XSS

Cross Site Scripting

Sotiris Ftiakas : 3076

TABLE OF CONTENTS

01 About XXS

02 Non-Persistent XSS

03 Persistent XSS

04 What is Penetration Testing

05 Penetration Testing Phases

06 Basic Examples

About XSS

About XSS

Cross-Site Scripting (XSS) is a security vulnerability typically found in websites, where **JavaScript** is used to target the users of a website.

XSS attacks are a case of **code injection**. They enable attackers to inject malicious scripts into otherwise benign and trusted websites.

The actual attack occurs when the victim visits the web page or web application that executes the malicious code. The browser has no way to know that the script should not be trusted, and will execute the script.

About XSS

A cross-site scripting vulnerability may be used by attackers to bypass access controls such as the **same-origin policy**. This policy prevents a malicious script on one page from obtaining access to sensitive data on another web page.

A successful XSS attack might enable the attacker to:

- Impersonate or masquerade as the victim user.
- Carry out any action that the user is able to perform.
- Read any data that the user is able to access.
- Capture the user's login credentials.
- ...

About XSS

XSS vulnerabilities have been reported and exploited since the 1990s.

Today, it is the **most commonly found security vulnerability** in software.
It is still considered a major threat vector, with big companies offering bounties of several thousand dollars to hackers who find such system flaws.

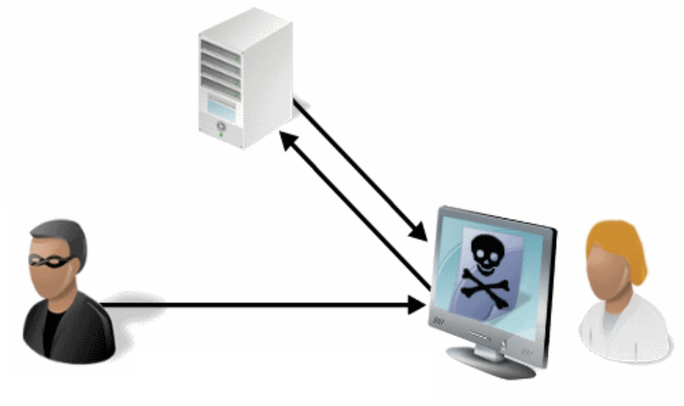
There is no single, standardized classification of cross-site scripting flaws, but most experts distinguish between two primary flavors of XSS flaws:
non-persistent and ***persistent*.**

Non-Persistent XSS

Non-Persistent XSS

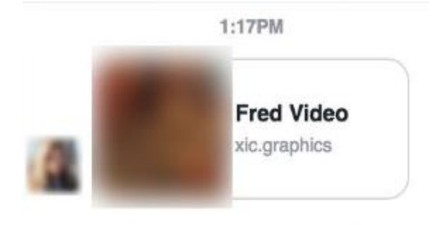
Non-Persistent XSS attacks, also known as **Reflected** attacks, occur when a script is activated through a link, which sends an XSS request to the vulnerable website, reflecting the attack back to the user.

To distribute the malicious link, a perpetrator typically embeds it into an email or third party website (e.g., in a comment section or in social media).



Non-Persistent XSS

Example:



1:17PM

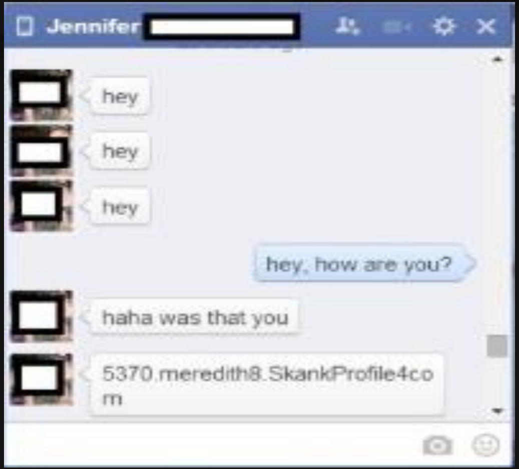
Fred Video

xic.graphics

Lets assume that an attacker sends the victim a misleading message with a link containing malicious JavaScript.

If the victim clicks on the link, the HTTP request is initiated from the victim's browser and sent to the vulnerable web application.

The malicious JavaScript is then reflected back to the victim's browser, where it is executed in the context of the victim user's session.



Jennifer [redacted]

hey

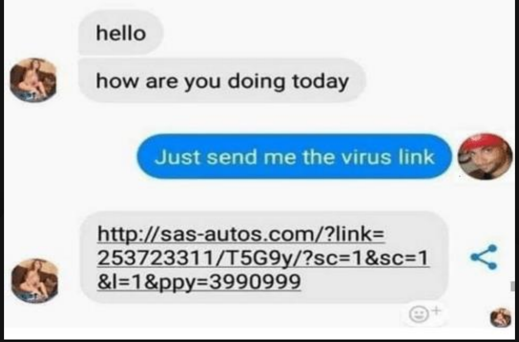
hey

hey

hey, how are you?

haha was that you

5370.meredith8.SkankProfile4com



hello

how are you doing today

Just send me the virus link

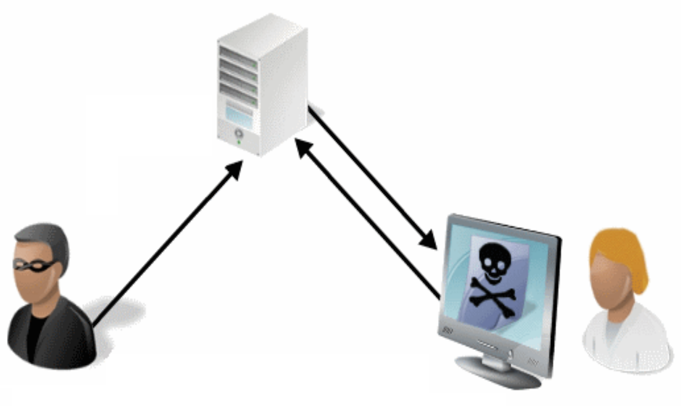
http://sas-autos.com/?link=253723311/T5G9y/?sc=1&sc=1&l=1&ppy=3990999

Persistent XSS

Persistent XSS

Persistent XSS attacks, also known as **Stored** attacks, are those where the injected script is **permanently** stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc.

The victim then retrieves the malicious script from the server when it requests the stored information.



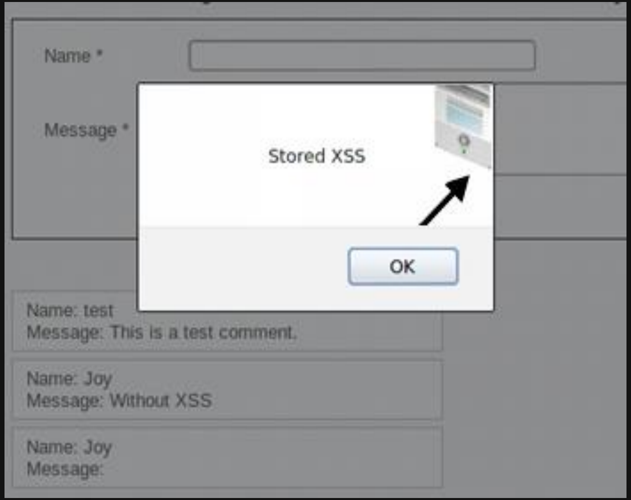
Persistent XSS

Example:

Consider a web application that allows users to enter a message that is displayed on each user’s profile page. The application stores each message in a local database.

A malicious user notices that the web application fails to sanitize the message field and inputs malicious JavaScript code as part of their message.

When other users view the messages board page, the malicious code automatically executes in the context of their session.



Stored XSS

Name: test
Message: This is a test comment.

Name: Joy
Message: Without XSS

Name: Joy
Message:

Impact of XSS

XSS effects vary in range from petty nuisance to significant security risk, depending on the sensitivity of the data handled by the vulnerable site.

Examples:

- In a brochureware application, where all users are anonymous and all information is public, the impact will often be minimal.
- In an application holding sensitive data, such as banking transactions, emails, or healthcare records, the impact will usually be serious.
- If the compromised user has elevated privileges within the application, then the impact will generally be critical, allowing the attacker to take full control of the vulnerable application and compromise all users and their data.

What is Penetration Testing

Penetration Testing

A **penetration test**, also known as a ***pen test***, ***pentest*** or ***ethical hacking***, is an authorized simulated cyberattack on a computer system, performed to evaluate the security of the system.

The test is performed to identify **weaknesses** (also referred to as vulnerabilities), including the potential for unauthorized parties to gain access to the system's features and data, as well as **strengths**, enabling a full risk assessment to be completed.

Penetration Testing

The process typically identifies the target systems and a particular goal, then reviews available information and undertakes various means to attain that goal.
A penetration test target may be:

- **Black Box:** Knowledge of internal working structure (Code) is not required for this type of testing. Only GUI (Graphical User Interface) is required for test cases.

- **White Box:** Knowledge of internal working structure (Coding of software) is necessarily required for this type of testing.

- **Grey Box:** Partially Knowledge of the internal working structure is required.

Penetraton Testing Phases

Pen Test Phases

The process of penetration testing can be broken down to **six (6)** phases:

1) Establish Goal - First, you need to establish your goal, e.g. breach a credit card database.

2) Reconnaissance – Gather important information on a target system, e.g. do research on the internet, find what servers belong to the organization, look up their social media, anything that will help better attack the target.

3) Scanning - Use technical tools to further your knowledge of the system, e.g. use tools to scan for open ports.

Pen Test Phases

4) Gaining Access - Use little pieces of software that automate attacks on known vulnerabilities in the network, in order to exploit and take control of the system.

5) Pivoting - Once you have exploited one vulnerability, you may try to gain access to other machines so the process repeats.

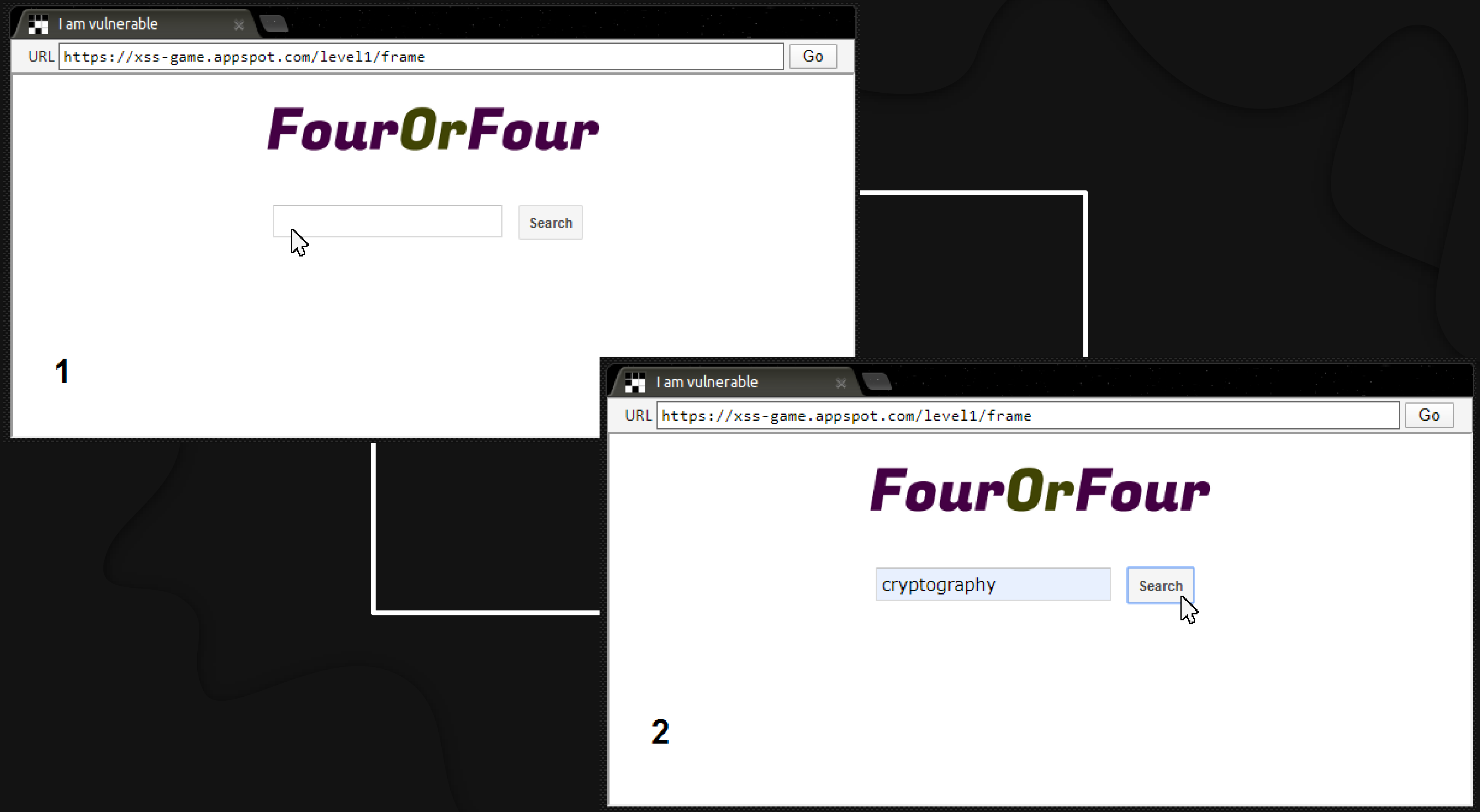
6) Covering Tracks – At the end of the attack, you must clear any trace of compromising the victim system, any type of data gathered, log events, in order to remain anonymous.

Basic Examples

xss-game.appspot.com

Ist Example

Non-Persistent XSS



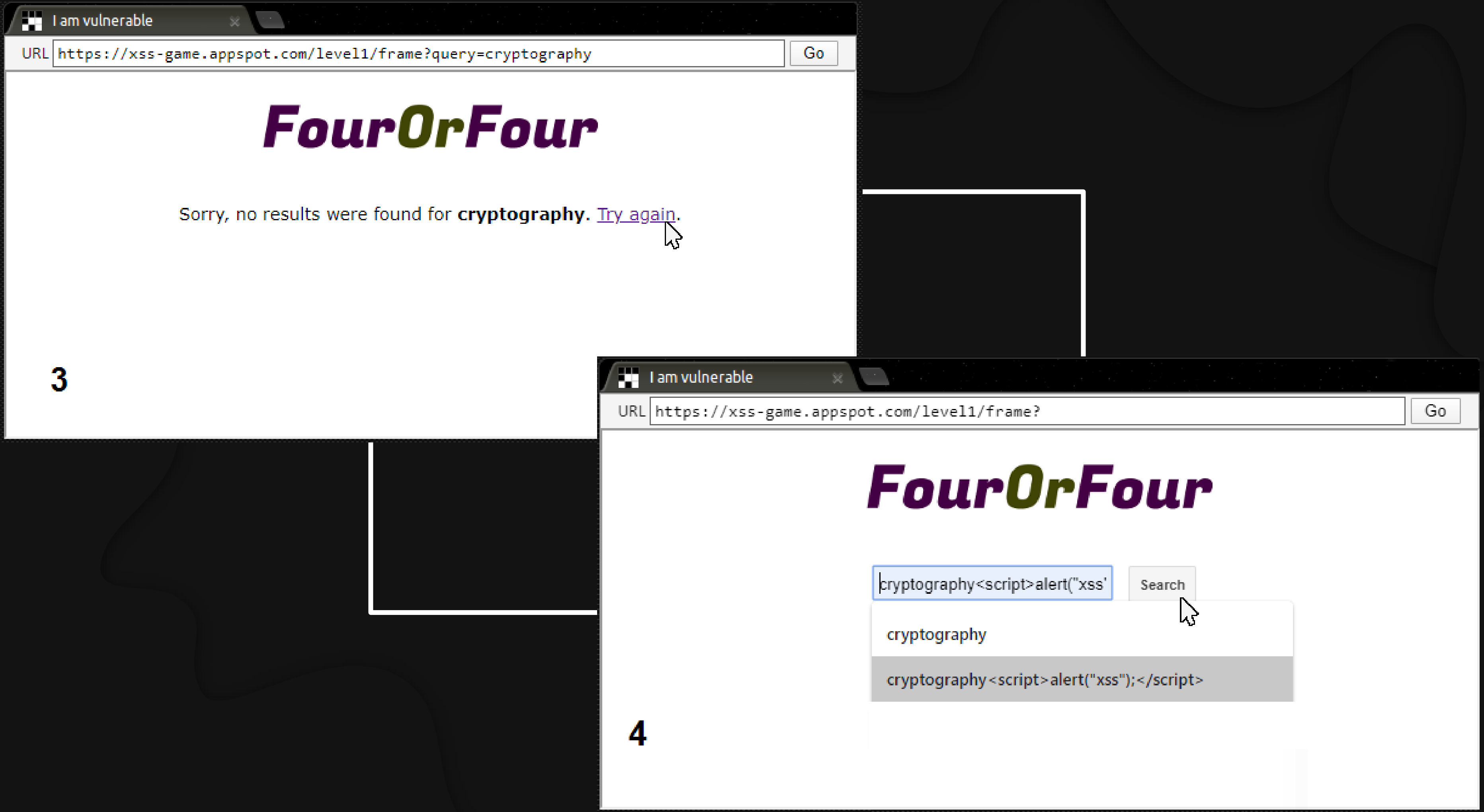
1

URL https://xss-game.appspot.com/level1/frame

2

URL https://xss-game.appspot.com/level1/frame

cryptography



https://xss-game.appspot.com/level1/frame?query=cryptography

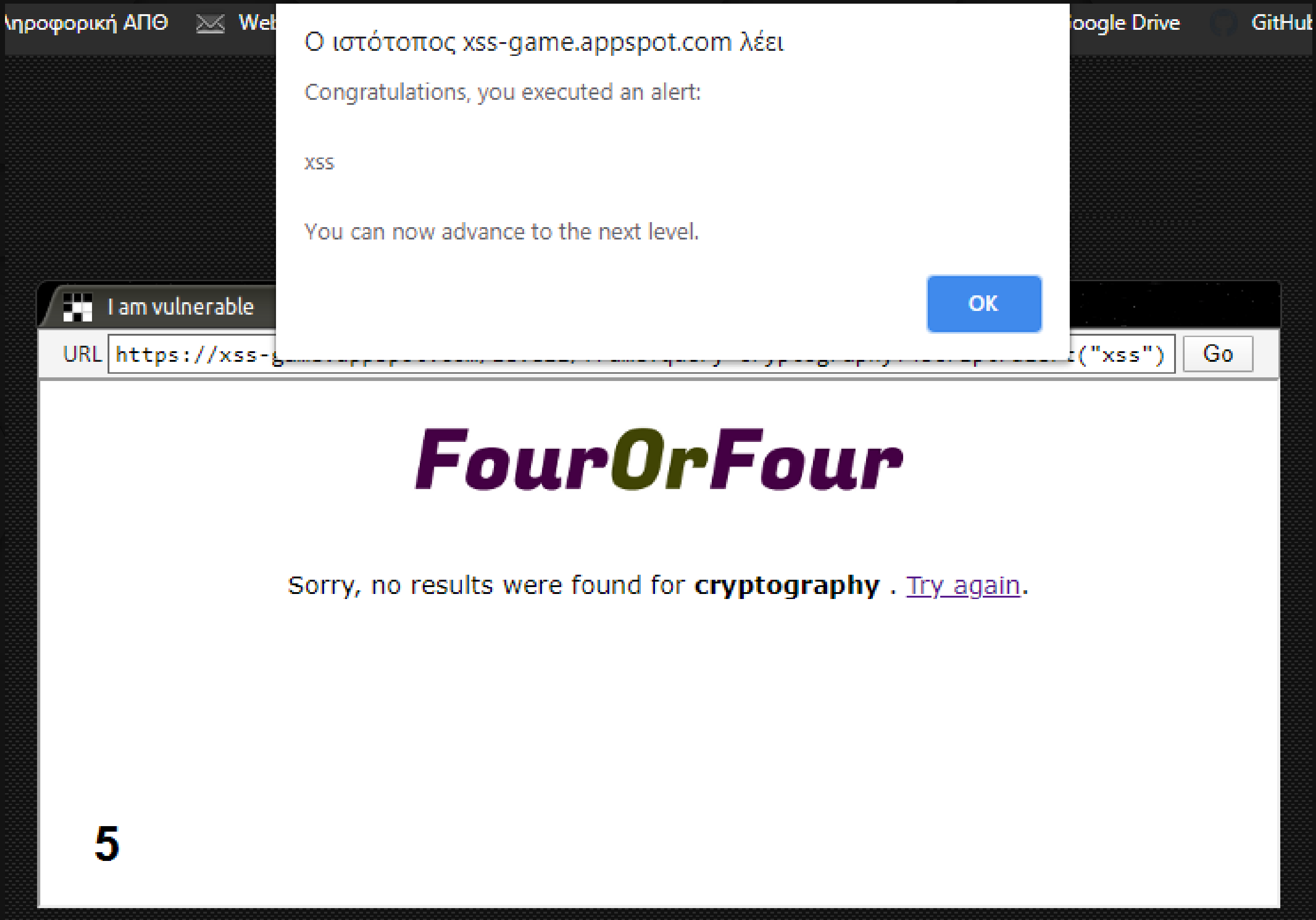
Sorry, no results were found for **cryptography**. Try again.

https://xss-game.appspot.com/level1/frame?

cryptography<script>alert("xss'

cryptography

cryptography<script>alert("xss");</script>



Ο ιστότοπος xss-game.appspot.com λέει

Congratulations, you executed an alert:

xss

You can now advance to the next level.

```
https://xss-g                                          ("xss")
```

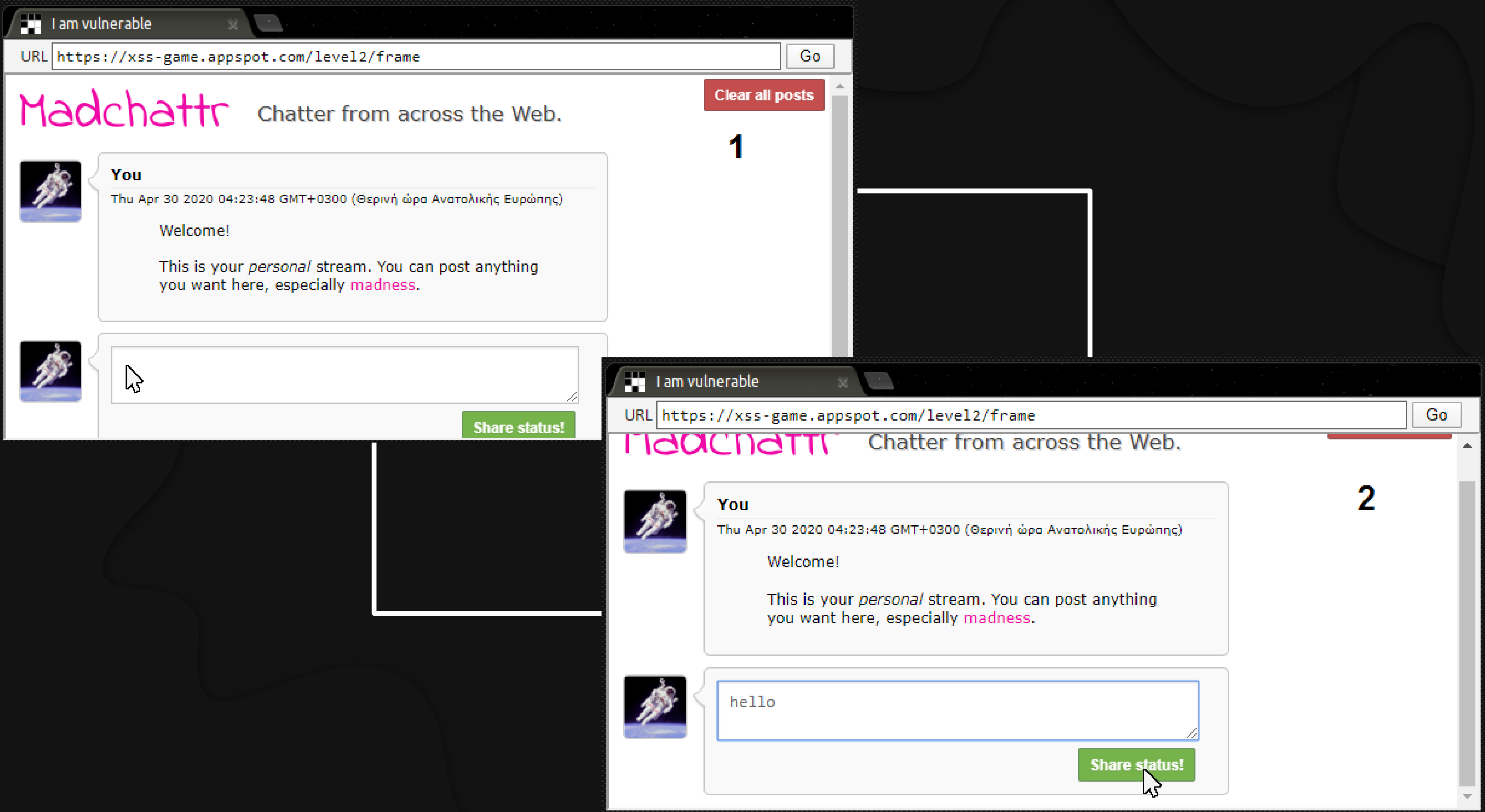
FourOrFour

Sorry, no results were found for **cryptography** . Try again.

5

2nd Example

Persistent XSS



https://xss-game.appspot.com/level2/frame

Chatter from across the Web.

You

Thu Apr 30 2020 04:23:48 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

Welcome!

This is your *personal* stream. You can post anything you want here, especially madness.

https://xss-game.appspot.com/level2/frame

Chatter from across the Web.

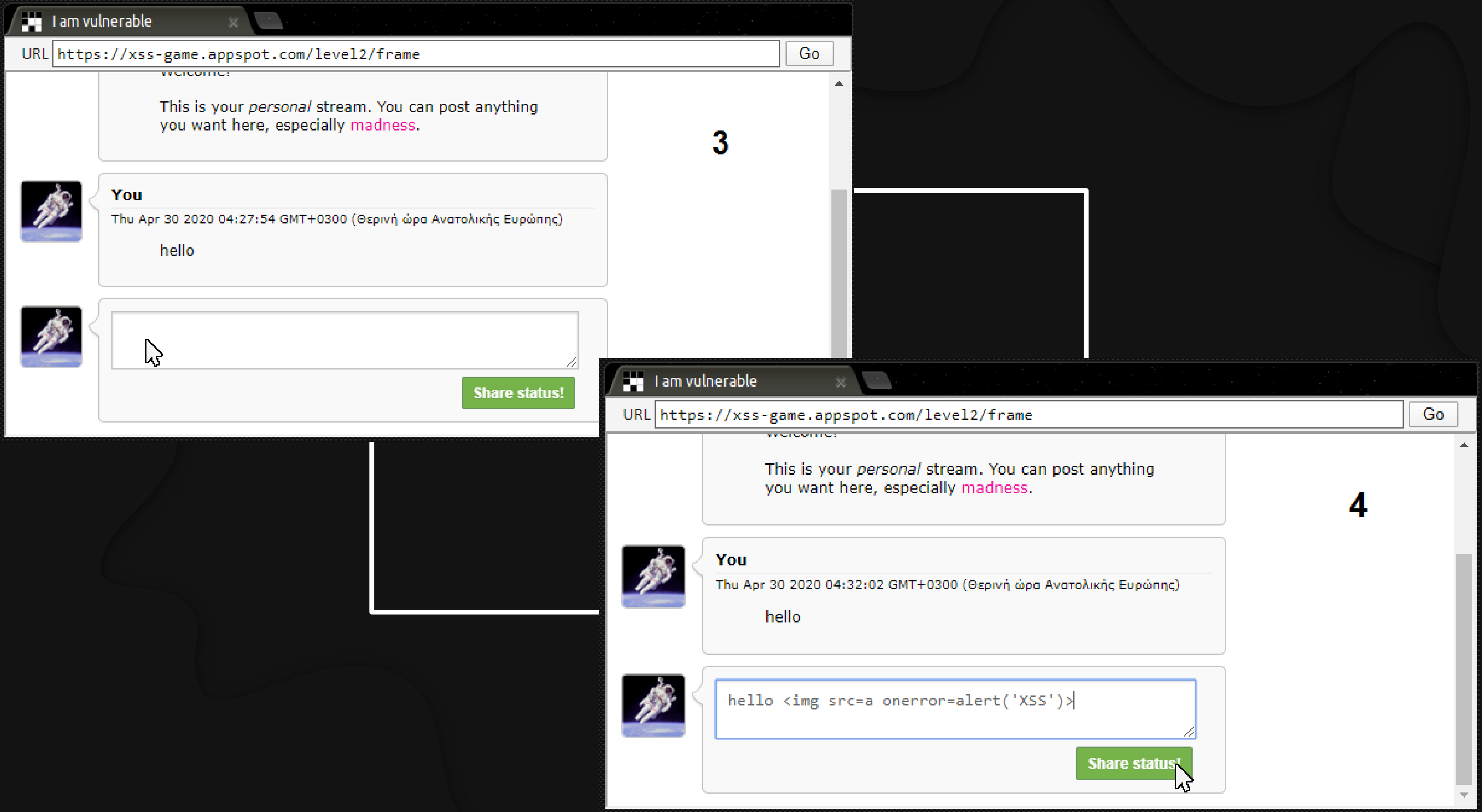
You

Thu Apr 30 2020 04:23:48 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

Welcome!

This is your *personal* stream. You can post anything you want here, especially madness.

hello



3
I am vulnerable
URL https://xss-game.appspot.com/level2/frame
Go

Welcome!

This is your *personal* stream. You can post anything you want here, especially madness.

You

Thu Apr 30 2020 04:27:54 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

hello

Share status!
4
I am vulnerable
URL https://xss-game.appspot.com/level2/frame
Go

Welcome!

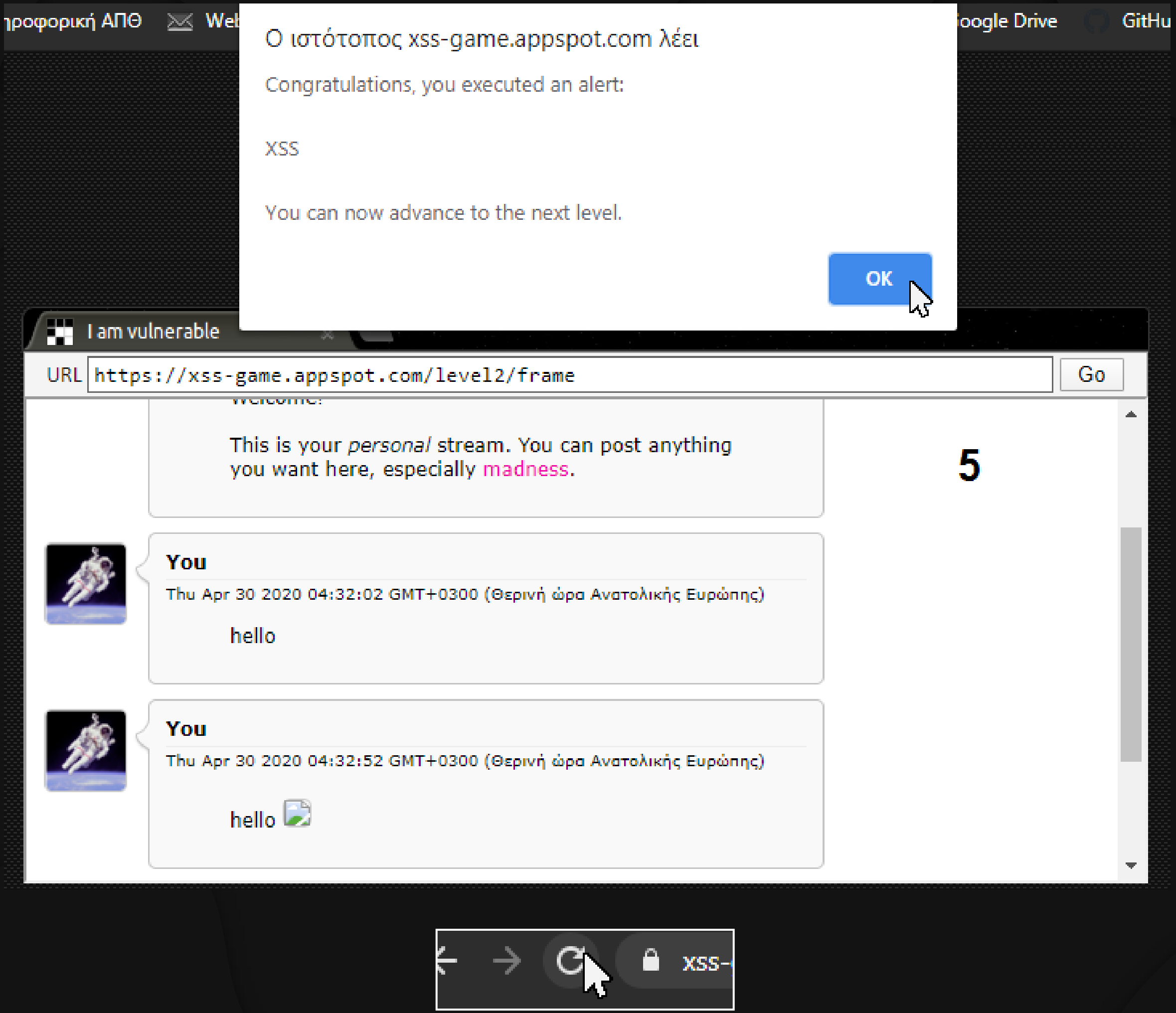
This is your *personal* stream. You can post anything you want here, especially madness.

You

Thu Apr 30 2020 04:32:02 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

hello

```
hello <img src=a onerror=alert('XSS')>
```



Ο ιστότοπος xss-game.appspot.com λέει

Congratulations, you executed an alert:

XSS

You can now advance to the next level.

https://xss-game.appspot.com/level2/frame

Welcome!

This is your *personal* stream. You can post anything you want here, especially madness.

5

You

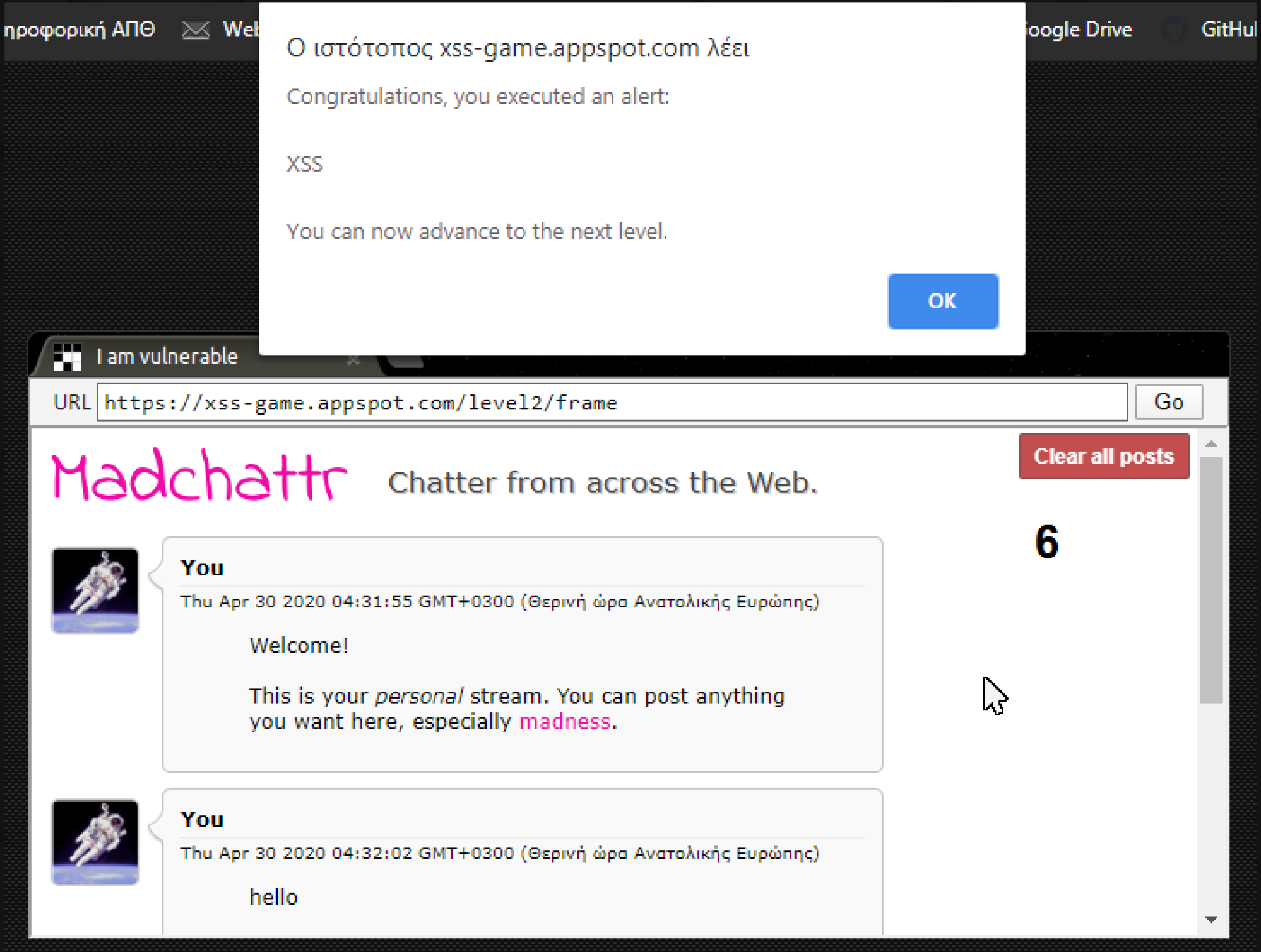
Thu Apr 30 2020 04:32:02 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

hello

You

Thu Apr 30 2020 04:32:52 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

hello



Ο ιστότοπος xss-game.appspot.com λέει

Congratulations, you executed an alert:

XSS

You can now advance to the next level.

URL https://xss-game.appspot.com/level2/frame

Chatter from across the Web.

You

Thu Apr 30 2020 04:31:55 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

Welcome!

This is your *personal* stream. You can post anything you want here, especially madness.

You

Thu Apr 30 2020 04:32:02 GMT+0300 (Θερινή ώρα Ανατολικής Ευρώπης)

hello

THANKS!

Does anyone have any questions?

https://www.acunetix.com/websitesecurity/cross-site-scripting/

https://portswigger.net/web-security/cross-site-scripting

https://en.wikipedia.org/wiki/Cross-site_scripting

https://en.wikipedia.org/wiki/Penetration_test

https://www.imperva.com/learn/application-security/reflected-xss-attacks/

https://www.veracode.com/security/xss

https://xss-game.appspot.com

CREDITS

- Presentation template by Slidesgo
- Icons by Flaticon
- Infographics by Freepik
- Images created by Freepik and rawpixel -Freepik
- Author introduction slide photo created by Freepik
- Text & Image slide photo created by Freepik.com