No matter how strictly you have implemented security in your system, you can never be completely sure of its complete protection. Even the most experienced web security experts and CIOs must stay vigilant to guard against the bad guys. One cannot be safe without having the knowledge of what they should look out for. Let’s have a look at five of the most common web security vulnerabilities that you should protect yourself against.

1. SQL Injection

It is a type of vulnerability in the web application security where an attacker makes an attempt to use application code for accessing or corrupting your database content. If the attacker is successful, they can read, create, update, chage or delete stored data in the back-end database. It is one of the most predominant web application security vulnerability.

1. Cross Site Scripting

Cross-Site Scripting or XSS targets the users of an application by injecting code into a web application’s output. This code is often a client-side script, for instance, JavaScript. The concept behind it is to operate a web apps client-side scripts in a desired manner. These scripts executed in a target’s browser can hijack their user-session, redirect them to a malicious website or deface a website.

1. Broken Authentication and Session Management

It encompasses a number of security issues related to maintaining a user’s identity. If a user’s authentication credentials and session identifiers are not safeguarded during the entire session, a cybercriminal can hijack their active session and assume their identity.

1. Insecure Direct Object References

This happens when a web application security vulnerability relates to a web app exposing a reference to any internal implementation object. These objects include database records, files, database keys and directories. When an app exposes a reference to these in a URL, an attacker can manipulate it to get access to a target user’s personal information.

1. Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery attack happens when an hacker tricks a user into performing an action they did not intend to do. A third-party website sends request to a web application that the user has already acquired authentication for e.g. the user’s bank. The hacker then accesses functionality through the victim’s browser which is already authenticated. Target web applications for this type of attack include social media, email clients and online banking accounts.

Want to protect your website from these common web applications security vulnerabilities? Get professional help today from a team of cyber security experts to help you asses your vulnerability to these threats.