer also needs to use non-broken password encryption method. If they can use a couple of steps of the encryption process in authentication, this will provide a secure application. Because if anyone try to get password of a user from database, he will get encrypted password which is not readable.

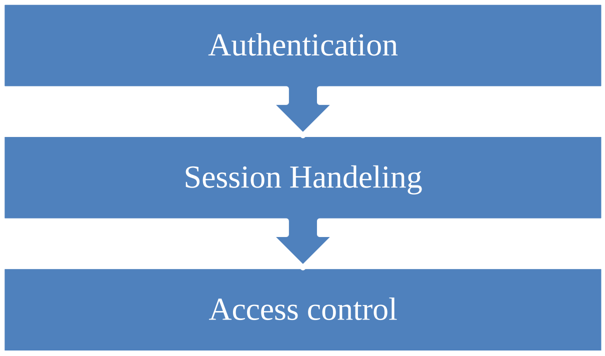

Figure 2: Session Management

3.4 Session Handling

Normally session handle the access control in web application. After authentication server needs to keep session to access logged user’s data. This session also used to determine user group of an application. But session hijacking is one of the most reliable ways for attacker steal others data. After login real user sends requests to the web application using a cookie id for authentication [11]. As this request sent over HTTP, an attacker can eavesdrop the request to capture the cookie id and hijack the session. So developer needs to be concerned about session handling. They can build their web application which will logout a user automatically if the same session remains active more than one. Man-in-the-Middle attacks can be reduced by providing session security [14].

3.5 Login Tracking

User access needs to be controlled after the successful login in an application. Nowadays people love to keep their account logged in those devices they regularly use. So it will very useful if application track logged device and last login time, devices so that valid user can notify about own security and can take further steps to secure their account. The application can also notify about devices which try to log in. For maximum security, the application can provide two steps verification whenever a new device logged in.

4. LIMITATION

The proposal of this paper is only focused on the basic steps. The proposed steps will not guarantee hundred percent security. There are more advanced steps available to secure web application which are not proposed in this paper.

5. FUTURE WORKS

The future goal is to work with advanced steps to secure web application and establish correlation among steps. Evaluating performance benchmark due to implementing steps will also be focused.

6. CONCLUSION

From the viewpoint of objectives, this research is a combination of descriptive research & explanatory research. Observational research method has been followed in this research paper.

The revolution of web application is increasing day by day. And new web technology also inventing each and every year. Adversary always tries to find out weaker point and attack on it. If application developer follows the basic steps this will assure a standard security. They always need to observe unusual behavior and the new attack. They also need to know the weak point and work against them and work with advanced steps to provide maximum security of web application.

7. ACKNOWLEDGMENTS

This work is done under the guidance of DR. DIP NANDI. Author would like to thank SAEF ULLAH MIAH and ASIF-UR-RAHMAN for their support during the tenure of this article preparation.

8. REFERENCES

[1] Higgins, M., Ahmad, D., Arnold, C. L., Dunphy, B., Prosser, Mand Weafer, V., “Symantec Internet Security Threat n Report—Attack Trends for Q3 and Q4 2002,” Symantec, Feb 2003.

[2] D. Scott and R. Sharp, "Developing secure Web applications", IEEE Internet Computing, vol. 6, no. 6, pp. 38-45, 2002.

[3] M. Lam, M. Martin, B. Livshits and J. Whaley, "Securing web applications with static and dynamic information flow tracking", Proceedings of the 2008 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation - PEPM '08, 2008.

[4] Y. Huang, S. Huang, T. Lin and C. Tsai, "Web application security assessment by fault injection and behavior monitoring", Proceedings of the twelfth international conference on World Wide Web - WWW '03, 2003.

[5] Y. Huang, F. Yu, C. Hang, C. Tsai, D. Lee and S. Kuo, "Securing web application code by static analysis and runtime protection", Proceedings of the 13th conference on World Wide Web - WWW '04, 2004.

[6] A. Griffith, CodeIgniter 1. 7 professional Development. Birmingham: Packt Publishing, Limited, 2010.

[7] B. Adida, "Sessionlock", Proceeding of the 17th international conference on World Wide Web - WWW '08, 2008.

[8] S. Jiang, K. Minami, and S. Smith. Securing Web Servers against Insider Attack. In Annual Computer Security Applications Conference, 2001

[9] R. Saravanaguru, G. Abraham, K. Ventakasubramanian and K. Borasia, "Securing Web Services Using XML Signature and XML Encryption", 2013.

[10] Boyd, S. and Keromytis, A., 2004. SQLrand: Preventing SQL injection attacks. In Applied Cryptography and Network Security (pp. 292-302). Springer Berlin/Heidelberg

[11] Dacosta, I., Chakradeo, S., Ahamad, M. and Traynor, P., 2012. One-time cookies: Preventing session hijacking attacks with stateless authentication tokens. ACM Transactions on Internet Technology (TOIT), 12(1), p.1.

[12] Y. HEREN, "Design and implementation of web based on Laravel framework.", Atlantis Press, 2015.’

[13] D. Upton, Codelgniter for Rapid PHP application development. Birmingham [u.a.]: Packt Publ., 2007.

[14] D. Ryck, "Client-Side Web Security: Mitigating Threats against Web Sessions", 2014.

[15] B. Delipetrev and S. Ristova, "Performance benchmark of PHP frameworks with database select methods", IX INTERNATIONAL CONFERENCE FOR YOUNG RESEARCHERS, pp. 38-41, 2015.

[16] Popa, Raluca Ada, Emily Stark, Steven Valdez, Jonas Helfer, Nickolai Zeldovich, and Hari Balakrishnan "Building web applications on top of encrypted data using Mylar", 11th USENIX Symposium on Networked Systems Design and Implementation, 2014.

[17] T. Otwell, "Security - Laravel - The PHP Framework For Web Artisans", Laravel.com, 2011. [Online]. Available: https://laravel.com/docs/4.2/security

[18] Swamidas, M., A. Govardhan, and D. Vijayalakshmi, "QoS Web Service Security Dynamic Intruder Detection System for HTTP SSL services", arXiv preprint arXiv:1605.00918, 2016. Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems.