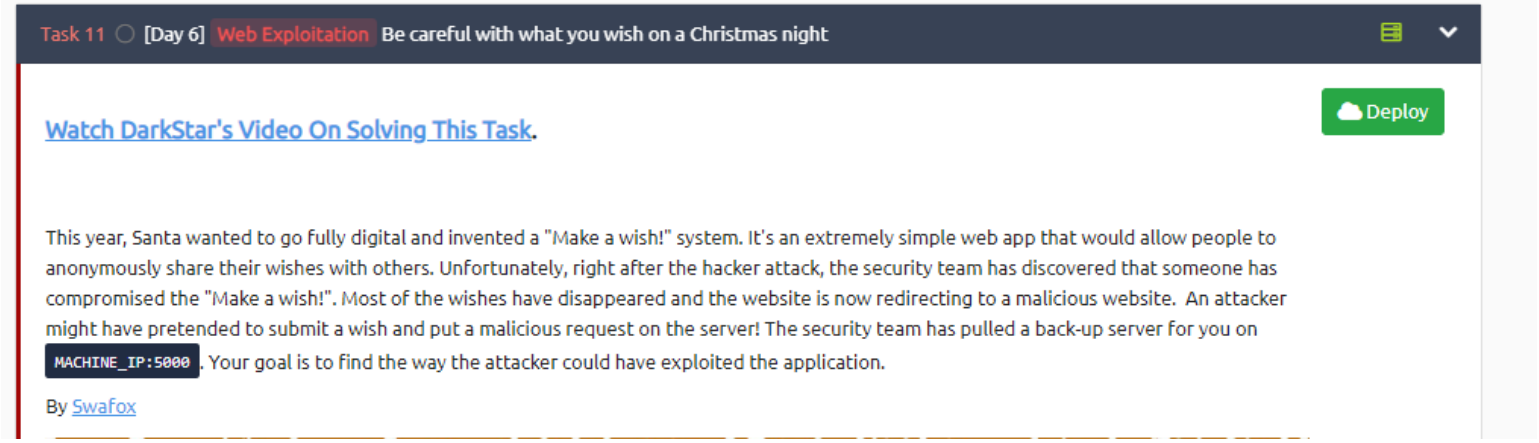
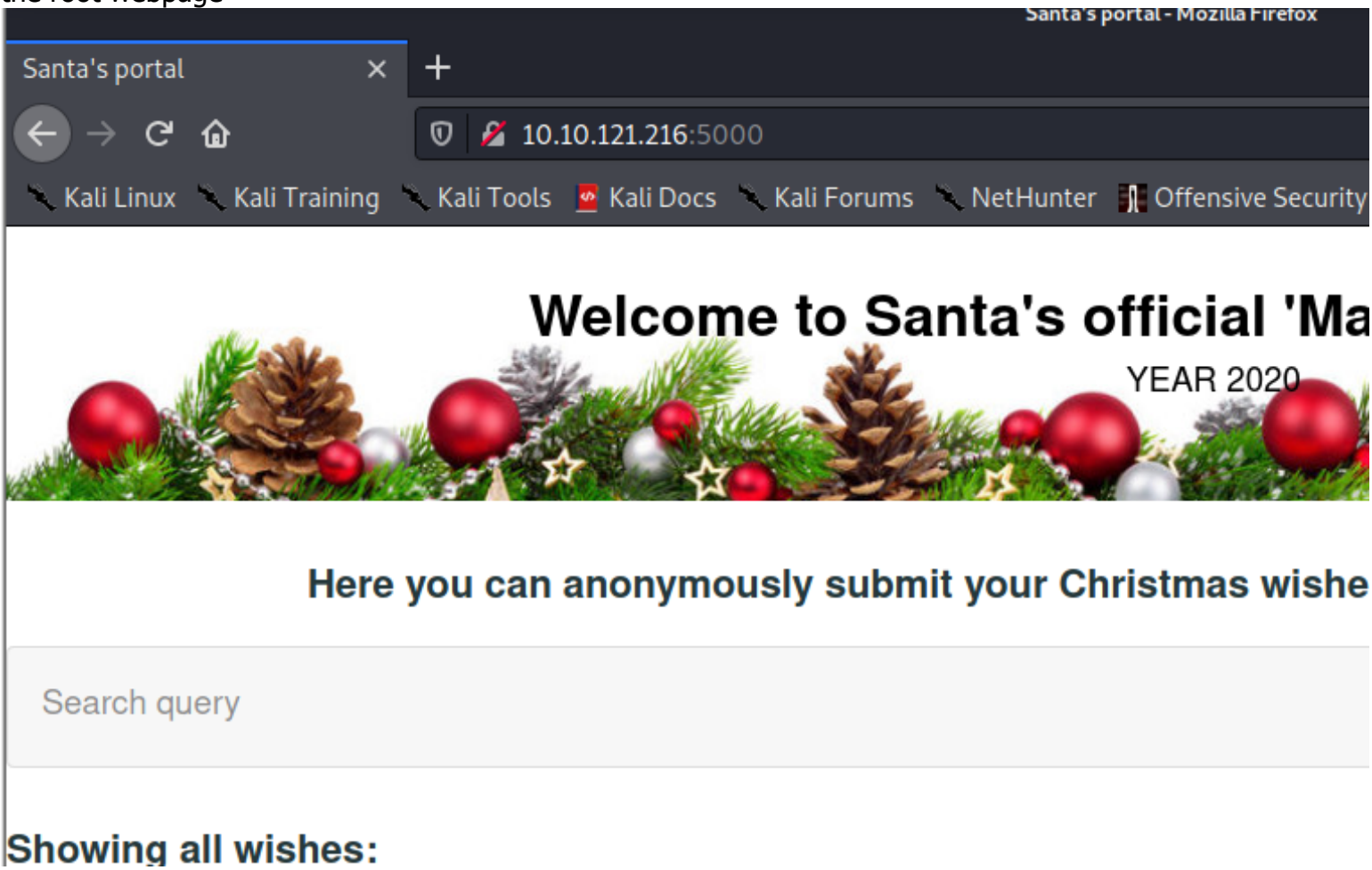
Day 6 - Be careful with what you wish on a Christmas night

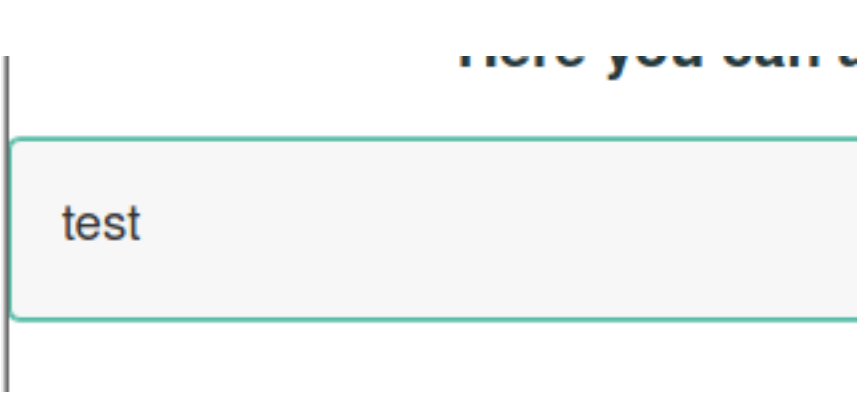
Scenario



the root webpage

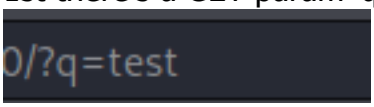


enter test into search query



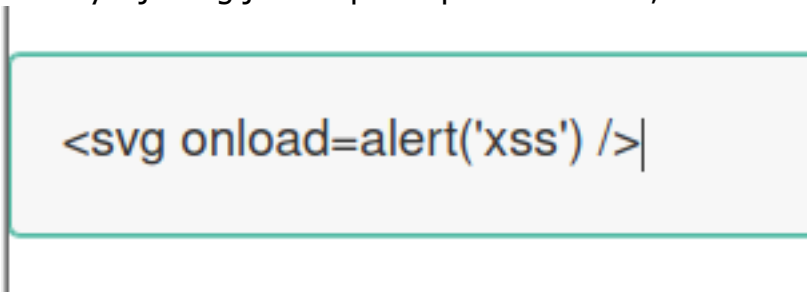
& it shows 2 thing here:

1st there's a GET param ‘q’ with our ‘test’ string as the query

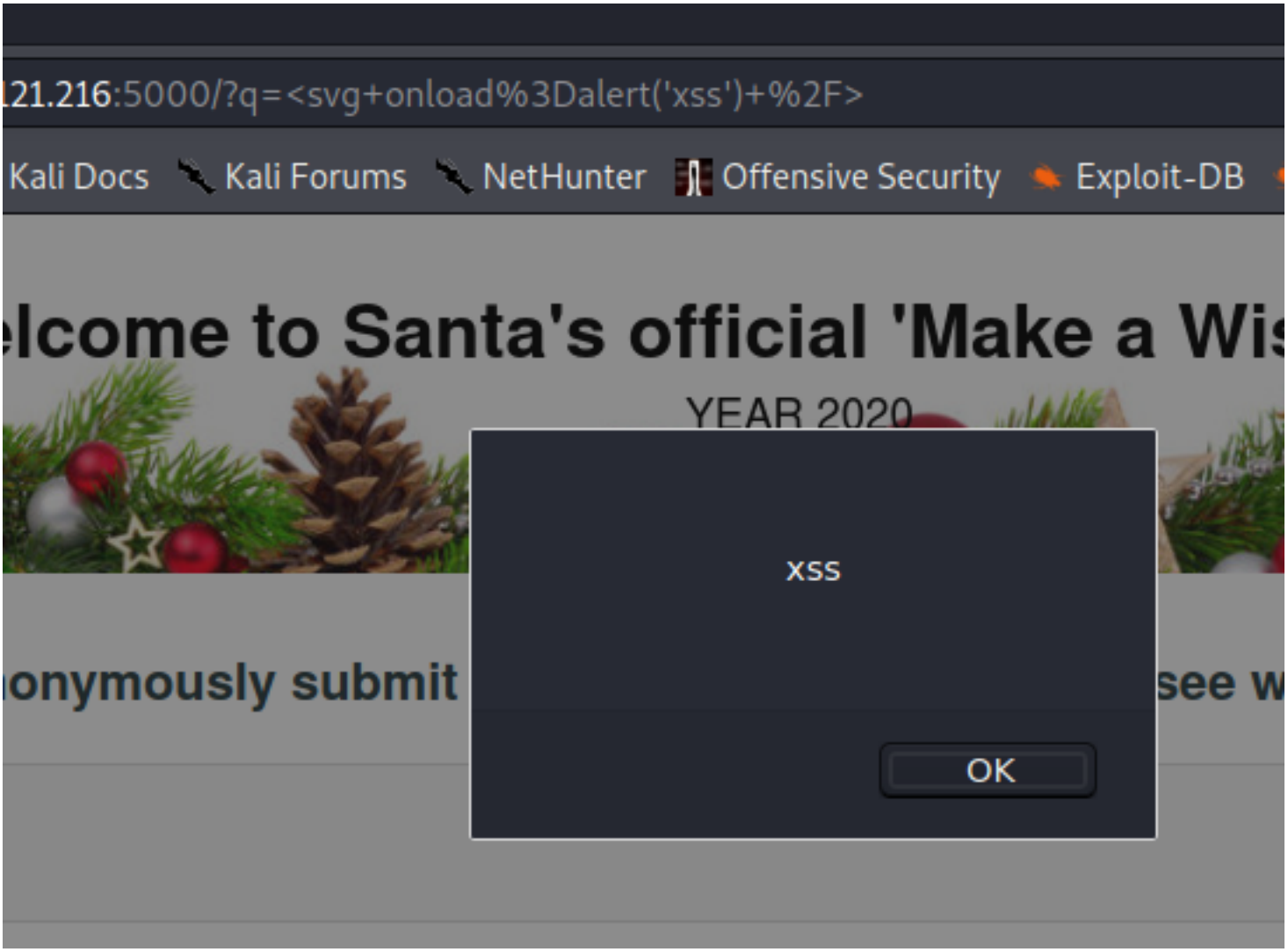


2nd is that the ‘test’ string will be shown up on the webpage

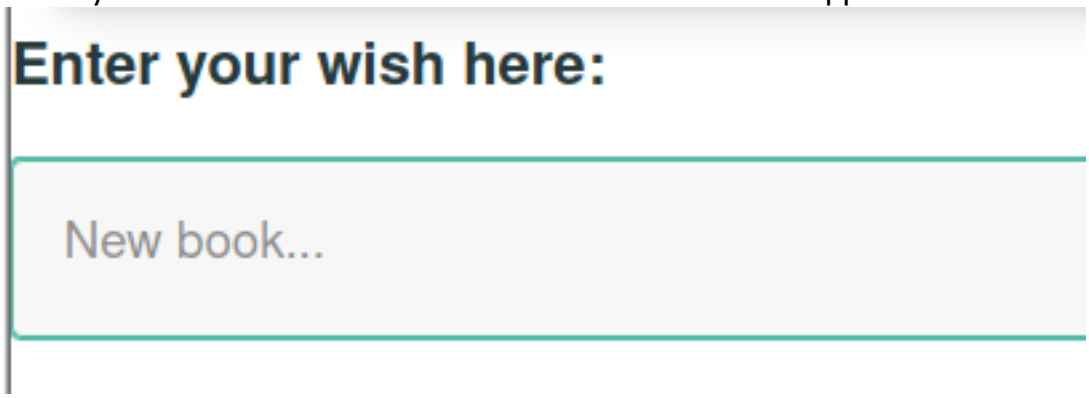
let's try injecting javascript to spawn alert box, see whether it's vulnerable to XSS or not



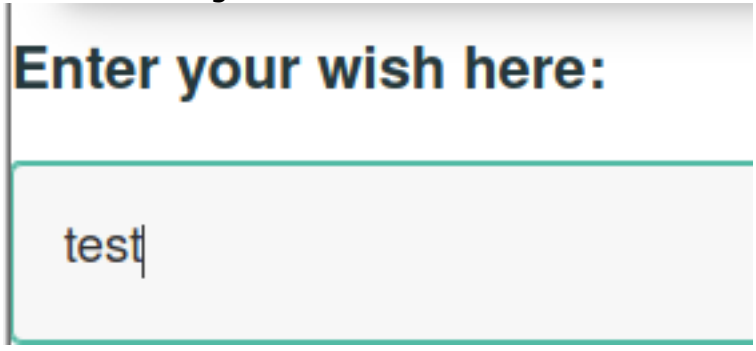
omg the website is vulnerable to reflected XSS vulnerability



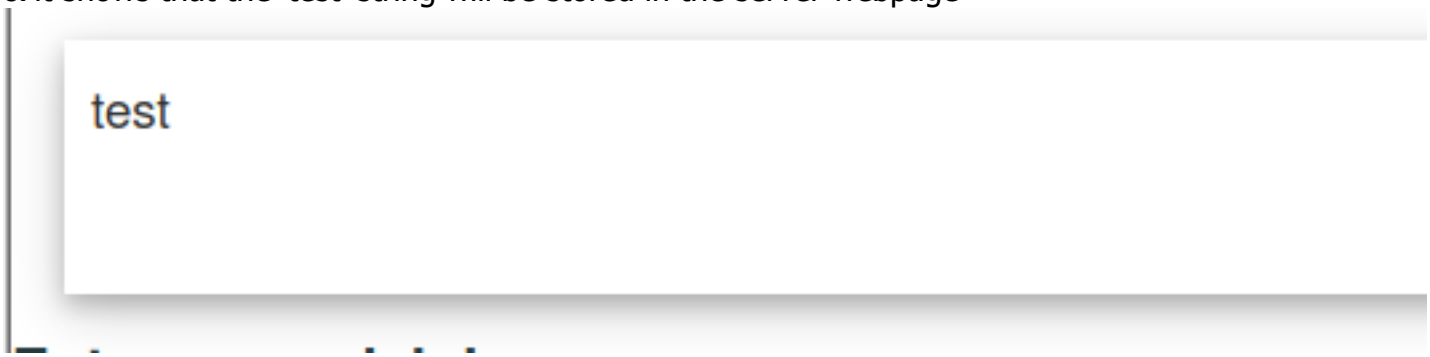
let's try another one here the wish box to see what will happen here



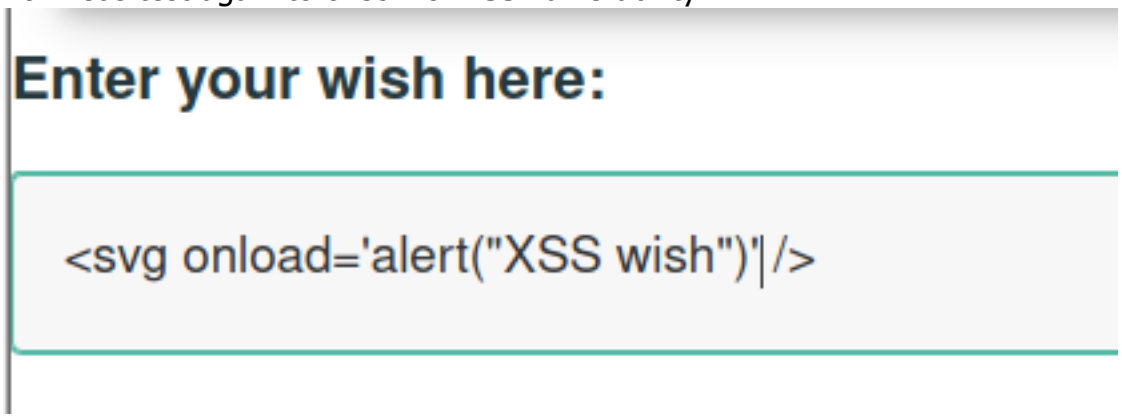
enter test string into it



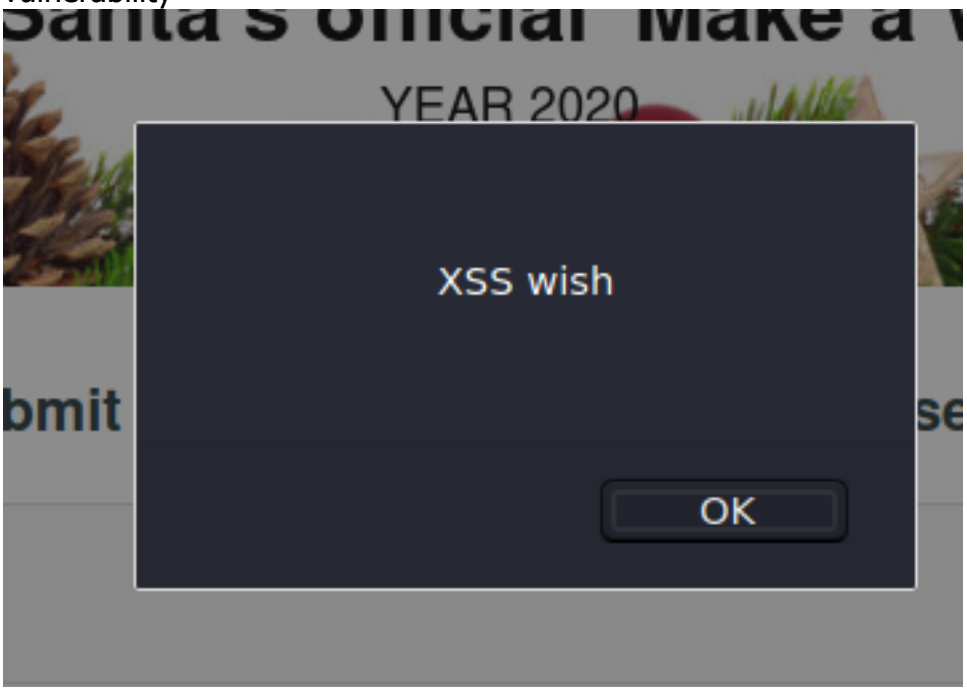
& it shows that the 'test' string will be stored in the server webpage



now let's test again to check for XSS vulnerability



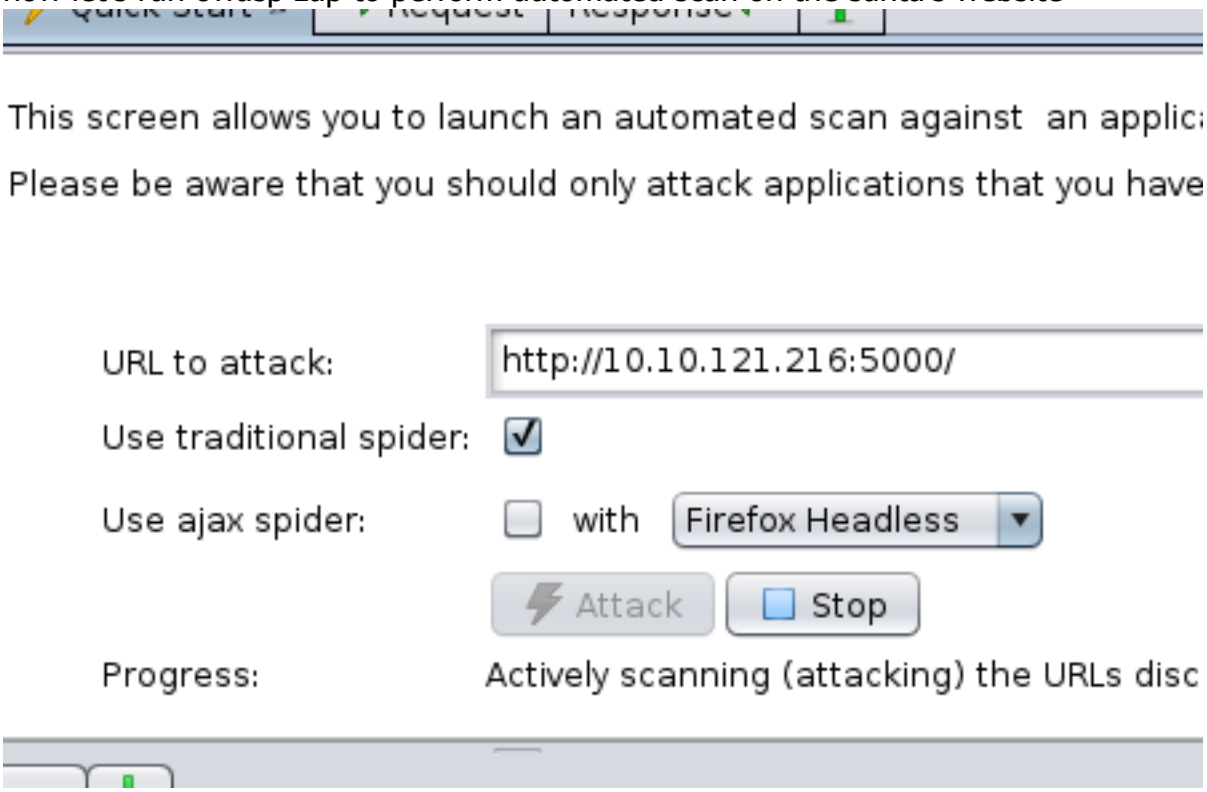
& it works!! everytimes we refresh the webpage it'll popup this vulnerability. This is called stored XSS vulnerability



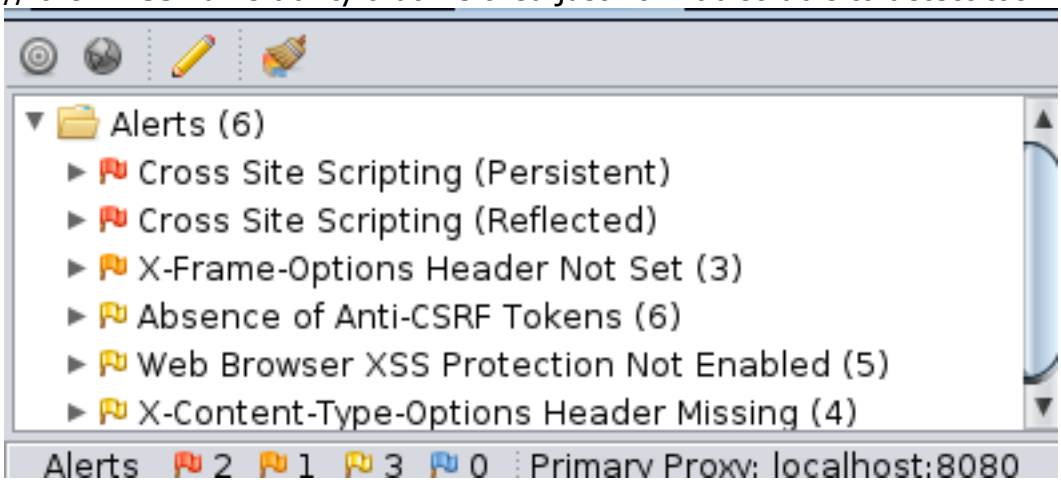
Question: What vulnerability type was used to exploit the application?
-stored cross-site scripting

Question: What query string can be abused to craft a reflected XSS?
-q

now let's run owasp zap to perform automated scan on the santa's website



now here it shows that we've 6 alerts that owasp zap found here
// the 2 XSS vulnerability that we tried just now it also able to detect too



Question: Run a ZAP (zaproxy) automated scan on the target. How many XSS alerts are in the scan?
-2