Web Applications Vulnerabilities- A Survey

Baldev Singh
Lyallpur Khalsa College, Jalandhar (Punjab) India

Abstract- Web applications are prone to attack. With the development of new tools and technologies, attackers gets success to attack web applications although enormous security controls are deployed. Attackers use the web application vulnerabilities to attack the web applications. The reason of attack may be any either to defame the web application or owner of the web application, steal the credentials, black mailing or financial gain. Although percentage of frequency of attacks are reducing but still attackers are exploiting vulnerabilities to gain access of control of web applications or adversely affecting the web applications and/or their clients. This paper is a survey of various web application vulnerabilities, their sources of attacks, most prevailing types of vulnerabilities and their impact domains. Considering all these aspects, researchers further my work on countermeasures of such vulnerabilities so that impact of web application vulnerabilities can be reduces at the maximum extent.

Keywords: Web applications, Authentication, Authorization, Cross-site scripting, SSL, SQL Injection.

I. INTRODUCTION

Web applications are one of the types of software that stored on a remote server and run on web serverby using Internet as a communication medium. These application are accessed by the clients by using a web browser or by using the interface of that application. Web applications require both server-side scripts and client-side scripts [2] for the proper functionality. Some of the common web applications' functions [1,3]include Online auctions, Online shopping, online banking, web logs, wikis, email, Content management systems, documents, shopping carts, online forms, instant messaging services, interactive information like by using Wikipedia etc. Best use of web applications is by the way of interaction between the user and the owner of the web applications andthat require some type of authentication, authorization and security.

Securing the Web Applications is always a great challenge as they are at high risk. Although various enterprises and organizations trying their best to secure the web applications and are spending lot on the security and trustworthiness of web applications by using additional hardware and software as firewall but actually the lack of security of the web applications lies in the applications themselves. With the exploration of web applications, numerous attacks [6]are concieved that leads to varierty of secuirty issues in the use of web applications. Proliferation of the use of different new tools and technologies, various new range of vulnerabilties got inception in addition to the earlier various Internet secuirty issues, threats, risks and vulnerabilities. Vulnerabilties [5,16] in the web application system leads to unauthotized acces and breach of data and credentials.

II. VULNERABILITIES IN STUDY

It is widespread writing that "*This site is absolutely secure as it is secured with Secure Socket Layer (SSL) technology. It is generally termed as the use of 128-bit Secure Socket Layer (SSL) technology helps to prevent unauthorized users from viewing any of secure information. That is why, any one may use this (SSL) site with peace of mind because the data is safe with SSL technology".* [19] Inspite of the SSL, there are numerous web applications that are not free from different categories of vulnerabilities and attacks [18]. Here some kinds of vulerabilities are briefly explained.

Broken Authentication: Authentication can be breached by the attackers if there exist defects or flaws in authentication mechanism and that can be compromised like weak password, password bypassing and bruit-force attack.

Broken Access Control: If there is lack of enforcement of proper restrictions [14,15] relevant to authentication, it causes vulnerabilities in authentication process what further leads to unauthorized access to other users' accounts, alteration of access rights and privileges, view and/or modify other sensitive and confidential data.

SQL Injection:SQL Injection [3,4] is another important vulnerability that enables the attackers to input untrusted data by way of a command or query which is injected in SQL query statement [8,9] and able to intervene with the logic of the web applications, fetch arbitrary data from the web application. These queries or commands are executed over the database server.

Information leakage: Information leakage in web applications can be due to vulnerabilities [13] like poor and defective error handling and/or inappropriate behaviour adopted. Lacking in Logging & Monitoring

activities causes key web application security issues. Lacking in Logging & Monitoring activities cause various breaches in security, granting access to attackers in the system, tampering and/or destroy of data.

Cross-site scripting: Cross-site scripting [1, 2, 12] is one of the main vulnerabilities due to which attacks occurs on web applications. This type of occurrence takes place when web applications accepts malicious embedded inputs known as injection attack which triggers as client-side code. When this input which is not properly validated, put to client as a reflected script code causes phishing attack. The malicious script is also injected in the database [15]for malicious purpose which is a widespread security issue. The malicious code injected in the legitimate web page(s) of a web application affects the client/user of the web application [17]. That particular website is used as vehicle to deliver the illegitimate code to the client. Generally discussion forums, web pages that accept comments and or message boards are used as vehicles of malicious code if these vehicles are vulnerable.

Buffer Overflows Attacks: Buffer overflows vulnerabilities cause the code injection or denial of service attacks. Denial of service attack further leads to the process crash. Code injection modifies the actual content by injecting malicious content in it through which attacks [12] are generated by the attackers.

Canonicalization: Canonicalization is way of attack over web applications. For instance, one can input name "Narain" as "Narayan", "Nerain", NaRain". Although it looks like same to pronounce but their ASCII value is different and having different representation. This type of evade in input validation can be helpful to fulfil the objective of hackers although it looks least ambiguous. If hackers succeed to use Canonicalization in an offensive way then this type of bug refers to vulnerability that further be the cause of attacks on web applications [10].

Command Injection Attack Vulnerability: Lack of sufficient validations of the commands (operating system/Execution of system) leads to source code vulnerabilities through which an attacker can pass malicious commands through which the attacker gains the access of the system control. Such command injection [8, 9] vulnerability helps attackers to assault the control of the web application.

Directory Traversal Attack (DTA) Vulnerability: DTA vulnerability is the source of password hacking. DTA vulnerability [11] can be used by attackers to identify the weak password even compute the password by using program of iteration process.

III. WEB APPLICATION VULNERABILITY THREAT AREAS AND IMPACTS

Various web tools and technologies like ActiveX, JavaScript, VBScripts, Flash, and CSS can be the source of unsanitized user inputs. Attacker get the access of objects and cookies (including session cookies) of the users and also approached to the confidential and sensitive data of users. Frequency of various web application vulnerabilities differs.

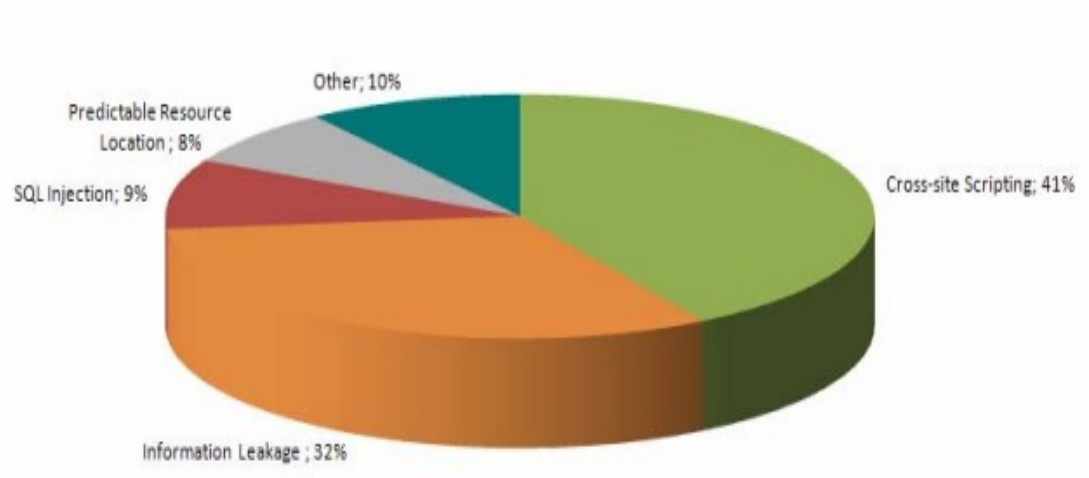

Figure 1: Types based Web Application Vulnerability Frequency [7]

The higher side web application vulnerabilities are Cross-side Scripting and Information Leakage [1,7]. Figure-1 shows vulnerability frequency on the basis of some of the different web application vulnerabilities [7].

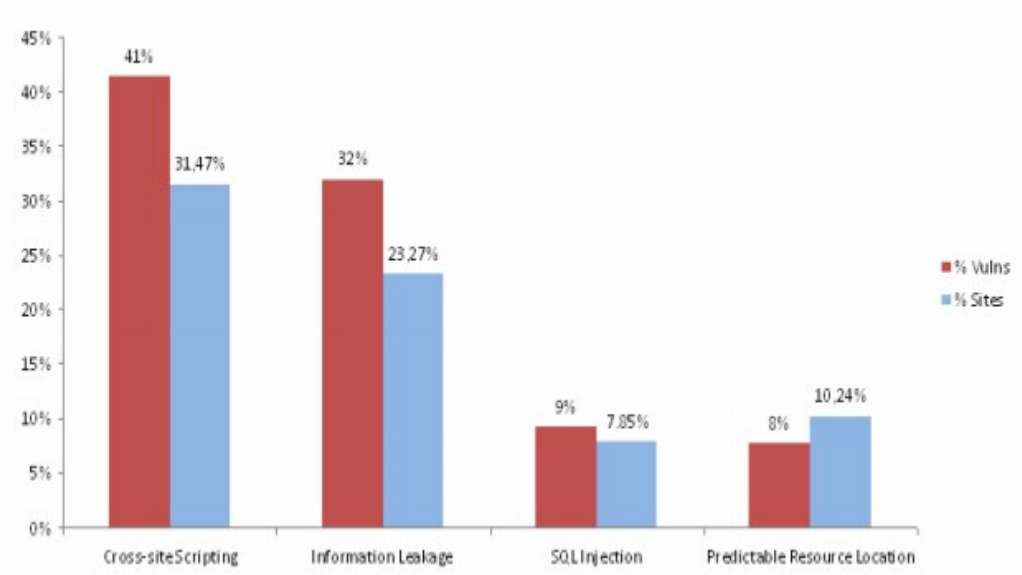

Figure 2: Most prevalent Web Application Vulnerabilities [7]

Web application vulnerabilities not only differ in frequency in context of prone to attacks but also number of web sites being adversely affected or having impact of these vulnerabilities. Figure-2 shows the percent of sites on which most prevalent vulnerabilities have impact along with percentage of vulnerabilities of web applications [7].

Table-1: Web Application Vulnerability Prone Areas and their Threats

Sr.No.	**Vulnerability Prone Area**	**Vulnerability Threat**
1	Script-code	Cross-Site Scripting
2	Script-code	SQL Injection
3	Script-code	Session Fixation
4	Script-code	XML Injection
5	Script-code	Improper Parsing
6	Script-code	Lack in Authentication
7	Script-code	Lack in Authorization
8	Script-code	Insufficient Session Expiration
9	Script-code	Parsing not proper and inline
10	Script-code	XQuery Injection
11	Script-code	URL Redirectors
12	Script-code	Integer Overflow
13	Script-code	Buffer Overflow
14	Script-code	Remote File Inclusion
15	Script-code	OS Commanding

16	Lack in Administration	Denial of Service (DOS)
17	Lack in Administration	HTTP Request/ Response Smuggling
18	Lack in Administration	Misconfiguration of Application
19	Lack in Administration	Misconfiguration of server
20	Lack in Administration	Lack of Transport Layer/ Data Protection

Above Table-1 shows various web application vulnerability prone areas and their threats. The attackers mainly exploit the code by way of altering the script code or injecting malicious code in the actual code. Attackers also gain access by finding and exploiting administrative controls. If there exists insufficient administrative controls that may cause to malicious activities by the hackers and attackers.

Table-2: Web Application Vulnerability Threats and their Impact

Sr. No.	**Vulnerability Threat**	**Impact Side (Client/Server)**
1	Cross-Site Scripting	Client
2	SQL Injection	Server
3	Session Fixation	Server
4	XML Injection	Server
5	Improper Parsing	Server
6	Lack in Authentication	Server
7	Lack in Authorization	Server
8	Insufficient Session Expiration	Server
9	Parsing not proper and inline	Server
10	XQuery Injection	Server
11	URL Redirectors	Client
12	Integer Overflow	Server
13	Buffer Overflow	Server
14	Remote File Inclusion	Server
15	OS Commanding	Server
16	Denial of Service (DOS)	Server
17	HTTP Request/ Response Smuggling	Client
18	Misconfiguration of Application	Server
19	Misconfiguration of server	Server
20	Lack of Transport Layer Protection	Client

Table-2 above shows some of the web application vulnerability threats and their impact. The impact of vulnerability is either client side or server side. Attackers exploit vulnerabilities and attacks either client or server side. Ultimately loss of attacks will be reflected on the web application owners/users. By identifying these vulnerabilities well in time the adverse impact of attacks can be minimized. Important is to identify and remove these vulnerabilities mainly undiscovered vulnerabilities.

IV. CONCLUSION

Web Applications proliferation is taken place as their use is witnessed in almost everysphere of life of the human life. With the propagation of web applications there are flooding of large number of attacks on web applications. Web applications vulnerabilities like injection vulnerabilities such as SQL Injection, Cross site Scripting, Cross site Request Forgery (CSRF) are very common and exploit personal and financial credentials of the concerns. This paper specifically focuses on various kinds ofvulnerabilities. Proper defence mechanisms are need of the day to handle such vulnerabilities so that unauthorized access to the legitimate data, malicious input by way of injections and/or malicious code, improper handling of data and administration can be avoided. This study can be used further for research on various countermeasures to handle web application vulnerabilities and attacks.

REFERENCES

[1] Shalini, S & Usha, S. (2011). Prevention of Cross-Site Scripting Attacks (XSS) On Web Applications in The Client Side. Int. J. Comput. Sci. Issues. 8.

[2] Weinberger, Joel & Saxena, Prateek & Akhawe, Devdatta & Finifter, Matthew & Shin, Eui & Song, Dawn. (2011). A Systematic Analysis of XSS Sanitization in Web Application Frameworks. 150-171. 10.1007/978-3-642-23822-2_9.

[3] Halfond, William & Orso, Alessandro. (2005). AMNESIA: analysis and monitoring for NEutralizing SQL-injection attacks.. 20th IEEE/ACM International Conference on Automated Software Engineering, ASE 2005. 174-183. 10.1145/1101908.1101935.

[4] Halfond, William & Orso, Alessandro. (2005). Combining static analysis and runtime monitoring to counter SQL-injection attacks. ACM SIGSOFT Software Engineering Notes. 30. 1-7. 10.1145/1082983.1083250.

[5] Common Weakness Enumeration (2012) CWE-89: improper neutralization of special elements used in an SQL command (‘SQL injection’). http://cwe.mitre.org/data/definitions/89.html. Accessed on 05 January, 2012

[6] Common Weakness Enumeration (2012) CWE-89: improper neutralization of special elements used in an SQL command (‘SQL injection’). http://cwe.mitre.org/data/definitions/89.html. Accessed on 06 January, 2012

[7] WEB APPLICATION SECURITY STATISTICS 2008 http://projects.webappsec.org/f/WASS-SS-2008.pdf Accessed on 10 January, 2012.

[8] Junjin, Mei. (2009). An Approach for SQL Injection Vulnerability Detection. 1411 - 1414. 10.1109/ITNG.2009.34.

[9] Kim, Jeom-Goo. (2011). Injection Attack Detection Using the Removal of SQL Query Attribute Values. International journal on information. 14. 10.1109/ICISA.2011.5772411.

[10] Kiezun, Adam &Guo, Philip &Jayaraman, Karthick& Ernst, Michael. (2009). Automatic Creation of SQL Injection and Cross-Site Scripting Attacks. Software Engineering, 2009. ICSE 2009. 10.1109/ICSE.2009.5070521.

[11] Sadalkar, Kunal& Mohandas, Radhesh&Pais, Alwyn. (2011). Model Based Hybrid Approach to Prevent SQL Injection Attacks in PHP. 7011. 3-15. 10.1007/978-3-642-24586-2_3.

[12] Halder, Raju&Cortesi, Agostino. (2010). Obfuscation-based analysis of SQL injection attacks. 931-938. 10.1109/ISCC.2010.5546750.

[13] Putthacharoen, Rattipong&Bunyatnoparat, Pratheep. (2011). Protecting cookies from Cross Site Script attacks using Dynamic Cookies Rewriting technique.

[14] Boyd, Stephen &Keromytis, Angelos. (2004). SQLrand: Preventing SQL injection attacks. Proceedings of the 2nd Applied Cryptography and Network Security (ACNS) Conference. 3089. 10.1007/978-3-540-24852-1_21.

[15] Stephen T, Williams L, Xie T (2009) On automated prepared statement generation to remove SQL injection vulnerabilities, Department of Computer Science, North Carolina State University, Raleigh.

[16] KiranMaraju, CISSP, CEH, ITIL, SCJP, Security Code Review- Identifying Web Vulnerabilities.

[17] Atashzar, H. &Torkaman, Atefeh&Bahrololum, M. &Tadayon, Mohammad Hesam. (2011). A survey on web application vulnerabilities and countermeasures. 647-652.

[18] Anthony Dessiatnikoff, Rim Akrout, Eric Alata, Mohamed Kaâniche, Vincent Nicomette. (2011) A clustering approach for web vulnerabilities detection. 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC 2011), Dec 2011, Pasadena, CA, United States. pp.194-203, 10.1109/PRDC.2011.31

[19] DafyddStuttard Marcus Pinto, (2007). The Web Application Hacker’s Handbook Discovering and Exploiting Security Flaws, Wiley Publishing, Inc.