

XSS(Cross Site Scripting)

INTRODUCTION

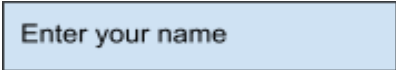
You want to visit a site , you typed in your browser and visited it. But have you ever thought that maybe a person out there in some corner of the world may steal information and all the data about you just after you visited the site? This is when the term Cyber Attacks get Introduced. When an unauthorized third party simply called as cybercriminals attempts to disable, destroy the computer system or steal or gain important data from it is called cyber attacks. There are different types of cyber attacks. XSS is one of the most famous cyber attacks about which we are going to talk in detail.

XSS is the abbreviation used for Cross Site Scripting. XSS is basically injection of malicious code on the client's side via the input section. The code gets executed when the user/victim visits the web page. XSS attacks are used for cookies stealing, modifying content and access over sensitive information.

HOW TO CHECK WHETHER THE WEBSITE IS VULNERABLE TO XSS ATTACK OR NOT

We often come across input sections such as name, email id ,etc, whenever we visit a web page.

Let us say we have a website like this in which you have to enter your name and it will display "Your name is (The name you entered)" :



Let us say we entered Joe so it will display:



But what if we write a tag or a script For example: `<h1>Hello World</h1>`

1.) If the site displays

Your name is `<h1>Hello World </h1>`

Then the site is protected and is not exposed to XSS attacks

2.) If the site displays

Your name is **Hello World**

Then the site is vulnerable to XSS attacks

So, what is actually happening? When you enter the information the website sends that information to the database (web server) and stores/reflects it. If the site decodes the script or tag that means it cross site scripting can be done. For more details we will look into the different types of XSS attacks.

DIFFERENT TYPES OF XSS ATTACKS

- Reflected XSS(Non-Persistent)

In the case of reflected xss, data is not stored in the web server.

Script is stored on the victim's side.

If the site is vulnerable to reflected xss attack, it simply means that if a script is injected in the website it will bounce back or reflect the script after decoding it and the third party can inject any malicious code in order to steal the information. If the website takes any part of the https request from the user it could enable the cyber criminal to inject the script to get the data.

Common way to check the vulnerability is to use a script tag in order to create an alert dialog box. After that you can get an access to session id. Session id is basically an unique string assigned to a user while the session is going on. If someone gets your session id he or she can get your sensitive information even the passwords.

- Stored XSS (Persistent)

Script is stored and executed on the server.

Code is executed every time when any request is made to the malicious site.

For example, Joe uploaded a photograph on facebook. The data that is the image is sent to the database. Jenny, Joe's friend wants to visit Joe's profile to open the image and post a comment. When Jenny tries to open Joe's profile a request is made to the server to show the content of the profile. In this attack malicious code gets executed on the server and whenever someone opens the malicious site or any request is made over the site it gets executed.

- DOM(Document Object Model) XSS

DOM XSS attack is slightly different from the above too. It mainly depends upon the way the site has been constructed.

Client side attack Script is not sent to the server.

Genuine site server script(whose content has to be executed) is executed first and then the malicious code is executed.

Input sections are not there so we have to make changes or inject the code via URL.

The third party manipulates the URL used or the URL being generated.

PREVENTIONS

- For some extent we can control what the input user is giving. For example we can deny the use of special symbols and characters in the input sections.

- General formats for inputs such as email-id, phone numbers can be initialized.
- Input sections must be sanitized.
- Use right response headers. What data can be received, what data can be sent.