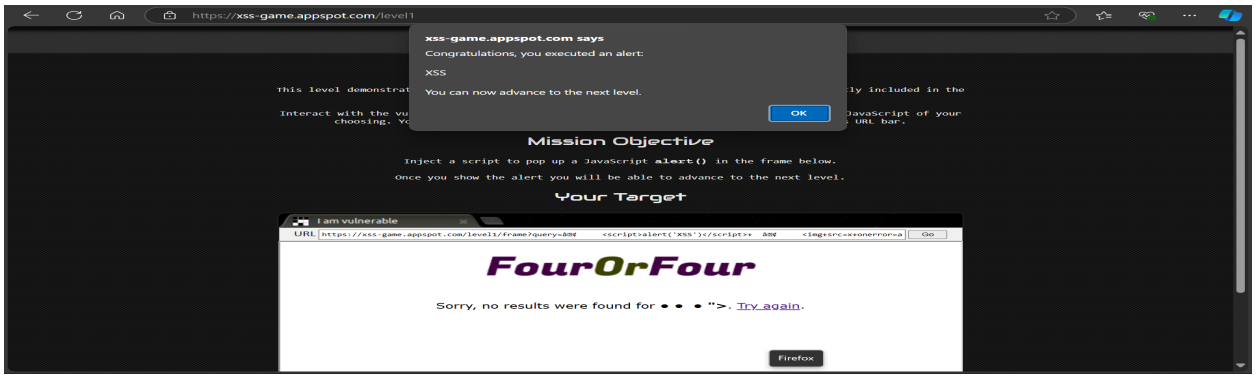
CONDUCTING A CROSS SITE SCRIPTING XXS ATTACK.

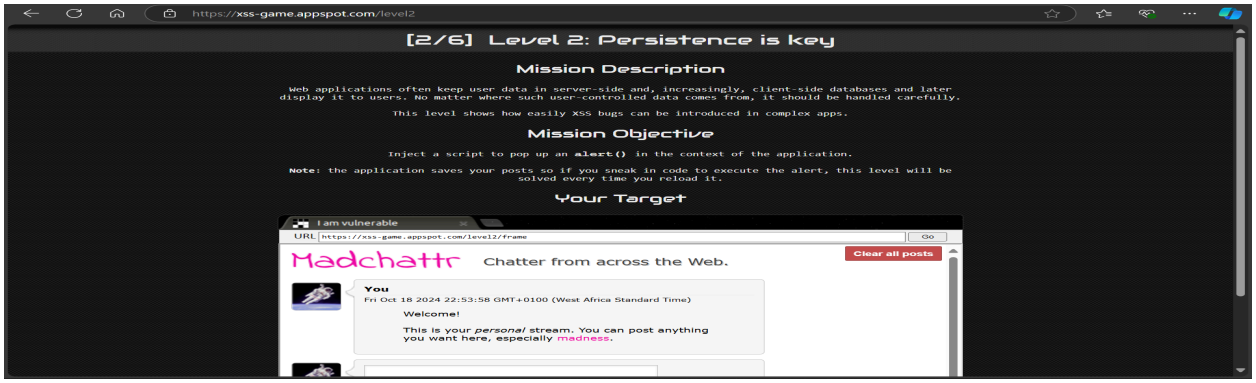
TOOLS : WEB BROWSER
Project-Site : xss-game.appspot.com

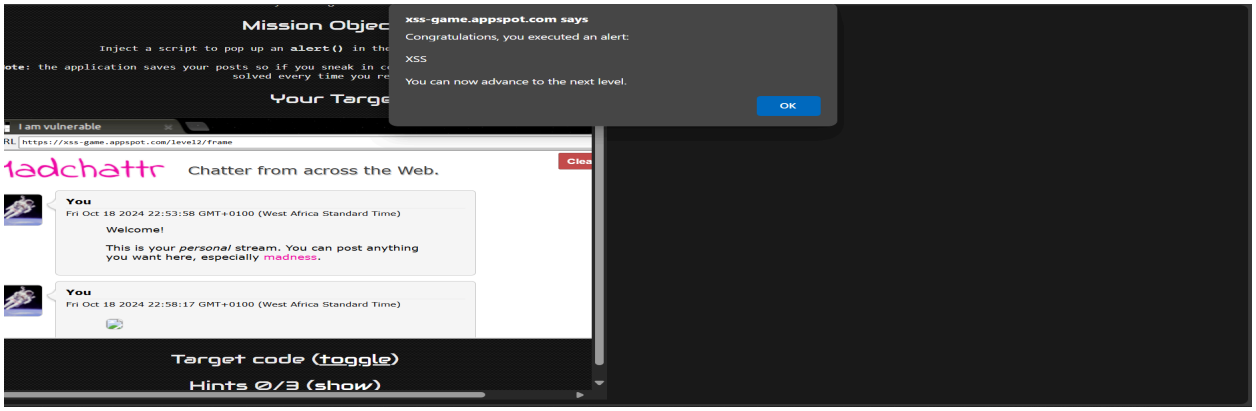
XXS is a common vulnerability in web applications and is frequently listed as a top vulnerability in the OWASP top ten. XXS occurs when web applications execute JavaScript, which is input into the form sections of a web application. The applications perform no security checks on the entered data. It simply passes it straight to the server, causing inputted JavaScript to execute.

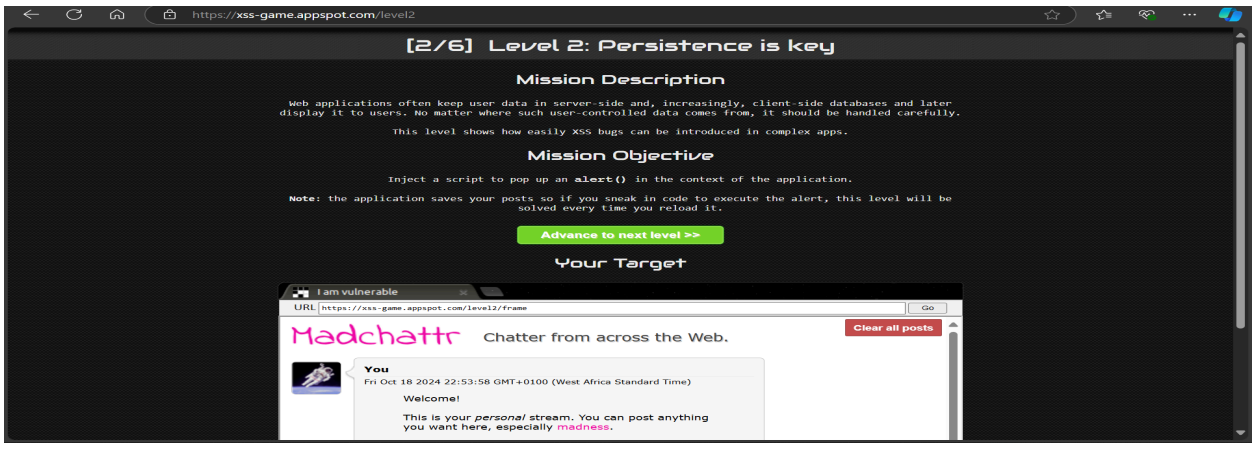
INPUT FROM WEBSITE :

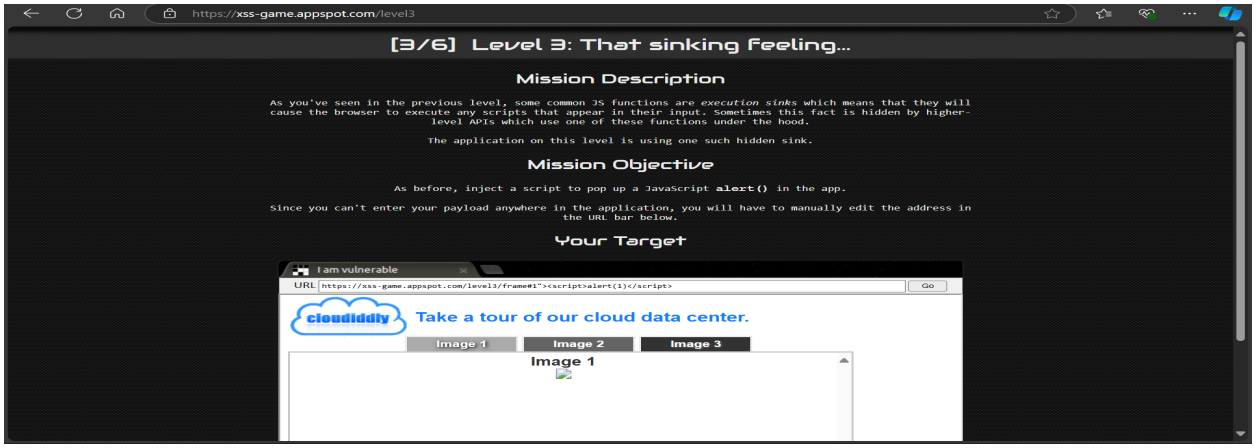


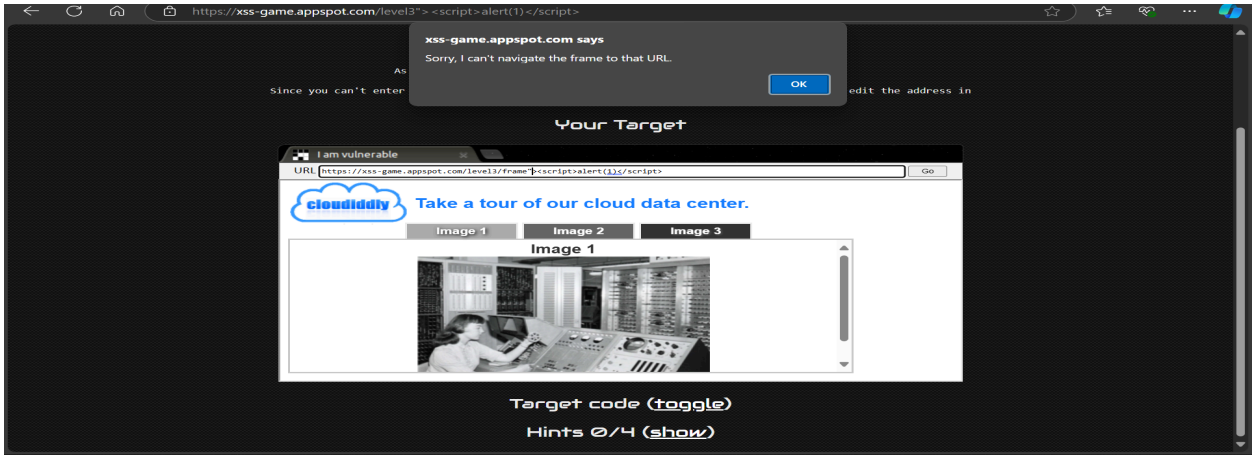

https://xss-game.appspot.com/level1
[1/6] Level 1: Hello, world of XSS
Mission Description
This level demonstrates a common cause of cross-site scripting where user input is directly included in the page without proper escaping.
Interact with the vulnerable application window below and find a way to make it execute JavaScript of your choosing. You can take actions inside the vulnerable window or directly edit its URL bar.
Mission Objective
Inject a script to pop up a JavaScript alert() in the frame below.
Once you show the alert you will be able to advance to the next level.
Advance to next level >>
Your Target
I am vulnerable
URL
Go
FourOrFour
Sorry, no results were found for
Try again.


https://xss-game.appspot.com/level2
[2/6] Level 2: Persistence is key
Mission Description
Web applications often keep user data in server-side and, increasingly, client-side databases and later display it to users. No matter where such user-controlled data comes from, it should be handled carefully.
This level shows how easily XSS bugs can be introduced in complex apps.
Mission Objective
Inject a script to pop up an alert() in the context of the application.
Note: the application saves your posts so if you sneak in code to execute the alert, this level will be solved every time you reload it.
Your Target
I am vulnerable
URL
Go
Madchattr
Chatter from across the Web.
Clear all posts
You
Fri Oct 18 2024 22:53:58 GMT+0100 (West Africa Standard Time)
Welcome!
This is your personal stream. You can post anything you want here, especially madness.


xss-game.appspot.com says
Congratulations, you executed an alert:
XSS
You can now advance to the next level.
OK
Mission Objec
Your Targe
I am vulnerable
Chatter from across the Web.
You
Fri Oct 18 2024 22:53:58 GMT+0100 (West Africa Standard Time)
Welcome!
This is your personal stream. You can post anything you want here, especially madness.
You
Fri Oct 18 2024 22:58:17 GMT+0100 (West Africa Standard Time)
Target code (toggle)
Hints 0/3 (show)


https://xss-game.appspot.com/level2
[2/6] Level 2: Persistence is key
Mission Description
Web applications often keep user data in server-side and, increasingly, client-side databases and later display it to users. No matter where such user-controlled data comes from, it should be handled carefully.
This level shows how easily XSS bugs can be introduced in complex apps.
Mission Objective
Inject a script to pop up an alert() in the context of the application.
Note: the application saves your posts so if you sneak in code to execute the alert, this level will be solved every time you reload it.
Advance to next level >>
Your Target
I am vulnerable
URL
Go
Madchattr
Chatter from across the Web.
Clear all posts
You
Fri Oct 18 2024 22:53:58 GMT+0100 (West Africa Standard Time)
Welcome!
This is your personal stream. You can post anything you want here, especially madness.





RESULT FOR CONDUCTING CROSS SITE SCRIPTING (XXS) ATTACK ON XSS-GAME.APPSPOT.COM

STEPS TAKEN :

After navigating to the website that we would be carrying the attack on we tried some words like [**HELLO]** on the search bar. This helps us know that the application is vulnerable to XXS, as our input is directly reflected in the output of the search result.

Now we want to execute the XXS attack, we start by entering this script in the search box {<script>alert('XSS')</script>} This will cause an alert text box to pop up on our screen with “1” on it. We have successfully executed an XXS attack.

Now we are presented with a forum page for level 2. The script we entered for level 1 will not work here. We need to first enter a HTML tag which will adopt the script we entered in level 1, so that every time this page is visited and the tag is loaded, the XXS attack will run. This is a method of achieving a persistent XXS

attack on a site. For this the script we would be using is {<img src=x onerror=alert('XSS')>} This bit of HTML is loading an image, which doesn’t exist into the forum. Every time there is an error, the JavaScript alert will run. Considering that the image doesn’t exist and that it will be loaded every time a user visits the forum, the JavaScript alert will always run.

To protect a website from XSS attacks:

1. Input Validation: Sanitize and whitelist user input to remove harmful characters.

2. Output Encoding: Encode data before rendering it on the page (HTML, JavaScript).

3. HTTP-Only and Secure Cookies: Use HttpOnly and Secure flags on cookies to prevent access via JavaScript and ensure they're sent over HTTPS.

4. Content Security Policy (CSP): Implement CSP to restrict script sources and prevent inline scripts.

5. Avoid Inline JavaScript: Use external scripts instead of inline event handlers.

6. Use Secure Frameworks: Leverage frameworks with built-in XSS protection.