

A Study on Web Application Security and Detecting Security Vulnerabilities

Sandeep Kumar¹, Renuka Mahajan², Naresh Kumar³, Sunil Kumar Khatri⁴

^{1,3,4}Amity Institute of Information Technology, Amity University Uttar Pradesh, Noida, India

²Jaipuria Institute of Management, Noida, India

¹sandeep5.mahla@gmail.com., ²renuka.mahajan@jaipuria.ac.in

³naresh.dhull@gmail.com, ⁴skkhatri@amity.edu, sunilkkhatri@gmail.com

Abstract — The world is exceedingly reliant on the Internet. Nowadays, web security is biggest challenge in the corporate world. It is considered as the principle framework for the worldwide data society. Web applications are prone to security attacks. Web security is securing a web application layer from attacks by unauthorized users. A lot of the issues that occur over a web application is mainly due to the improper input provided by the client. This paper discusses the different aspects of web security and its weakness. The main elements of web security techniques such as the passwords, encryption, authentication and integrity are also discussed in this paper. The anatomy of a web application attack and the attack techniques are also covered in details. This paper explores a number of methods for combatting this class of threats and assesses why they have not proven more successful. This paper proposes a better way for minimizing these type of web vulnerabilities. It also provides the best security mechanisms for the said attacks.

Keywords — Web Security Threats, SQL Injection, Cross Site Scripting, Security Standards, Security Vulnerabilities.

I. INTRODUCTION

Web security is an important aspect for web applications. Today web security is a real concern related to the Internet. It is considered as the principle framework for the worldwide data society. Web applications provide a better interface for a client through a web page. The web page script gets executed on client web browser.

Web applications are a main base of attacks such as cross-site scripting, cookie-session theft, browser attack, self-propagating worms in web email and web sites. These types of attacks are called 'injection attacks' which attacks by the use of malicious code. Injection attacks have commanded the highest point of web application vulnerability lists for a significant part of the previous decade.

There are two most common security vulnerabilities today: SQL injection and cross-site scripting [11], [12],[16],[21]. A security evaluation of application defence center, which had more than 250 e-commerce applications, online banking and the corporate sites came up with a statement that more than 85% of web applications are vulnerable to attacks [2], [10],[15],[22],[23].

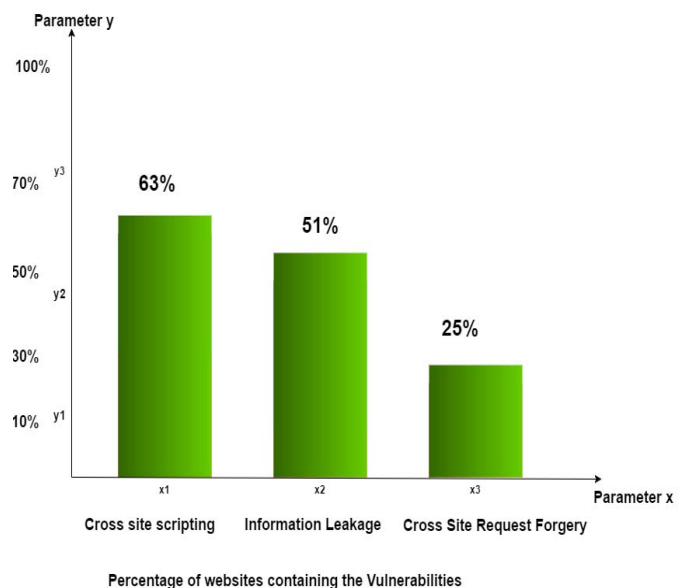


Fig. 1. Percentage of websites containing the vulnerabilities

II. LITERATURE REVIEW

The main issue in web security research is in enabling a user a safe and trusted platform for communication with the web application. But some people continue to do business with insecure site. Some organizations or companies don't want to reveal the information about their own security holes. So, it's very hard task to get the reliable information about the state of web security today [1],[3][14],[18][21].

There are two common important security vulnerabilities today: SQL injection and cross-site scripting. These types of vulnerabilities directly affect web servers, application servers, and web application environment. [13],[4],[5],[7],[20].

OWASP in this paper explores a number of Table 1: Reasons for Attacks [2] methods for detecting threats and assess why they have not proven more successful. A better mechanism for minimizing such type of web vulnerabilities is proposed in this paper. Currently, there are many privacy risks in web applications. Today too many websites are hacked by anonymous people. They target website because of different types of reasons. They are mentioned in table 1.

Attack Goal	%
Stealing Sensitive Information	42%
Defacement	23%
Planning Malware	15%
Unknown	08%
Deceit	03%
Blackmail	02%
Link Spam	03%
Worm	01%
Phishing	01%
Information Warfare	01%

Fig. 1. OWASP focussed on identify some vulnerability for the broad array of the organization [2],[10],[11],[23],[19].

III. PROPOSED ARCHITECTURE AND FRAME FOR DETECTING SECURITY VULNERABILITIES

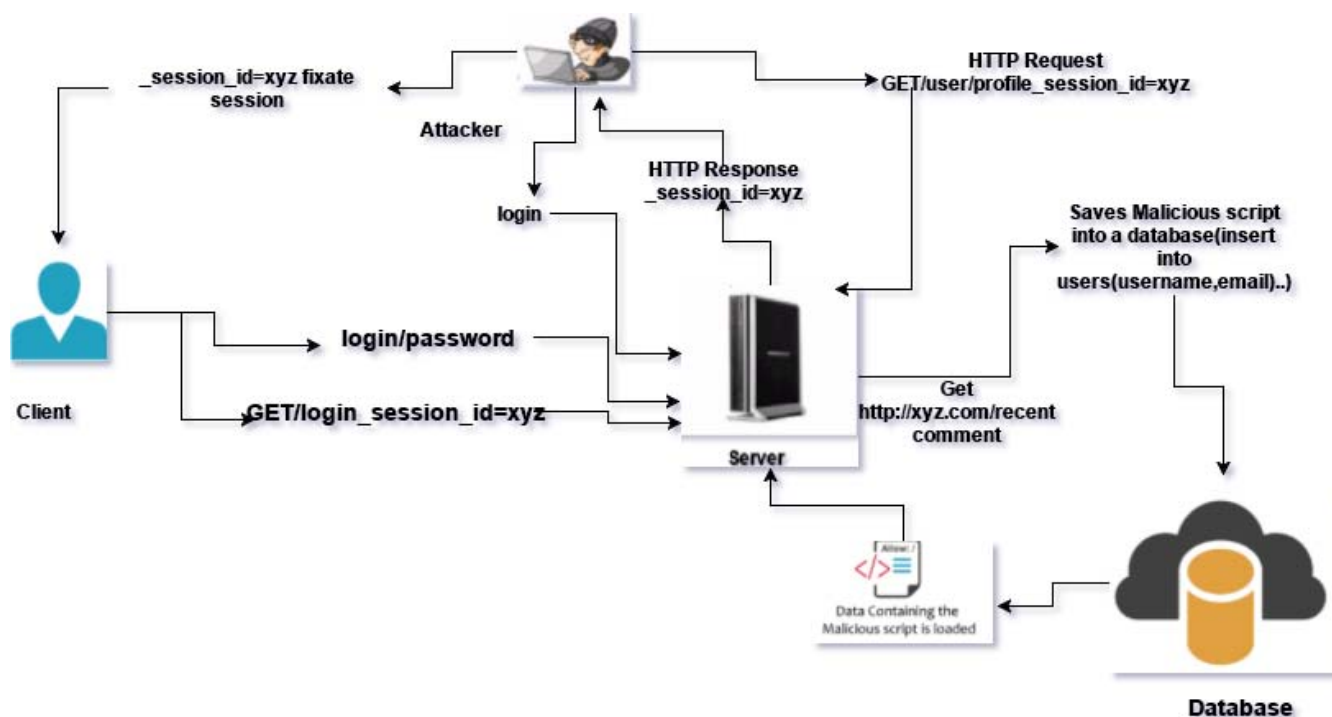


Fig. 2. Website In Vulnerability

The literature survey study of different vulnerabilities define that two common security vulnerabilities are SQL injection and XSS. Many technology and mechanism are proposed by different researchers to prevent the SQL and XSS attack. During our research we have found out that SQL injection attack and XSS attacks are still possible even after implementing preventing mechanism presently available in the market, and so provide a preventive mechanism we have proposed the architecture shown in figure 2. In this figure we

have proposed a scheme through which we will attack any website with SQL injection whether it is prevented by any of current available mechanism.

In our architecture we describe the all sessions.

Client: The client of a web browser is effectively making client requests for pages from servers all over the web. In this

article client login to system normally, client sends request to server and gets response. This happens only in normal scenario.

Attacker: Attacker is an unauthorized user. Typically this kind of attacker would be a proficient programmer or engineer with sufficient technical knowledge to understand the weak points in a security system. In this article Attacker attacks the website through SQL injection and XSS. Uses of SQL injection and XSS by the attacker is mentioned below.

a. SQL Injection Attack:

The common use of SQL injection attack is to abuse web pages that allows users to input data into form fields for database queries. Injection is an unintended command sent to an interpreter. Attackers can enter the modified SQL query for user information. The queries directly communicate with database for operations on data like data delete, create and change. The queries create link of the static part and value intended for attack. For example: suppose there are two form fields, one for entering the username and one for password, the authentication is done as follows:

1. String strr = "select count(*)"Form(user) Where uname=" ? ? "& Password=" ? " ";
2. SELECT * FROM users WHERE email = 'xx?@x?x.x?x' AND password = md5("??") OR 1 = -1]');

If user has to be allowed to enter apostrophe('), use replace functions of String class:
"string strrr = userInput.Replace(" ? ", " ? ");"

b. Url Injection Attack

Query url is also a way of attack which is a well crafted attack url. If we have a web page with the url. For example: if you get an URL like
`http://xyz.&.in/word /abc/abc.html`

Then it means, we do not have any vulnerable points in the page. But if the URL is like
`http://xyz.s&.com/pro.php?u_id='xxx'`

then 'u_id=xxx' is a string type query for the url for can be altered by an attacker. The attacker can then enter his query in url which can give him access to the database, causing an attack.
`SELECT * FROM obj WHERE u_id='xxx'`.

'u_id' is parameter of this query and xxx is its value. It is fixed type of parameter but attacker can modify its value, which makes it vulnerable. For Example:-
`http://localhost1/?E_Id='xxx';`
String E_id= "DROP TABLE EmpTable"

Another type of attack is when the attacker uses a UNION query and merges the special crafted query with the original query used by the user.
`http://localhost/?EmpId=' UNION .`

This url will change the following SQL statement `SELECT e_info FROM E_Table WHERE E_ID = " UNION .`

c. Cross Site Scripting Attack(XSS)

Cross site scripting(XSS) is also serious problem of web application that can be used by an attacker. The attacker can insert the malicious script in web application through any external resource.

The web browser executes the malicious code as a legitimate code. For example:

The hacker can modify the URL and execute the malicious code in URL box.

`http://xyz1.com/index23.asp?search=`

The attacker can add modify statement to the URL and hijack the client to his domain.

1. {get Element sByTagName("formpage"[02].act io =)}
2. ">><script>document.location='http://www.xyz.com/bin1/cookies.cgi_?' +document.cookies</script>"
3. varmsg = '<p style="color: red </p>';
msg.addInfoMessage(msg);

The attacker uses this type of script code for cookie theft with the stolen cookie and it helps in accessing the users account.

Server: A server is a program that uses HTTP to serve the files that form web pages to users, in response to their requests, which are forwarded by their system HTTP client. In this article client sends request to server and gets a response. Attacker tracks the session id of user by sending http request to server (eg: GET/user/profie_session_id="xyz") using several malicious code. After this request, server will respond to the user's session_id (eg: _session_id="xyz"). Finally, attacker will attaches malicious script into a database(commands) and gets response for the query accordingly.

IV. PROTECTION AGAINST SQL INJECTION ATTACKS

Malicious attacks make web applications less secure because the intruder can harm the integrity of the database by applying malicious queries.

```
PreparedStatementpobj;  
Pobj = con.prepareStatement("select * from std where userid = ?");  
pobj.setString(01, "?.");
```

The above mentioned query is simple way to curb application attacks. This can be formed by applying simple changes into the server site code. Binding variables is one more way for control SQL injection attacks and through binding variables

- [16] S. W. Boyd, G. S. Kc, M. E. Locasto, A. D. Keromytis, and V. Prevelakis, "On the general applicability of instruction-set randomization," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 3, pp. 255–270, 2010.
- [17] K. Elshazly, Y. Fouad, M. Saleh, and A. Sewisy, "A survey of SQL injection attack detection and prevention," *Journal of Computer and Communications*, vol. 2, no. 8, pp. 1–9, 2014.
- [18] A. Azfar, K.-K. R. Choo, and L. Liu, "A study of ten popular Android mobile VoIP applications: are the communications encrypted?" in *Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS '14)*, pp. 4858–4867, IEEE, Waikoloa, Hawaii, USA, January 2014.
- [19] A. Azfar, K. K. R. Choo, and L. Liu, "Forensic taxonomy of popular Android mHealth apps," in *Proceedings of the 21st Americas Conference on Information Systems (AMCIS '15)*, San Juan, Puerto Rico, August 2015.
- [20] A. Azfar, K. K. R. Choo, and L. Liu, "An android communication app forensic taxonomy," *Journal of Forensic Sciences*, vol. 61, no. 5, pp. 1337–1350, 2016.
- [21] Yousra Faisal Gad MahgoupElhakeem ,Bazara I. A. Barry," Developing a Security Model to Protect Websites from Cross-site Scripting Attacks Using Zend Framework Application", *International Conference on Computing, Electrical and Electronics Engineering (ICCEEE)*, August 2013, PP. 624-629
- [22] Atul S. Choudhary and M.L Dhore, "CIDT: Detection Of Malicious Code Injection Attacks On Web Application", *International Journal Of Computing Applications Volume-52-N0.2*, August 2012, PP. 19- 25.
- [23] Avancini, A. and Ceccato, M. "Towards Security Testing with Taint Analysis and Genetic Algorithms," *ICSE Workshop on Software Engineering for Secure Systems*, vol. 5, pp. 65–71, 2010.