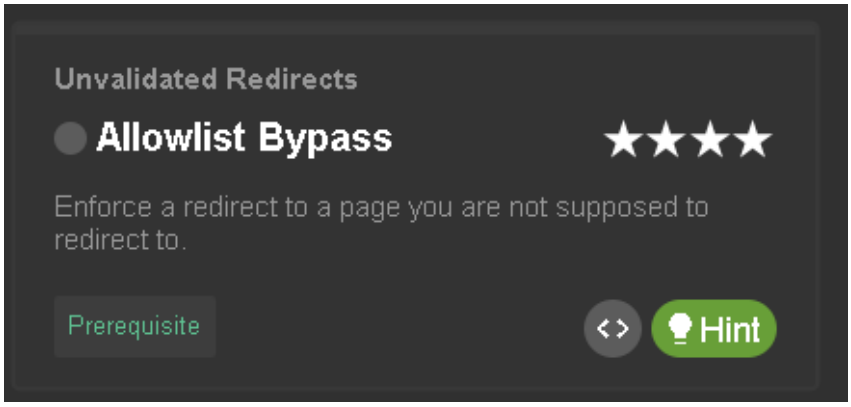
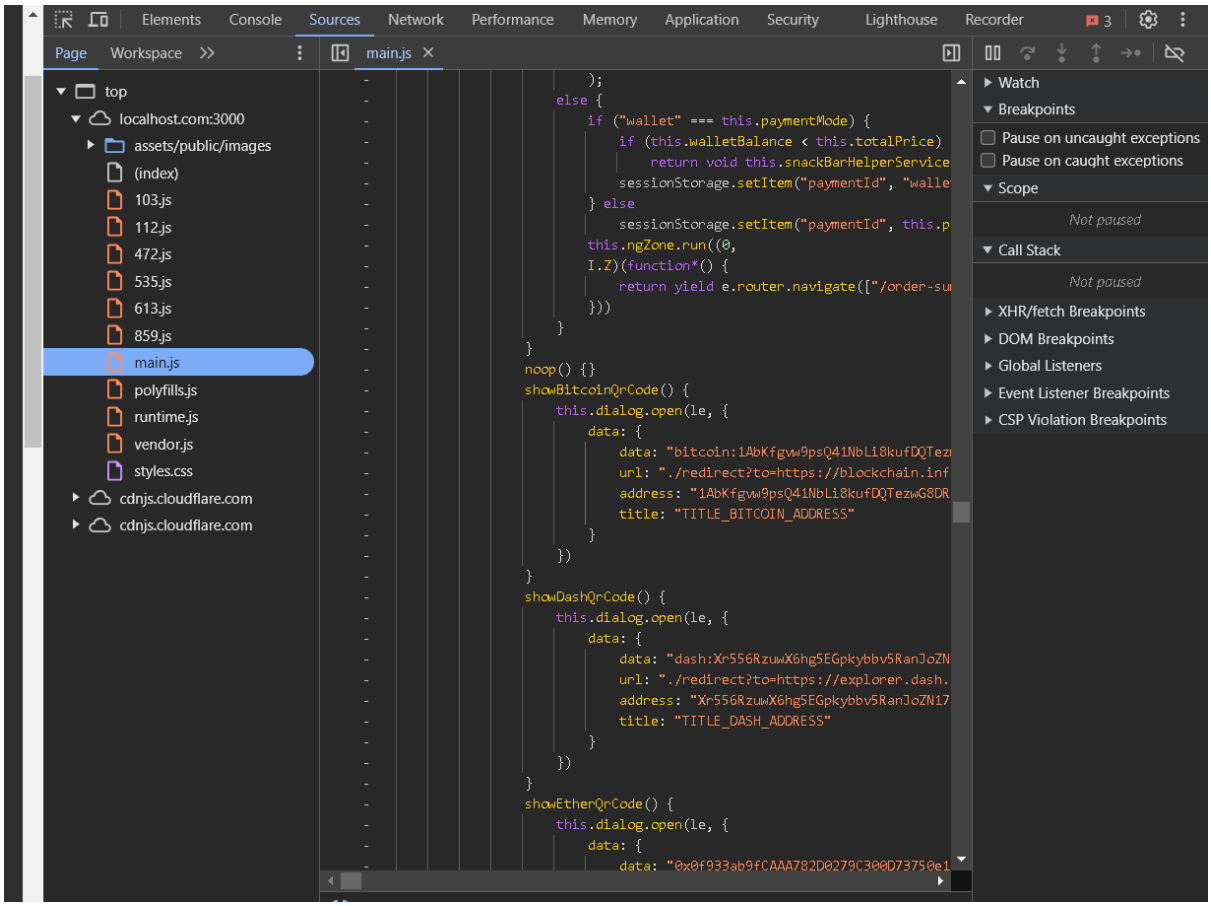
Unvalidated Redirects in OWASP Juice Shop



The objective is to enforce a redirect to a page that the application is not supposed to redirect to, effectively bypassing the allowlist restrictions

Go to the developer tools by clicking inspect In the owasp juice shop page



in the sources section, in main.js

check for the redirection endpoint at /redirect?to=URL, which is intended to allow redirects only to URLs specified in its allowlist.

The challenge is to find a way to bypass this allowlist and redirect to an arbitrary URL.

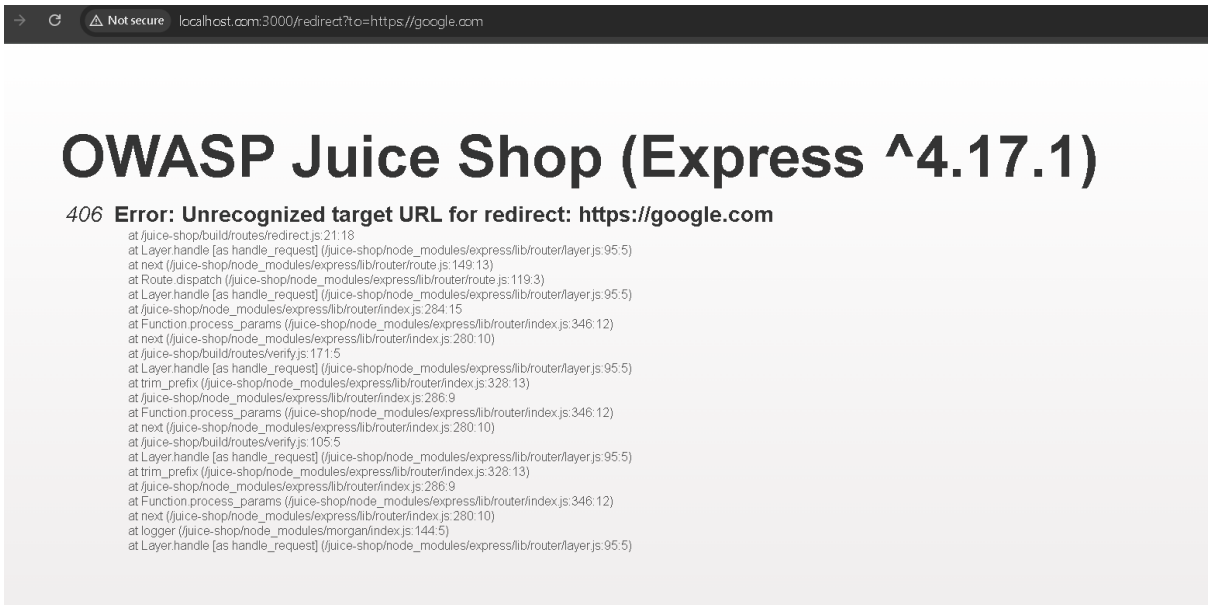
construct a URL that manipulates the redirection logic to bypass the allowlist.

using URL encoding

from the analysis
/redirect?to=https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm

is allowed url

encoding google.com
http://localhost.com:3000/redirect?to=https://google.com

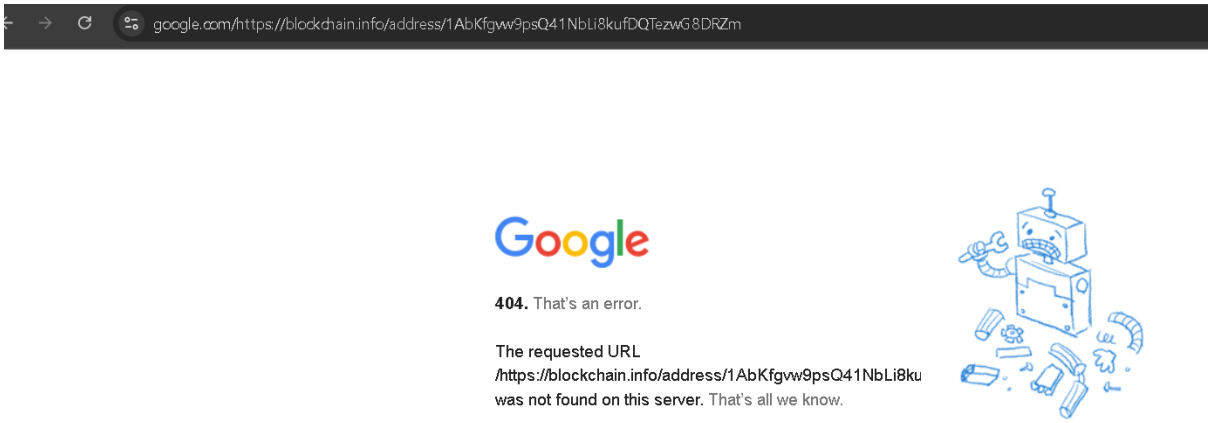


google.com is not allowed so we need to enode this url with the allowlist links to bypass it

using

http://localhost.com:3000/redirect?to=https://google.com/https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm

here we encoded google.com in the allowlist url



now it redirected to the google page

we bypassed the allowlist of owasp juice shop

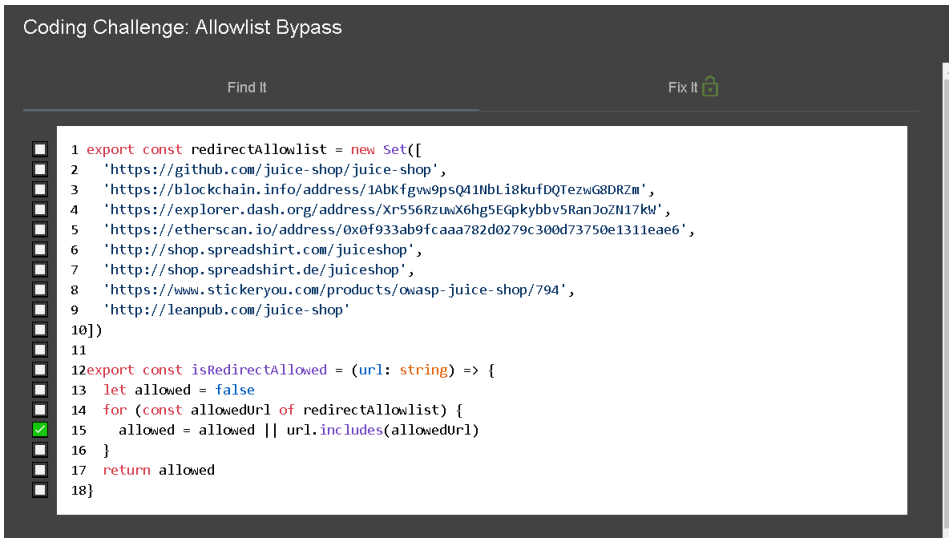
Preventive Measures:

To prevent such vulnerabilities in real-world applications:

- **Implement Strict Allowlist Validation:** Ensure that the allowlist validation logic is robust and cannot be bypassed through encoding tricks or parsing anomalies.
- **Use Built-in URL Parsers:** Utilize standard libraries or built-in functions to parse and validate URLs, reducing the risk of custom parsing errors.
- **Conduct Regular Security Testing:** Regularly test redirection mechanisms for potential bypass techniques and update the allowlist logic as needed.

Patching the vulnerability

The vulnerability is becz of this line



To fix

Coding Challenge: Allowlist Bypass

```
export const redirectAllowlist = new Set([
  'https://github.com/juice-shop/juice-shop',
  'https://blockchain.info/address/1AbKfgvw9psQ41NbLi8kufDQTezwG8DRZm',
  'https://explorer.dash.org/address/Xr556RzuwX6hg5EGpkybbv5RanJoZN17kW',
  'https://etherscan.io/address/0x0f933ab9fcaaa782d0279c300d73750e1311eae6',
  'http://shop.spreadshirt.com/juiceshop',
  'http://shop.spreadshirt.de/juiceshop',
  'https://www.stickeryou.com/products/owasp-juice-shop/794',
  'http://leanpub.com/juice-shop'
 ])

 export const isRedirectAllowed = (url: string) => {
  let allowed = false
  for (const allowedUrl of redirectAllowlist) {
-   allowed = allowed || url.includes(allowedUrl)
+   allowed = allowed || url === allowedUrl
  }
  return allowed
 }
```