YesWeHack

#YWH-PGM2123-1786 BB Resolved

RCE on ROOT Page via VM ESCAPE

YesWeHack Dojo

Submitted by **alperkaya** on **2025-01-31**

REPORT DETAILS

7.5 HIGH
CVSS

Bug type	Code Injection (CWE-94)
CVE ID	
Impact	Remote Code Execution (RCE)
Scope	https://dojo-yeswehack.com/challenge-of-the-month/dojo-39
Endpoint	/
Severity	High
CVSS vector string	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Vulnerable part	post-parameter
Part name	INPUT_CODE
Payload	(this.constructor.constructor("return console.log(process.env.FLAG)"))()
Technical env.	Browser
App. fingerprint	

BUG DESCRIPTION

DESCRIPTION

First, here is the flag: FLAG{Y0u_ju5t_h4v3_t0_dive_d33p}

Because of the unsafe usage of vm module of NodeJS, any attacker can achieve Remote Code Execution without needing any sort of authentication or previous steps. To hit the code branch which executes attacker code inside an unsafe vm, you have to use PunyCode to bypass hostname check at WebsiteIsAvailable function and the if block which is just under the first call of this function. The first issue is, the website variable is modified after the check. It should have been the opposite, any modification should be done before checking it against a whitelist. The second issue is assuming that NodeJS vms are safe, infact even documentation is warning people that it's not safe to execute user input with vm module .

EXPLOITATION

While executing the user code, an empty object is given as a context. And even if its empty, every object has default attributes and functions. One of which is the constructor attribute. And it is surprisingly possible that by chaining constructor attributes, a code fragment can be executed outside the vm. And this allows any attacker to run their code outside the vm and access environment variables which otherwise shouldn't be possible.

POC

Here is the PoC code fragment:

```
(this.constructor.constructor("return console.log(process.env.FLAG)"))()
```

Note: (function declaration)() is a self-executing function.