A Study on Web Application Security Vulnerabilities

**Abstract**

The world is exceedingly dependent on the Internet, Web applications are one of the most prevalent platforms for information and services Exchange over Internet today. Nowadays, web security is greatest challenge in the corporate world. As almost all organizations has using the web application service to share or store sensitive information.

So Web applications are inclined to security attacks and Number of security vulnerabilities in web application has grown with the tremendous growth of web application in last two decades. so web applications become a well known and important target for security attacks by attackers. So it is very vital to secure a web application from attacks by unauthorized users. A lot of the issues that occur over a web application is basically due to the improper input provided by the client.

This paper reviews the area of web application security Vulnerabilities,

1. **Introduction**

World Wide Web has advanced from a framework that delivers static pages to a stage that supports distributed and dynamic applications, known as web applications and become one of the foremost predominant technologies for delivering information and service over Web.

Web application advancements give a promising system of coordinating numerous useful segments over the web and therefore empower people and associations to cooperate each other utilizing application program interface along enormous topographical separations. Billions of people everywhere throughout the world use web application advancements to exchange data, perform money related exchanges, and have fun and communicate and to socialize themselves [3, 5, 6].

Web application grew tremendously in the last few decades and it has brought great benefits to the people, however, these benefits are associated with some challenges and one of the most important challenges is that of security. Security in web application refers to the threat which occurs due to flaws in software design, coding, testing and implementation. Web application services are more prone to cyber attacks due to their public access. And web applications are increasingly used to deliver security critical services so they become a valuable target for security attacks. Many web applications interact with back-end database systems, which may store sensitive information (e.g., financial, health), the compromise of web applications would result in breaching an enormous amount of information, leading to severe economical losses, ethical and legal consequences [7,8].

The Web platform is a complex ecosystem composed of large number of components and technologies, including HTTP protocol, web server and server-side application development technologies (e.g., CGI, PHP, ASP), web browser and client-side technologies (e.g., JavaScript, Flash). Web application built and hosted upon such a complex framework faces characteristic challenges postured by the highlights of those components and innovations and the irregularities among them.

For developers with insufficient security vulnerabilities knowledge or awareness results in a high rate of web applications sent on the Web is uncovered to security vulnerabilities. According to a report by the Internet Application Security Consortium, around 49% of the internet applications being looked into contain vulnerabilities of tall hazard level and more than 13% of the websites can be compromised totally naturally [1]. A later report [2] uncovers that over 80% of the websites on the Web have had at least one serious Vulnerability.

Vulnerability refers to a weakness in system’s security requirement, design, coding or operation that could accidently occur or intentionally violated and result in security failure. In last few years, number of reported web application security vulnerabilities has increased. Some commonly reported web application vulnerabilities include SQL injection, cross site scripting, command line injection, cross site request forgery and malicious file upload and execution [3, 4].

1. **Introduction to web application**

The Web Application Security Consortium (WASC) [11] defines a web application as “a software application, executed by a web server, which responds to dynamic web page requests over HTTP.”

A web application is comprised of a collection of scripts, which reside on a web server and interact with databases or other sources of dynamic content. Using the infrastructure of the Internet, web applications allow service providers and clients to share and manipulate information in a platform-independent manner. For a good introduction to web application from the penetration tester’s perspective, see [12]. The technologies used to build web applications include PHP, Active Server Pages (ASP), Perl, Common Gateway Interface (CGI), Java Server Pages (JSP), JavaScript, VBScript, hyper Text Markup Language(HTML) etc. Some of the broad categories of web application technologies are communication protocols, formats, server-side and client-side scripting languages, browser plug-ins, and web server API.

A web application has a distributed n-tiered architecture. Typically, there is a client (web browser), a web server, an application server (or several application servers), and a backend (database). Figure 1 presents a simplified view of a web application. There may be a firewall between web client and web server.



Figure 1. Web Application Environment

1. **Vulnerabilities in Web application**

Vulnerability is a weakness in application which can be design flaw or implementation bug that allows an attacker to cause harm to stakeholders of an application.

Formally, vulnerability is defined as “The existence of a weakness, design, or implementation error that can lead to an unexpected, undesirable event compromising the security of the computer system, network, application, or protocol involved” ([Enisa, 2014](https://scialert.net/fulltextmobile/?doi=jse.2014.116.126" \l "66656_an)) .

Vulnerabilities are caused because of poor design, configuration mistakes, inappropriate and insecure coding techniques, complexity of software, unchecked user input, weak password management. The impact of vulnerabilities is very harmful, if a hacker obtains the bank account details of an individual, he can misuse this information (like account number, account balance, loan amount, etc.) and can also alter the data to cause harm to the concerned individual. Vulnerability management in web applications is the cyclical practice of identifying, classifying, remediating and mitigating vulnerabilities. Today, thousands of vulnerabilities are present in web applications. Vulnerability classification is done by different organizations like OWASP and Microsoft ([Williams and Wichers, 2013](https://scialert.net/fulltextmobile/?doi=jse.2014.116.126#66590_an); [Microsoft, 2003b](https://scialert.net/fulltextmobile/?doi=jse.2014.116.126#43212_bc)) based on their risk rating according to exploitability, prevalence, detect ability and impact. Classification according to OWASP, top 10 application security risks are given below:

|  |
| --- |
| OWASP TOP 10 |
| A1:2017-injection |
| A2:2017-Broken Authentication |
| A3:2017-Sensitive Data Exposure |
| A4:2017-Xml External Entities(XXE) |
| A5:2017-Broken Access control |
| A6:2017-Security misconfiguration |
| A7:2017-Cross-site Scripting |
| A8:2017-Insecure Deserialization |
| A9:2017-Using components with known Vulnerabilities |
| A10:2017-Insufficient logging and Monitoring |

**Table1.** OWASP (2017) Top Ten application security Risks

In 2018, around 70 types of weaknesses in web applications are found. As always, Cross-Site Scripting (XSS) vulnerabilities are present in many web applications. Four out of five web applications contained configuration errors such as default settings, standard passwords, error reporting, full path disclosure, and other information leaks that might have value for potential intruders [9].More applications are vulnerable to information exposure. Access to configuration and debug information, source code, session identifiers, and other sensitive information is possible in 79 percent of web applications. This is concerning when compared to past years such as 2016 (60%) and 2017 (70%).



The types of application vulnerabilities based on severity are high, medium, low.fig 2 shows the percentage of web application affected based on severity levels of vulnerability

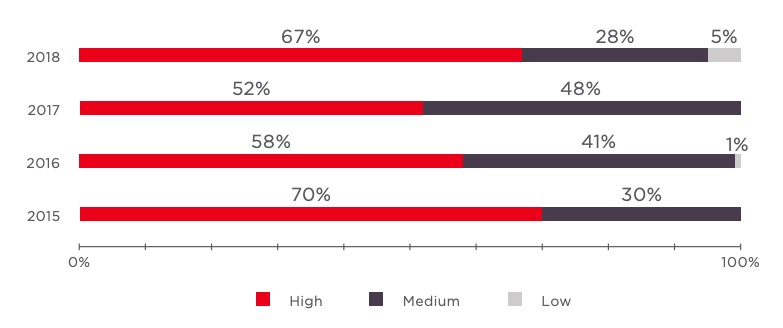


Figure 2. Percentage of Web Application Affected yearly

1. **Sql injection and its type**

**jk**

1. **Xss and its types**

**uuu**

1. **Broken Authentication and Improper Session Management**

**Hfg djk**

1. **Conclusion**

**Sk fjgvd,jmf**

1. **References**

[1]WebApplicationSecurityStatistics,“http://projects.webappsec.org/w/page/13246989/WebApplicationSecurityStatistics.”

[2]White Hat Security, “White Hat website security statistic report 2010.”

[3] Zhang, D., Liu, D., Csallner, C., Kung, D., & Lei, Y. (2014). A distributed framework for demand-driven software vulnerability detection. Journal of Systems and Software, 87, 60-73.

[4] Huang, C. C., Lin, F. Y., Lin, F. Y. S., & Sun, Y. S. (2013). A novel approach to evaluate software vulnerability prioritization. Journal of Systems and Software,86(11), 2822-2840.

[5] Kaur, N., & Kaur, P. (2014). Input Validation Vulnerabilities in Web Applications. Journal of Software Engineering, 8(3), 116-126.

[6] Austin, A., Holmgreen, C., & Williams, L. (2013). A comparison of the efficiency and effectiveness of vulnerability discovery techniques. Information and Software Technology, 55(7), 1279- 1288.

[7] Shar, L. K., & Tan, H. B. K. (2012). Automated removal of cross site scripting vulnerabilities in web applications. Information and Software Technology, 54(5), 467-478.

[8] Avancini, A., & Ceccato, M. (2013). Comparison and integration of genetic algorithms and dynamic symbolic execution for security testing of cross-site scripting vulnerabilities. Information and Software Technology, 55(12), 2209-2222.

[9] Web application Vulnerability Statistics(2018).” https://www.ptsecurity.com/ww-en/analytics/web-application-vulnerabilities-statistics-2019/”.

[11] Web Application Security Consortium Glossary, http://www.webappsec.org/projects/glossary/

[12] Jody Melbourne and David Jorm, Penetration Testing for Web Applications, in SecurityFocus, 2003.