## Cross Site Scripting(XSS)

Cross site scripting is a type of computer security vulnerability found in web application. XSS enables attackers to inject client-side script into web pages viewed by other users. The attacker aims to execute malicious scripts in the web browser of the users by including malicious code in the web page or web application. The actual attack occurs when the user visit the web page that execute the malicious code. We can say that the web page becomes a vehicle that delivers a malicious script to the user’s browser. It mostly uses forums, message boards and web pages that allows comments as a vehicle for cross site scripting.

XSS attacks are possible in VBscript, JavaScript, CSS, Flash, etc. It is most common in JavaScript as JavaScript is fundamental in most of the web application. XSS is one of the most common security vulnerabilities in today's date. XSS vulnerabilities can have consequences such as tampering and sensitive data theft. A successful XSS attack leads to an attacker controlling the victim’s browser or account on the vulnerable web application. Although XSS is enabled by vulnerable pages in a web application, the victims of an XSS attack are the application's users, not the application itself. The potency of an XSS vulnerability lies in the fact that the malicious code executes in the context of the victim's session, allowing the attacker to bypass normal security restrictions.

For example: Most of the web application maintain user sessions in order to identify user across multiple requests. Session cookies are sensitive information which if compromised, allows attacker to impersonate and gain access to their web session. This attack is called session hijacking. It may leads to tampering the victim’s information or commit malicious activities from the victim’s account.

#### Impact of XSS

When attackers succeed in exploiting XSS vulnerabilities, they can gain access to account credentials. By exploiting XSS vulnerabilities, an attacker can perform malicious actions, such as:

* Hijack the account
* Spread web worm
* Access browser history and clipboard the content
* Control browser remotely

## Role based access control

Role based access control is an idea of assigning permissions to the users based on their roles within any organization. It provides a control and offers a simple, manageable approach to access management that is less prone to error than assigning permissions to users individually. When using RBAC, you analyze the system needs of your users and group them into roles based on common responsibilities and needs.

#### Benefits of RBAC

* create systematic, repeatable assignment of permissions
* easily audit user privileges and correct identified issues
* cut down on the potential for error when assigning user permissions
* integrate third-party users by giving them predefined roles