changing their email address, and so forth. If the victim is an administrative account, CSRF can compromise the entire web application.

Properties: Confidentiality: High, Integrity: High, Availability: Low, Time of Introduction (TOI) in SDLC Phase: Design (D); Likelihood to Exploit (LoE): High

C. SQL Injection (SQL i)

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system [15,20].

Properties: Confidentiality: High, Integrity: High, Availability: Low, Time of Introduction (TOI) in SDLC Phase: Design (D), Implementation (I), Operation (O); Likelihood to Exploit (LoE): Very High

D. Information Leakage

Information Leakage is an application weakness where an application reveals sensitive data, such as technical details of the web application, environment, or user-specific data [15,17]. Sensitive data may be used by an attacker to exploit the target web application, its hosting network, or its users. Therefore, leakage of sensitive data should be limited or prevented whenever possible. Information Leakage, in its most common form, is the result of one or more of the following conditions: A failure to scrub out HTML/Script comments containing sensitive information, improper application or server configurations, or differences in page responses for valid versus invalid data.

Properties: Confidentiality: High, Integrity: Low, Availability: Low, Time of Introduction (TOI) in SDLC Phase: Design (D), Implementation (I); Likelihood to Exploit (LoE): High

E. Session Management

Session management attack includes session fixation, insufficient session expiration and broken session attacks. This type of attack occurs when a web application permits an attacker to reuse old session credentials or session IDs for authorization without first invalidating the existing session [16]. The lack of proper session expiration may increase the likelihood of success of certain attacks such as to steal or reuse users session identifiers that further can be used to view other users account or perform a fraudulent transaction.

Properties: Confidentiality: High, Integrity: High, Availability: Low, Time of Introduction (TOI) in SDLC Phase: Design (D), Implementation (I); Likelihood to Exploit (LoE): High

F. Authorization

Insufficient authorization vulnerability occurs when the web application's users are allowed to perform a function or access data without going through proper authorization checks. Attacker can gain access to protected data by exploiting authorization security policy. Weak cryptographic key's generation, weak algorithm usage and insecure storage are the common practices leading to insufficient authorization [15].

Properties: Confidentiality: High, Integrity: High, Availability: High, Time of Introduction (TOI) in SDLC Phase: Design (D), Implementation (I), Operation (O); Likelihood to Exploit (LoE): High

G. URL Redirection

In URL Redirection attack, an http parameter may contain a URL value and could cause the web application to redirect the request to the specified URL. By modifying the URL value to a malicious site, an attacker may successfully launch a phishing scam and steal user credentials. As the server name in the modified link is identical to the original site, phishing attempts have a more trustworthy appearance [16].

Properties: Confidentiality: High, Integrity: Low, Availability: Low, Time of Introduction (TOI) in SDLC Phase: Design (D); Likelihood to Exploit (LoE): Medium

H. Authentication

Authentication related attacks occur when a web site permits an attacker to access sensitive content or functionality without having proper authentication [17]. Hence, depending upon the web site resources, the web application should not be directly accessible to users without verifying the authentication.

Properties: Confidentiality: High, Integrity: Medium, Availability: Low, Time of Introduction (TOI) in SDLC Phase: Design (D), Implementation (I); Likelihood to Exploit (LoE): High

I. Brute Force

Brute Force attack is a method in which an attacker attempts to determine an unknown secret value to gain access to a protected asset by using an automated process usually trial and-error [18]. Entropy of the value is smaller than perceived is the main fact about this attack. Log-in credentials brute forcing in web application is possible, as users usually select easy to memorize words or phrases as passwords. Such an attack attempting to log-in to a system using a large list of words and phrases as potential passwords is often called a "word list attack" or a "dictionary attack".

Properties: Confidentiality: High, Integrity: High, Availability: High, Time of Introduction (TOI) in SDLC Phase: Design (D), Implementation (I); Likelihood to Exploit (LoE): High

J. Server and Application Vulnerabilities

Server and Application Misconfiguration attacks exploit configuration weaknesses found in server and web applications. Many applications come with unnecessary and unsafe features, such as debug features, enabled by default. These features may provide a means for a hacker to bypass authentication methods and gain access to sensitive information, perhaps with elevated privileges. Likewise, default installations may include well-known usernames and passwords, hard-coded backdoor accounts, special access mechanisms, and incorrect permissions set for files accessible through web servers [17]. All of these misconfigurations may lead to unauthorized access to sensitive information.

Properties: Confidentiality: High, Integrity: High, Availability: High, Time of Introduction (TOI) in SDLC Phase: Design (D); Likelihood to Exploit (LoE): High

5. Vulnerabilities Impact on System

This section calculates the Total Technical Impact of each vulnerability, Average impact and Risk of each vulnerability on the system. For each vulnerability, impact value is assigned from 0 to 9, for its impact on Confidentiality (C), Integrity(I) and Availability (A).Total Technical Impact and average impact is calculated with equation 3(a) and 3(b) respectively. In the similar way, Likelihood of Exploitation (LoE) of each vulnerability is assigned value from range 0-9 with its highest exploitability 9, medium 7 and low 3. Next, system risk is calculated according to equation 2 and the results are shown in Table 2.

Table II: Vulnerabilities Impact and Risk

S. No.	*Vulnerability*	*C*	*I*	*A*	*ToI*	*LoE*	*Total Technical Impact*	*Average Technical Impact*	*System Risk*
1	XSS	7	7	5	D,I	7	19	6.333333333	44.333
2	CSRF	7	7	3	D	7	17	5.666666667	39.666
3	SQLi	7	7	3	D,I,O	9	17	5.666666667	51
4	Information Leakage	7	3	3	D,I	7	13	4.333333333	30.333
5	Session management	7	7	3	D,I	7	17	5.666666667	39.666
6	URL Redirection	7	3	3	D	5	13	4.333333333	21.6666
7	Authorization	7	7	7	D,I,O	7	21	7	49
8	Authentication related	7	5	3	D,I	7	15	5	35
9	Brute Force	7	7	7	D,I	7	21	7	49
10	Server Vulnerabilities and configuration	7	7	7	D	7	21	7	49
11	Content Spoofing	3	7	3	D,I	5	13	4.333333333	21.666
12	Insufficient Transport layer protection	7	7	3	D,O	3	17	5.666666667	17
13	Directory Indexing	7	3	3	D	3	13	4.333333333	13
14	path traversal	7	7	7	D,I	7	21	7	49
15	Buffer overflow	7	7	7	D,I,O	7	21	7	49

It shows that SQLi vulnerability has the highest system risk value, and the vulnerabilities authorization, brute force, server mis-configuration, path traversal, and buffer overflow have the next value of risk and highest technical impact. SQLi is on the highest technical risk as it is easy to exploit and popular. While on the other hand, the vulnerabilities Insufficient Transport Layer Protection and Directory indexing are not as easy to exploit, they need sound technical knowledge for exploitation and thus have less risk on the system. Moreover, the system risk is dependent on the technical and business impact and here we are considering technical impact only as business impact varies with organization and their data. Any organization can find out the total impact of vulnerability by summing up the technical and business impact (according to its need). This may help the organization to decide that what they want in their web sites and how they can develop more secure web sites just by keeping in mind the risk that their organization can afford as shown in figure3.

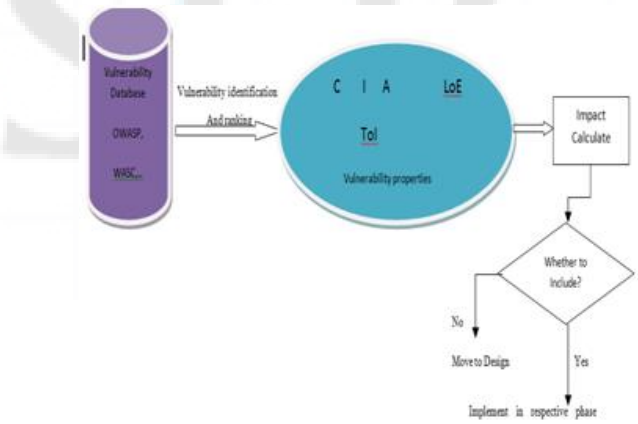

Figure 3: Vulnerability Classification and Decision making

Tolerance level of risk is decided by the organizations which may help them to decide whether to take the preventive actions for the vulnerability or to just ignore it without wasting much time on its preventive implementation.

6. Conclusion and Future Work

In this paper security vulnerabilities related to web applications were ranked according to their frequencies in various vulnerabilities database. This paper also describes top ten vulnerabilities found and calculates the technical risk of each vulnerability, which may help the organizations to decide to afford it or not. The major need is to devise a strategy to develop web applications immune to the vulnerabilities. One technique may be development of security requirements based on the vulnerability analysis and assessment. Requirements are considered as foundation stone on which the entire software is to be built and the requirements phase is the foremost opportunity for the product team to consider, how security will be integrated into a development process. This research work may help to provide effective and efficient ways to incorporate security 'in inception itself' in the development life cycle enabling secured web applications.

References

[1] "Application Vulnerability Trends Report: 2014" Cenzic, Inc. Retrieved from http: //info.cenzic.com/rs/cenzic/images/Cenzic-Application-Vulnerability-Trends-Report-2013.pdf
[2] Mead, R.Nancy and G.McGraw, A Portal for software Security. IEEE Security and Privacy. Published by IEEE Computer society 2005.
[3] "Cisco 2014 Annual Security Report" by Cisco. Retrieved from http: //www.cisco.com/web/offers/lp/2014-annual-security-report
[4] "HP 2012 cyber Security Report" by HP. Retrieved from http: //www.hpenterprisesecurity.com/collateral/whitepaper/HP2012CyberRiskReport_0213.pdf
[5] "Understanding Web Application security challenges", Retrieved from ftp: //ftp.software.ibm.com/software/rational/web/whitepapers/r_wp_webappsecurity.pdf
[6] "The Ten Most Critical Web Application Security Vulnerabilities". Retrieved from http: //umn.dl.sourceforge.net/sorceforge/owasp/OWASPTopTen2010.pdf,2010
[7] Imperva's Web Application Attack Report, Edition 2, January 2012. Retrieved From http: //www.imperva.com/download.asp?=114
[8] Web Application, Retrieved from http: //en.wikipedia.org/wiki/Web_application
[9] R.Crook, D.Ince, L.Lin and B. Nuseibeh, "Security Requirement engineering: When Anti Requirements Hit the Fan", Requirement Engineering, IEEE.pp 203-205 in 2002
[10] C.B.Haley, R.Laney and J.D. Moffett, "Security Requirement engineering: A framework for Representation and analysis", IEEE Transactions on software Engineering, 34(1), pp.133-153 in 2008
[11] C.P. Pfleeger and S.L. Pfleeger, "Security in Computing", 3rd edition. Prentice Hall PTR, 2003.
[12] H.c.Joh and Y.K. Malaiya, "Defining and Assessing Quantitative Security risk Measures Using vulnerability Life Cycle and CVSS Metrics", International Conference Security and Management. SAM'11
[13] "The OWASP Risk Rating Methodology", Retrieved from https: //www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#The_OWASP_Risk_Rating_Methodology
[14] CWE/SANS Top 25 Most Dangerous Programming Errors and Common Weakness Enumeration (CWE). Retrieved from http: //cwe.mitre.org/top25
[15] R.Kumar, "Development of security Requirements for Web Applications", Ph.D Thesis, Jamia Millia Islamia (A Central university), New Delhi, India. September 2011
[16] R.Kumar, "Revisiting Security Vulnerabilities: Web application Perspective", International Journal of Advance Research in Computer science and Software Engineering Vol.3, Issue6, June 2013.
[17] S.B. Chavan and B.B. Meshram, "Classification of Web Application Vulnerabilities", IJESIT Vol2, issue 2, March 2013.
[18] D.Endler," Brute force Exploitation of web application session Ids", Idefense 2001. Retrieved from http: //www.cgisecurity.com/lib/SessionIDs.pdf
[19] R.D.Kombade and B.B. Meshram," CSRF Vulnerabilities and Defensive Techniques" IJCNIS, 2012.
[20] A.TajPour, S. Ibrahim and M.Sharifi, "Web Application Security by SQL Injection Detection Tools", IJCSI Vol.9 Issue2, No. 3, March 2012