Web Security

1. [What’s OWASP?](#_OWASP)
2. [What is “Vulnerability”?](#_Vulnerability)
3. [What is “SQL Injection”?](#_SQL_Injection)
4. What is cross-site scripting (XSS)?

# OWASP

The **Open Web Application Security Project®** (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open-source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.

## OWASP Top 10 of 2021

* The Injection
* The Broken Authentication
* The Sensitive Data Exposure
* The XML External Entities (XXE)
* The Broken Access Control
* The Security Misconfigurations
* The Cross-Site Scripting (XSS)
* The Insecure Deserialization
* The Using Components with Known Vulnerabilities
* The Insufficient Logging and Monitoring

# Vulnerability

Vulnerability can be defined as the weakness of any system through which intruders or bugs can attack the system.  
If security testing has not been performed rigorously on the system then chances of vulnerabilities get increased. Time to time patches or fixes is required to prevent a system from the vulnerabilities.

# SQL Injection

SQL Injection is one of the common attacking techniques used by hackers to get critical data.

Hackers check for any loophole in the system through which they can pass SQL queries, bypass the security checks, and return back the critical data. This is known as SQL injection. It can allow hackers to steal critical data or even crash a system.

SQL injections are very critical and need to be avoided. Periodic security testing can prevent this kind of attack. SQL database security needs to be defined correctly and input boxes and special characters should be handled properly.

Solutions:

* Use stored procedure (SP)
* Re-validate data in stored procedures.
* Use parameterized query
* Use ORM tools (NHybernet, EF)
* Use regular expression to discard input string

# Cross Site Scripting (XSS)

<https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html>

Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application. It allows an attacker to circumvent the same origin policy, which is designed to segregate different websites from each other. Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data. If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data.

Preventing cross-site scripting is trivial in some cases but can be much harder depending on the complexity of the application and the ways it handles user-controllable data.

In general, effectively preventing XSS vulnerabilities is likely to involve a combination of the following measures:

* Filter input on arrival. At the point where user input is received, filter as strictly as possible based on what is expected or valid input.
* Encode data on output. At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.
* Use appropriate response headers. To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the Content-Type and X-Content-Type-Options headers to ensure that browsers interpret the responses in the way you intend.
* Content Security Policy. As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.