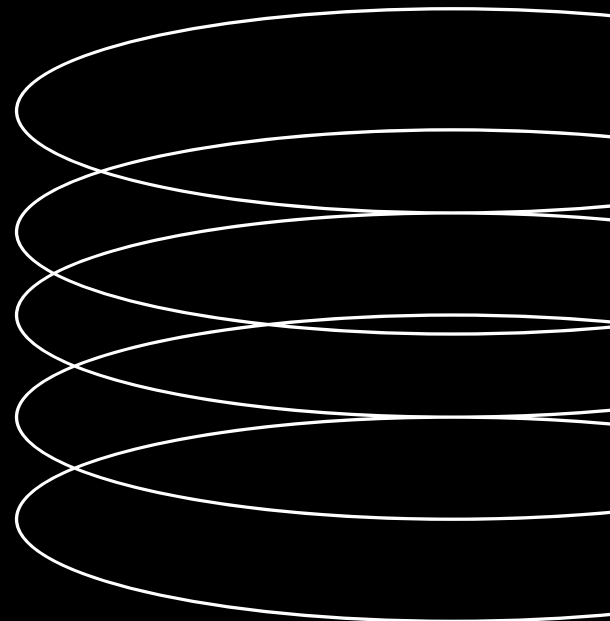


Traccia Bonus XSS Game



Level 1

XSS

Reflective

Procediamo alla visualizzazione del codice sorgente della pagina tramite tasto destro , view page source(oppure consultando semplicemente il source code che ci propone il gioco)

[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.

Your Target

I am vulnerable

URL

FourOrFour

Analizzando il source code possiamo notare la presenza di una vulnerabilità tramite la stringa di codice indicata di seguito:

```
def render_string(self, s):
    self.response.out.write(s)

def get(self):
    # Disable the reflected XSS filter for demonstration purposes
    self.response.headers.add_header("X-XSS-Protection", "0")

    if not self.request.get('query'):
        # Show main search page
        self.render_string(page_header + main_page_markup + page_footer)
    else:
        query = self.request.get('query', '[empty]')
```

La riga evidenziata disabilita il filtro XSS integrato nel browser, che di solito aiuta a prevenire alcuni attacchi XSS riflessi. Disabilitare questo filtro rende l'applicazione più vulnerabile agli attacchi

Proviamo quindi ad inserire all'interno del campo di testo la seguente stringa js:

<script>alert()</script>

The screenshot shows the 'xss-game.appspot.com' interface. A modal alert box is open, displaying 'Congratulations, you executed an alert!' and 'undefined'. The background shows a game level with a 'Mission Objective' to inject a script. The 'Your Target' section shows a browser window with the URL 'https://xss-game.appspot.com/level1/frame?query=<script>alert()</script>'. The 'Target code' section shows the HTML source of the page, including the 'X-XSS-Protection: 0' header.

In questo caso il codice javascript verrà eseguito nel contesto della pagina web , causando l'esecuzione dell>alert.

Level 2 XSS Stored

Nel secondo livello proviamo ad inserire lo stesso codice inserito nel livello 1. Come possiamo vedere non riusciamo a scatenare l>alert con la stessa facilità .

[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

Target code ([toggle](#))

Hints 0/3 ([show](#))

Procediamo quindi ad analizzare il codice sorgente fornito.

```
index.html level.py post-store.js

1 <!doctype html>
2 <html>
3   <head>
4     <!-- Internal game scripts/styles, mostly boring stuff -->
5     <script src="/static/game-frame.js"></script>
6     <link rel="stylesheet" href="/static/game-frame-styles.css" />
7
8     <!-- This is our database of messages -->
9     <script src="/static/post-store.js"></script>
10
11    <script>
12      var defaultMessage = "Welcome!<br><br>This is your <i>personal</i>"
13        + " stream. You can post anything you want here, especially "
14        + "<span style='color: #f00ba7'>madness</span>.";
15
16      var DB = new PostDB(defaultMessage);
17
18      function displayPosts() {
```

Hints 0/3 (show)

Possiamo provare ad iniettare un “onerror” dentro un immagine. Di seguito il codice inserito:

Creiamo un tag immagine specificando che se l’immagine “123” ci darà un errore(onerror), questo causerà un pop alert. Di seguito il risultato dell’alert:

