

S3- Db02

Sander van Deurzen

**XSS(Cross Side Scripting)**

21-04-2022

**Voorblad**

1. **Wat is Cross Site Scripting?**
2. **Soorten XSS aanvallen**
3. **Wat kan ik tegen een XSS doen?**
4. **Voorbeeld XSS**
5. **Conclusie**

## **Wat is Cross Site Scripting?**

*Cross-site scripting vulnerabilities (XSS henceforth) are a security problem that occurs in web applications. They were discovered in the 1990s in the early days of the World Wide Web . They are among the most common and most serious security problems affecting web applications**. They are a type of injection problems that enable malicious scripts to be injected into trusted web sites. This is a result of a failure to validate input from the web site users. What happens is either the web site fails to neutralize the user input or it does it incorrectly, thus opening an avenue for a host of attacks exploiting for vulnerabilities.* Bron: https://www.sciencedirect.com/

Met een XSS aanval kan je kleine aanvallen doen, zoals de kleur van je website achtergrond aanpassen of een javascript alert creëren, alleen kan je ook met XSS aanvallen iemand zijn database platleggen doormiddel van 1 query in een tekstveld. Dat maakt het super belangrijk om hier beveiligd voor te zijn. En daaruit komt de vraag:

***“Hoe kan ik mijn website beveiligen tegenover XSS aanvallen?”***

## **Soorten XSS aanvallen**

**Als je over XSS spreekt, spreek je niet gelijk over een algemeen onderwerp. We spreken van 3 verschillende soorten XSS aanvallen;**

**-Reflected XSS Attack:**

Stored XSS also known as persistent XSS occurs when user input is stored on the target server such as database/message forum/comment field etc. Then the victim is able to retrieve the stored data from the web application.

**- Stored XSS Attack:**

Stored XSS also known as persistent XSS occurs when user input is stored on the target server such as database/message forum/comment field etc. Then the victim is able to retrieve the stored data from the web application.

**- DOM Based XSS Attack:**

DOM Based XSS is a form of XSS when the source of the data is in the DOM, the sink is also in the DOM, and the data flow never leaves the browser.

## **Wat kan ik tegen een XSS Doen?**

**Gelukkig is Cross side scripting makkelijk te beveiligen. Populaire webframeworks zoals Laravel heeft ingebouwde beveiliging tegenover XSS aanvallen.**

**- Alle niet vertrouwde gegevens op basis van de HTML-Context, attribuut, Javascript, CSS of URL waarin de gegevens zijn geplaatst moeten uitgefilterd worden.**

## **4. Voorbeeld**

Hieronder heb ik een webapplicatie gemaakt die form data stuurt naar een database, vervolgens wordt de data opgeslagen en kan de gebruikers vervolgens die data terug zien op de website

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Zonder enige beveiligingschecks wordt de data gestuurd naar de database. Nu gaan we in een invoerveld een simpel javascript script toevoegen zoals hieronder wordt aangegeven:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**Wanneer ik op verzenden klik wordt de het script uitgevoerd:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

**Nu zie je dat ik elke keer als ik de pagina in kom ik deze melding krijg. Dit voorbeeld is hinderlijk voor je applicatie, alleen als je kijkt wat de verdere mogelijkheden zijn met XSS kan het je applicatie in een script helemaal slopen.**

## **Conclusie**

**XSS blijft nog steeds een groot probleem voor webapplicaties, ondanks het grootste deel van de oplossingen die tot nu toe geleverd zijn. Er is geen enkele oplossing die XSS-aanvallen effectief kan afzwakken. Er is meer onderzoek nodig op het gebied van het verwijderen van kwetsbaarheden uit de broncode van de applicaties voordat ze worden geïmplementeerd.**

**Elke ontwikkelaar moet bewust zijn van alle gevaren die de webapplicatie mee kunnen krijgen en daar is XSS een hele belangrijke van. De ontwikkelaar moet weten hoe hij zich ertegen kan beschermen en welke gevolgen dit kan hebben over de applicatie**