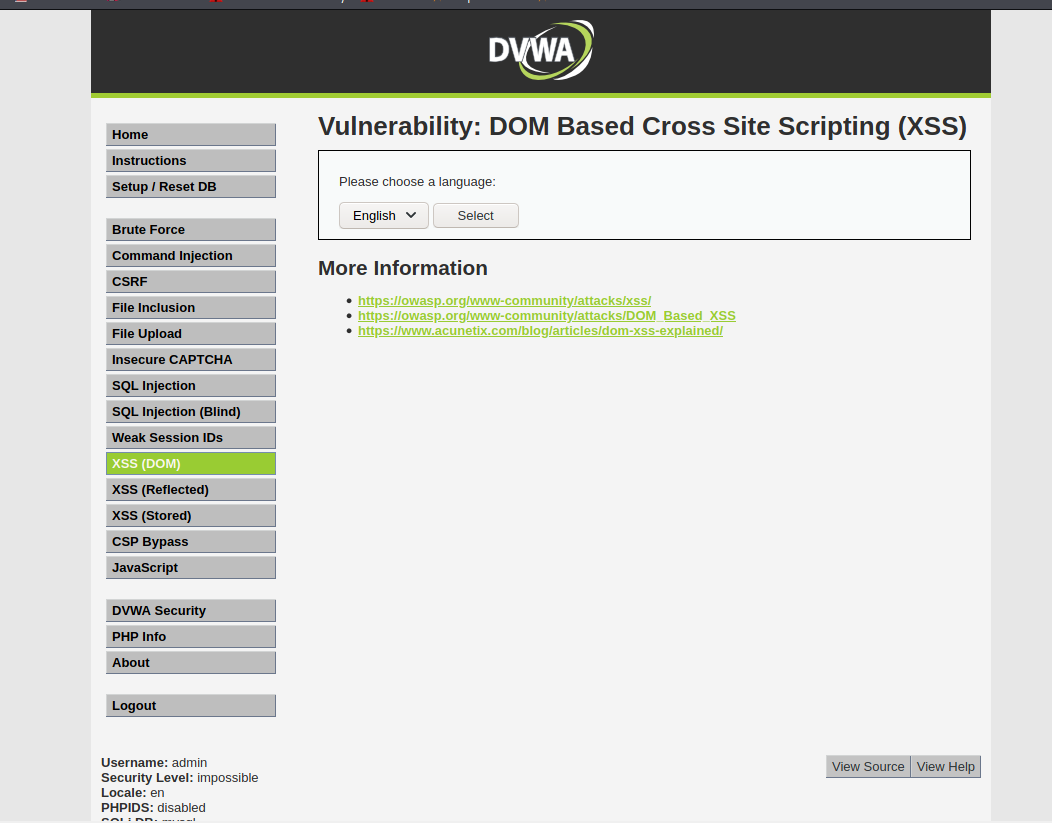
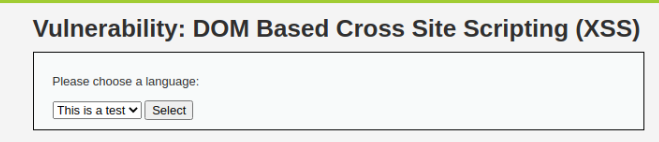
## #Implementing XSS attack on DVWA

**1: XSS DOM**

We first make sure the Damn Vulnerable Web Application security is set to low and navigate to the XSS(DOM) page on DVWA as shown on the image below.

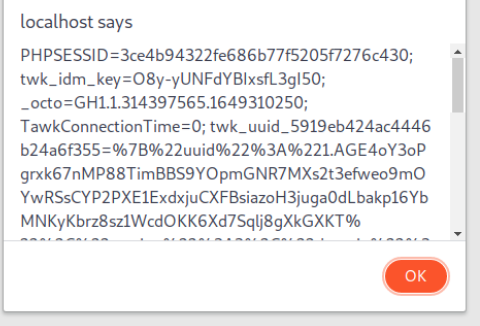


On the image above, we can see on the link that the default has a variable English which is the default language. We will try to enter some random information to see if the website is vulnerable. I tried by replacing English with a string; “This is a test". Once the word is replaced, we can see our new variable as the default variable now as shown on the below image.



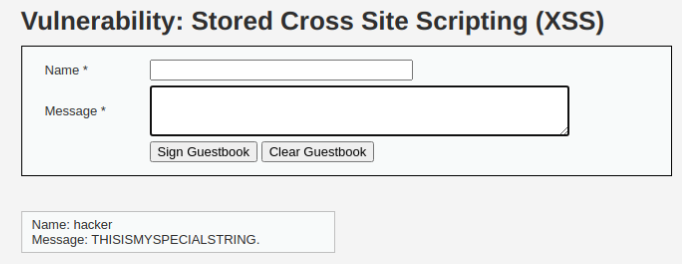
We have confirmed the website is vulnerable, we now move on to adding a script that will run our desired code and since want to get the cookie, we use will use the below script.

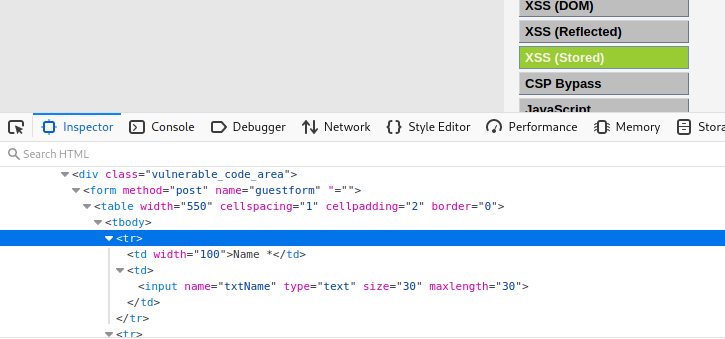
<script>alert(document.cookie)</script>

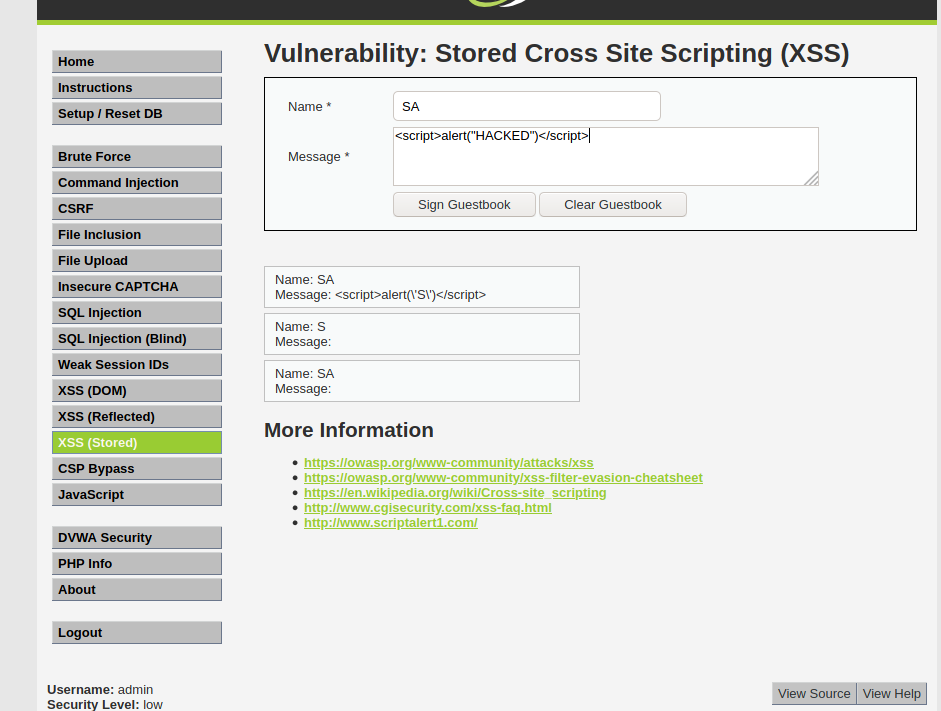


**2. Stored XSS**

In stored XSS, it is more dangerous than others since it is stored in the database. The vulnerability arises when user input data is not filtered before being stored in the database. Attackers injects malicious code to force pop ups to any person who visits the website. These pop ups can be modified to redirect website clients to malicious websites. To exploit this vulnerability on the Damn Vulnerable Web App, we navigate to the XSS(stored) option on the DVWA main page as shown below.







**3. Reflected XSS**

In reflected XSS attacks, the input is executed inside HTML tags unless it is empty hence it does not prevent executing of scripts loaded through the user input. This is one of the easiest but common web application attack. First we will navigate to the XSS reflected category on our damn vulnerable web application from where we are presented with an input field as shown on the image below.

