**WIKI.JS**

CVE-2021-43856

Author:h1biki

|  |  |
| --- | --- |
| Name\_zh | Wiki.js 跨站脚本漏洞 |
| Name\_en | Wiki.js-Cross-site Scripting |
| CVE | CVE-2021-43856 |
| CVSS评分 | Medium (4.6) |
| 威胁等级 | Medium |
| CNNVD | CNNVD-202112-2628 |
| 其他id | CWE-79 |
| 受影响软件 | Wiki.js |

简介

Wiki.js是Requarks.io团队的一套基于Node.js并使用JavaScript语言编写的开源Wiki软件。 Requarks Wiki.js 存在跨站脚本漏洞，该漏洞源于通过创建在浏览器中查看时可以执行内联 JS 的恶意文件，恶意 Wiki.js 用户可能会进行存储的跨站点脚本攻击。这允许攻击者在其他用户直接查看文件时执行恶意 JavaScript。该文件必须由用户直接打开，不会在普通的 Wiki.js 页面中直接触发。wiki.js的优势：1、基于nodejs 2、支持markdown 3、美观简单

Wiki.js is a wiki app built on Node.js. Wiki.js 2.5.263 and earlier is vulnerable to stored cross-site scripting through non-image file uploads for file types that can be viewed directly inline in the browser. By creating a malicious file which can execute inline JS when viewed in the browser (e.g. XML files), a malicious Wiki.js user may stage a stored cross-site scripting attack. This allows the attacker to execute malicious JavaScript when the file is viewed directly by other users. The file must be opened directly by the user and will not trigger directly in a normal Wiki.js page. A patch in version 2.5.264 fixes this vulnerability by adding an optional (enabled by default) force download flag to all non-image file types, preventing the file from being viewed inline in the browser. As a workaround, disable file upload for all non-trusted users

漏洞影响

Wiki.js <2.5.263

漏洞复现

实验环境

准备两台虚拟机

kali.2020 192.168.160.128

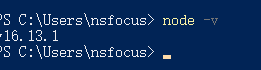
Windows 10 192.168.160.1

Wiki.js： 2.5.262

nodejs 16.13.1

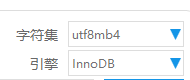
接下来利用这两台主机进行试验

Wiki.js要求我们用到nodejs的环境, node.js 10.12以上版本都可以；



注意要运行Wiki.JS的话要有对应的数据库,如今wiki.js支持以下数据库  
PostgreSQL 9.5 或更高版本  
MySQL 8.0 或更高版本 （部分支持MySQL 5.7.8）  
MariaDB 10.2.7 或更高版本  
MS SQL Server 2012 或更高版本  
SQLite 3.9 或更高版本

因为wikijs使用的字符长度是4字节的宽字节,我们要注意把mysql数据库的字符集从utf8改为utf8mb4,引擎改为innoDB,

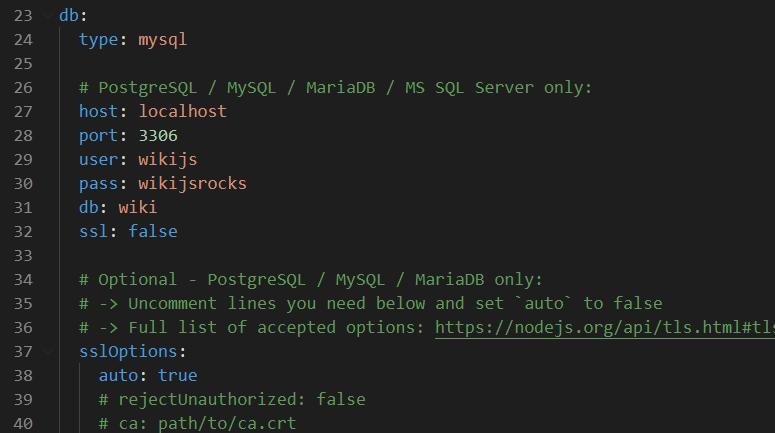


我们在官网下载具有漏洞的版本；https://github.com/Requarks/wiki

直接右键解压缩，右键重命名；config.sample.yml 修改成 config.yml

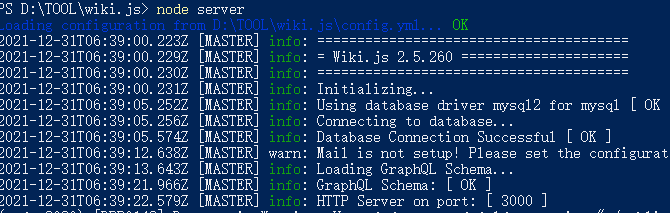
然后我们需要修改他的配置文件让他跟我们的mysql数据库配置相同

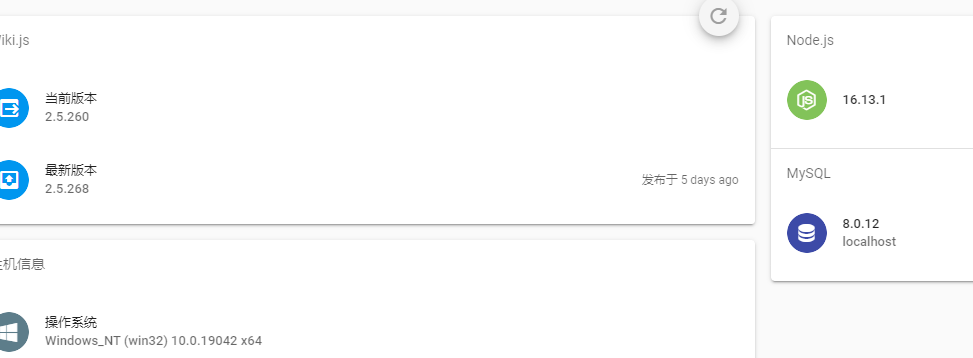
端口号默认是3000，可以不用改，主要是数据库配置。注意配置type中数据库的大小写要与要求的一致



node启动服务

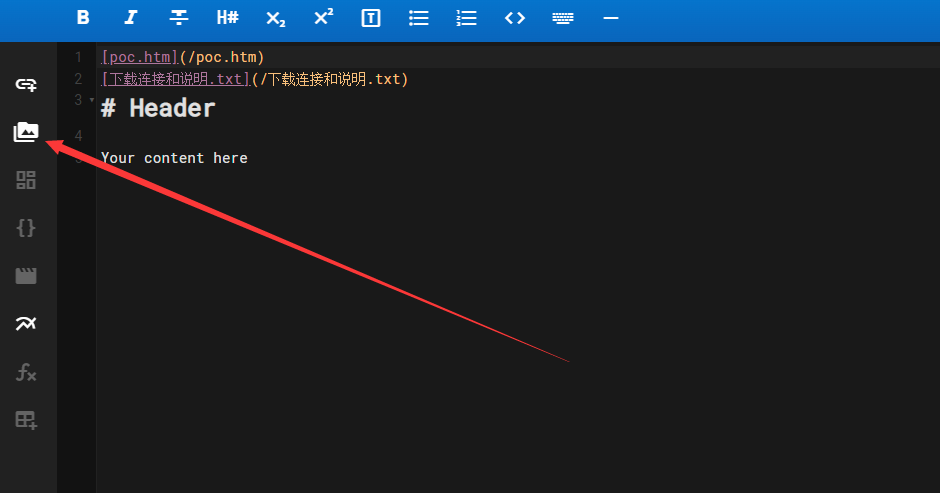
打开powershell，切换到wiki.js目录 启动node server

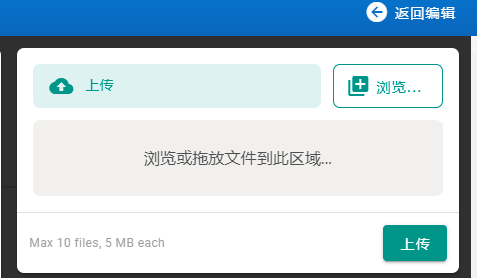




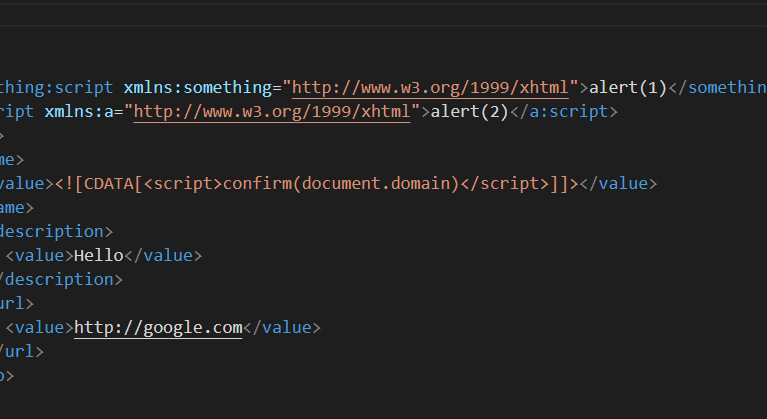
漏洞触发

漏洞点是在页面设置的插入文件功能中

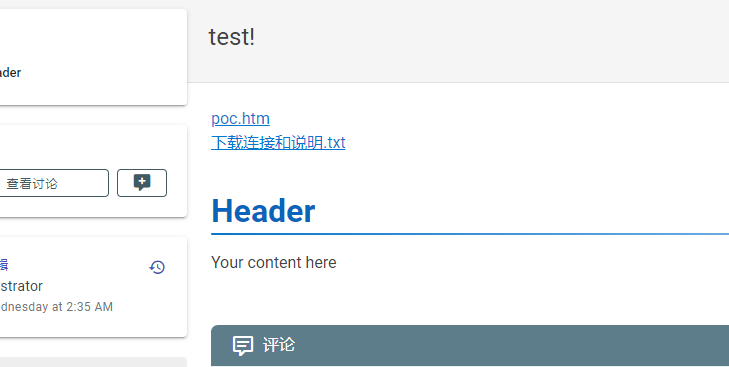




我们可以设置一个利用的文件来弹出我们的cookie



设置后页面内容会是这样



我们只要点击了页面上的poc.htm链接, 浏览器就会在查看时执行内联 JS ,从而触发恶意文件来达成我们的XSS



漏洞分析

代码的漏洞点是在源代码的client/components/admin/admin-security.vue 中,

漏洞出现的主要原因是没有检查来防止上传此类文件。导致发生了内联xss



可以看到在源代码中是没有对上传的文件做出过滤和检验的,但是在浏览器中查看时可以执行内联 JS 的恶意文件

在最新的版本中已经添加了对文件检测的功能防止内联注入



攻击检测

要注意检测在内联的 JavaScript 中，拼接的数据可以突破了原本的限制（字符串，变量，方法名等）。

标签属性中，恶意内容包含引号，从而突破属性值的限制，注入其他属性或者标签。

在设置文件上传时要注意设置合适的黑白名单如png', '.apng', '.jpg', '.jpeg', '.gif',

注意检测DOM 中的内联事件监听器，如 location、onclick、onerror、onload、onmouseover等， 标签的href属性，JavaScript 的eval()、setTimeout()、setInterval()等，都能把字符串作为代码运行

修复建议

只允许上传图片文件（png / html / webp / svg）

如果不允许文件类型限制，建议：

对于选择文件（.png、.jpg、.svg（清理后）、.pdf、.gif ...等）：

使用 Content-Disposition: 内联标题，以便可以在浏览器中查看它们。

如果文件扩展名与安全文件类型不匹配

使用 Content-Disposition: 附件标题，以便将它们直接下载到用户计算机。