

Electron Security



`allowRunningInsecureContent`

NOP Team



allowRunningInsecureContent

0x01 简介

大家好，今天和大家讨论的是 `allowRunningInsecureContent` 安全配置选项，这个选项非常容易理解，就是是否允许在 `HTTPS` 的网站加载或执行 `HTTP` 协议的 `JavaScript`、`CSS`、插件等

从 `Electron 2.0.0` 开始默认为 `false`，即不允许在 `HTTPS` 网站中加载或执行 `HTTP` 协议的内容

当 `webSecurity` 被设置为 `false` 时，会自动将 `allowRunningInsecureContent` 设置为 `true`

0x02 扩展探索

既然不允许 `HTTPS` 的网站加载 `HTTP`，是否会允许 `loadFile` 加载本地文件创建窗口加载 `HTTP` 的资源呢？

1. 正常功能测试

我们得找一个 `HTTPS` 的网站，并且还在网页内引用了 `HTTP` 的 `JavaScript` 等，搭建起来会比较麻烦，因为要有有效证书，还有配置同源策略等，我也不希望自签名证书通过自定义配置通过后再影响实验，因此通过 `fofa` 平台进行搜索

```
body="<script src=\"http://\" && port=\"443\" && country=\"CN\""
```

FQFA

body="<script src='http://' && port='443' && country='CN'"

会员 支持及工具 登录

端口排名

443 810,479

Server排名

nginx 494,891

Apache 53,343

wts/1.7.0 19,149

Microsoft-IIS/7.5 17,251

网站标题排名

官网 86,429

MXC抹茶交易所MEXC... 79,129

KAIYUN平台-欢迎光临 26,117

国内值得信赖的助学机... 25,763

https://106.14.76.175

106.14.76.175

中国 / 上海市 / Shanghai

ASN: 37963

组织: Hangzhou Alibaba Advertising Co.,...

2024-04-25

nginx/1.16.1

Header

Products

Connection: close

Content-Length: 2075

Accept-Ranges: bytes

Content-Type: text/html

Date: Thu, 25 Apr 2024 05:10:03 GMT

Etag: "6618ccb3-81b"

Last-Modified: Fri, 12 Apr 2024 05:54:59 GMT

Server: nginx/1.16.1

Vary: Accept-Encoding

+ Certificate

18684...

https://iqinban.com

琴伴, 你身边的练琴专家

47.103.146.228

中国 / 浙江省 / Hangzhou

ASN: 37963

组织: Hangzhou Alibaba Advertising Co.,...

iqinban.com

2024-04-25

nginx/1.17.3

Header

Products

HTTP/1.1 200 OK

Connection: close

Content-Length: 16725

Accept-Ranges: bytes

Content-Type: text/html; charset=utf8

Date: Thu, 25 Apr 2024 05:07:29 GMT

Etag: "649e4724-4155"

Last-Modified: Fri, 30 Jun 2023 03:08:20 GMT

Server: nginx/1.17.3

-1892...

选择这个 琴伴 作为案例

https://iqinban.com/

view-source:https://iqinban.com

http://

0/2

</div>

</div>

<!-- footer-底部 -->

<div class="footer_box">

<div class="footer_box_top">

<div class="footer_box_top_lianxi">

<h3>联系我们</h3>

地址: 中国 (上海) 自由贸易试验区芳春路400号1幢3层

邮箱: iqinban@sh-intellilink.cn

<div>电话: </div>

<div>

<p>+86 139-0180-5167 (高先生) </p>

</div>

<div class="footer_box_top_code">

<div class="footer_box_top_code_erwei">

<p>iPad学生端</p>

</div>

<div class="footer_box_top_code_erwei">

<p>iPhone学生端</p>

</div>

<div class="footer_box_top_code_erwei">

<p>安卓学生端</p>

</div>

<div class="footer_box_top_code_erwei">

<p>安卓教师端</p>

</div>

</div>

<div class="footer_box_bottom">

<p>©上海智凌信息技术有限公司 沪ICP备15032675号-2</p>

</div>

<div class="MessageBox" id="MessageBox_s">

<p id="conter"></p>

</div>

</div>

</body>

</html>

<script src="js/banner.js"></script>

<script src="http://www.jq22.com/jquery/jquery-1.10.2.js"></script>

<script src="js/jquery.waypoints.min.js"></script>

<script type="text/javascript" src="js/jquery.countup.min.js"></script>

<script type="text/javascript">

\$(function(){

\$('.counter').countUp({

delay: 10,

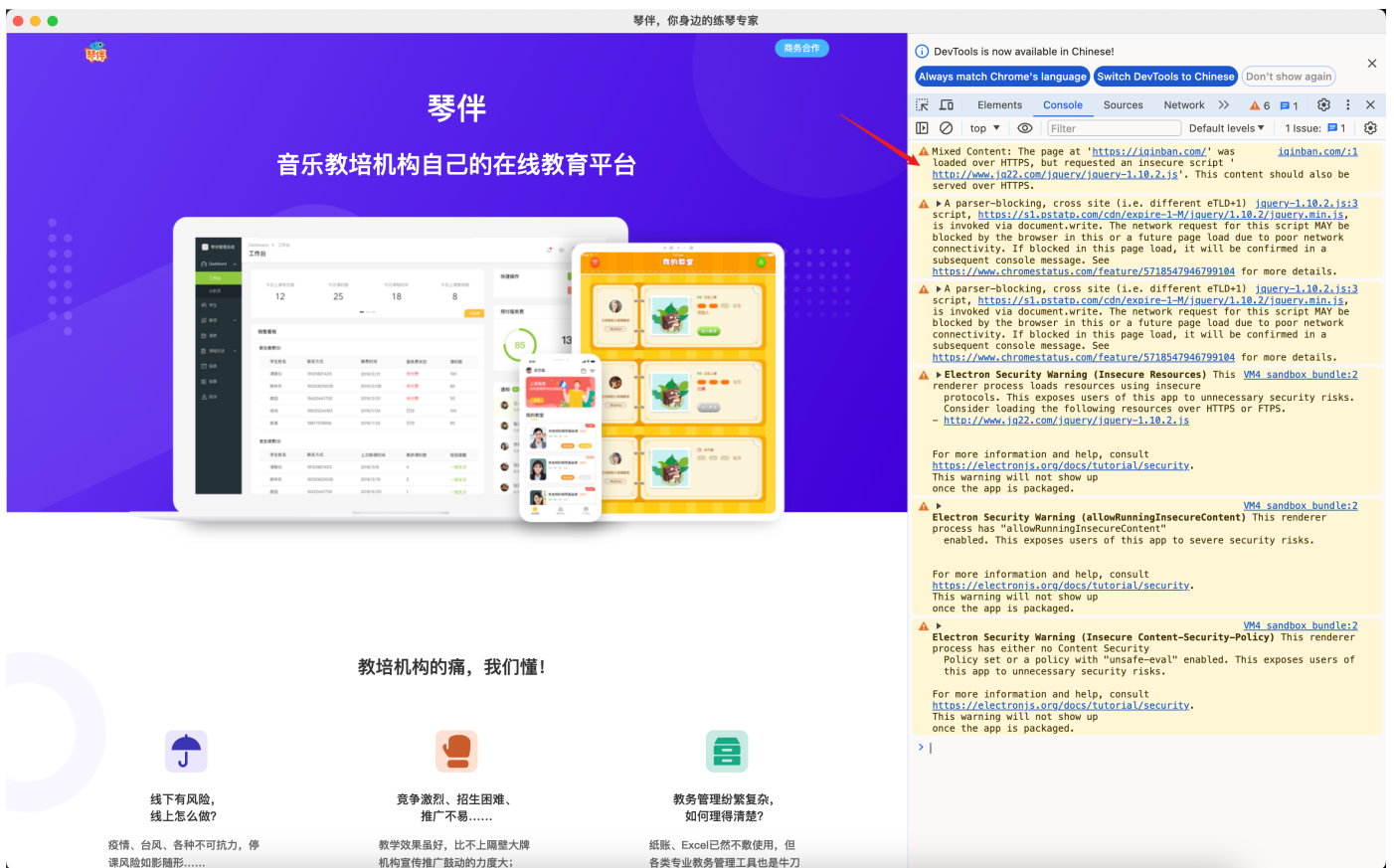
time: 2000

});

尝试默认情况下远程加载该网页创建窗口



在 Electron 31.0.0.alpha.2 中是会拦截的, 设置 `allowRunningInsecureContent` 为 `true` 后, 再次加载



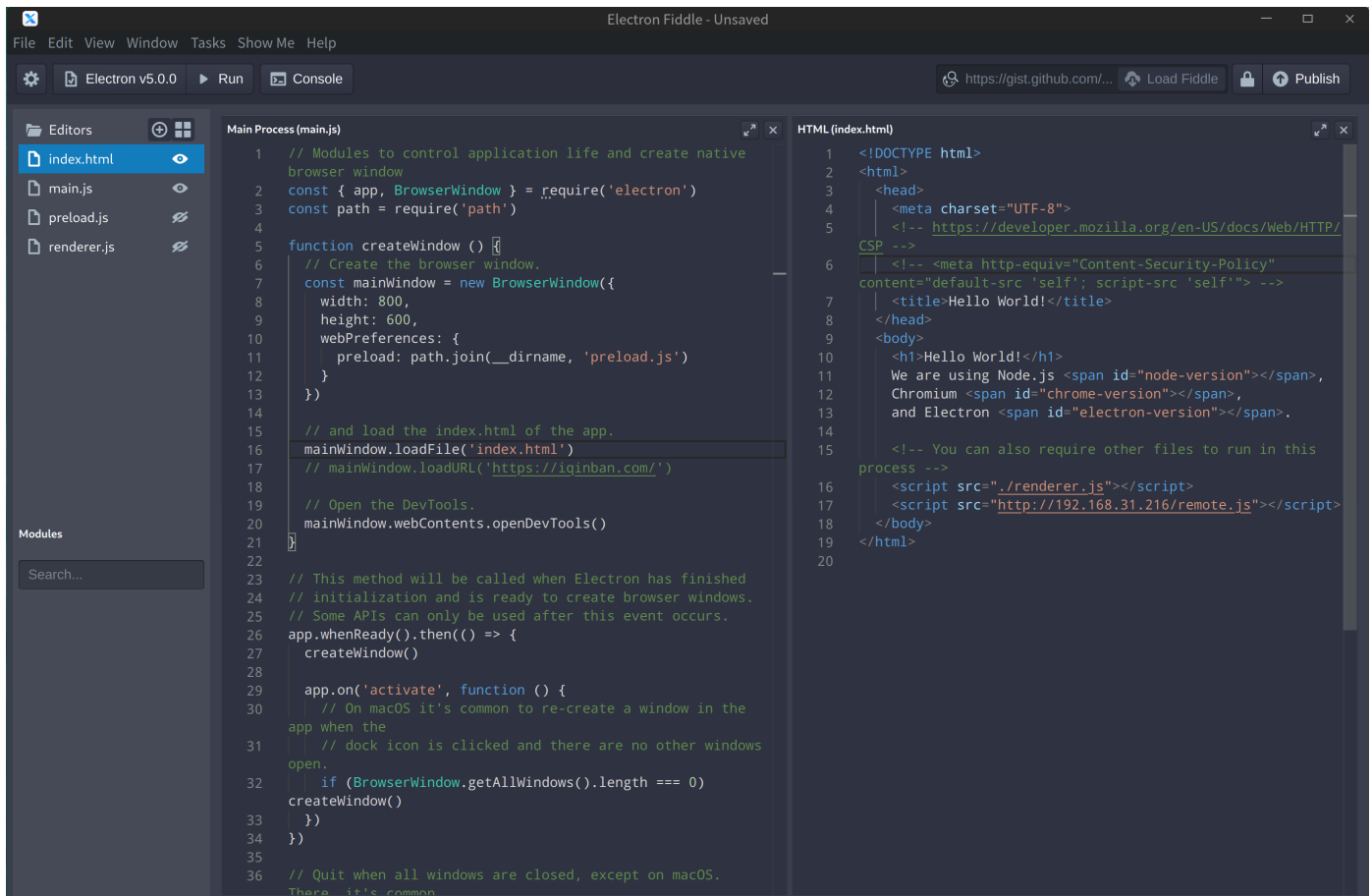
这回就可以加载了, 我们尝试在各个版本的 Electron 中进行尝试, 在 Electron 5.0、10.0、20.0、30.0 版本中执行结果保持一致

2. 本地加载文件测试

尝试在通过加载本地文件创建窗口情况下加载 HTTP 资源
搭建 xss 服务器

```
[~/D/t/21 >>> cat remote.js
alert('remote xss is running')
[~/D/t/21 >>> sudo python3 -m http.server 80
[Password:
Serving HTTP on :: port 80 (http://[::]:80/) ...
```

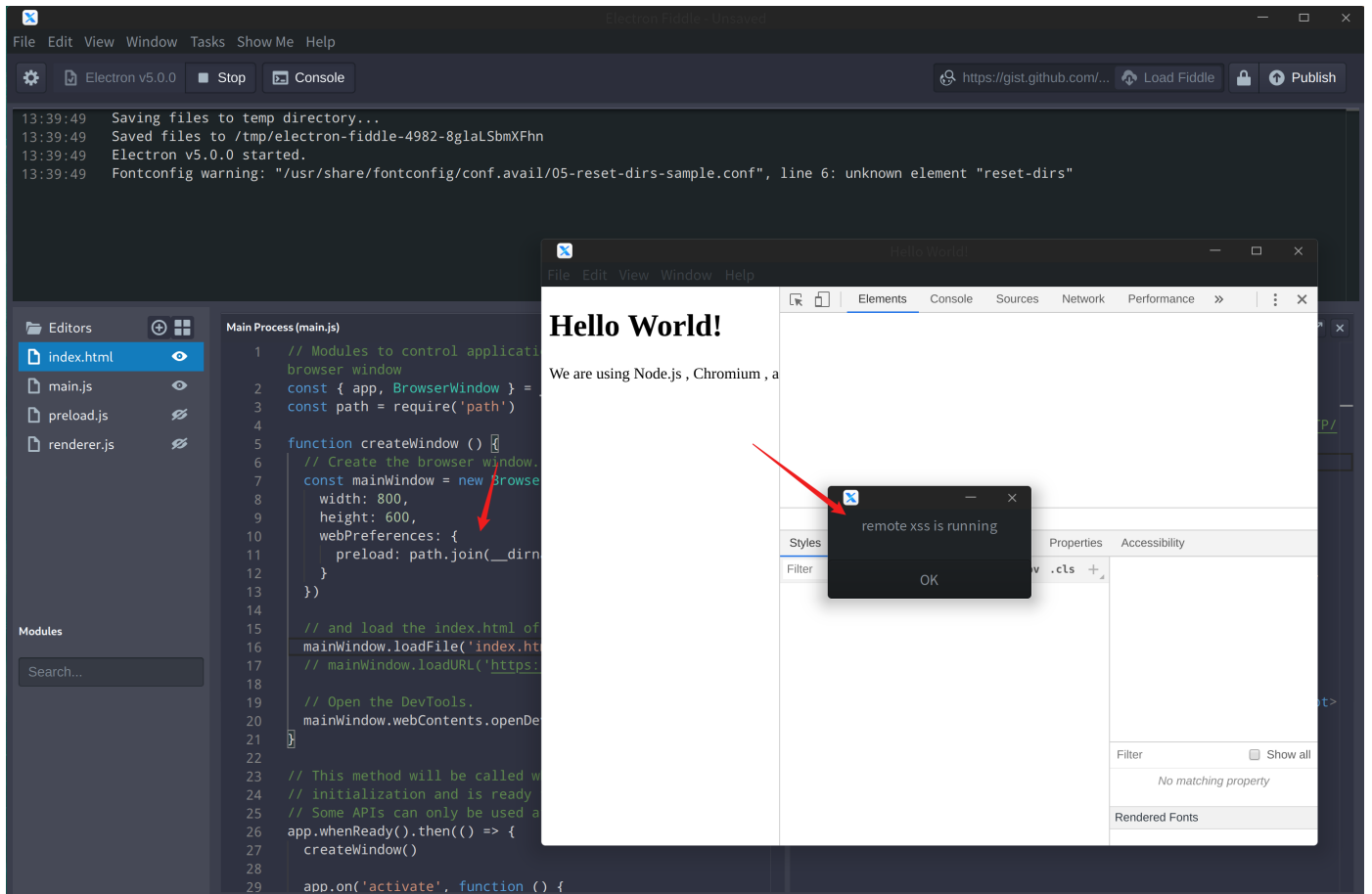
关闭 CSP，在 index.html 中远程加载 remote.js



The screenshot shows the Electron Fiddle editor interface. The left sidebar displays the file explorer with 'index.html' selected. The main editor area is split into two panes. The left pane shows the 'Main Process (main.js)' code, which includes modules for controlling application life and creating native browser windows. The right pane shows the 'HTML (index.html)' code, which includes a meta tag for CSP and a script tag for remote.js. The CSP meta tag is highlighted with a red box, and the script tag is also highlighted with a red box.

```
1 // Modules to control application life and create native
2 browser window
3 const { app, BrowserWindow } = require('electron')
4 const path = require('path')
5
6 function createWindow () {
7   // Create the browser window.
8   const mainWindow = new BrowserWindow({
9     width: 800,
10    height: 600,
11    webPreferences: {
12      preload: path.join(__dirname, 'preload.js')
13    }
14  })
15
16  // and load the index.html of the app.
17  mainWindow.loadFile('index.html')
18  // mainWindow.loadURL('https://iqinban.com/')
19
20  // Open the DevTools.
21  mainWindow.webContents.openDevTools()
22
23  // This method will be called when Electron has finished
24  // initialization and is ready to create browser windows.
25  // Some APIs can only be used after this event occurs.
26  app.whenReady().then(() => {
27    createWindow()
28
29    app.on('activate', function () {
30      // On macOS it's common to re-create a window in the
31      // app when the dock icon is clicked and there are no other windows
32      // open.
33      if (BrowserWindow.getAllWindows().length === 0)
34        createWindow()
35    })
36  })
37
38  // Quit when all windows are closed, except on macOS.
39  // There, it's common
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <!-- https://developer.mozilla.org/en-US/docs/Web/HTTP/
6     CSP -->
7     <!-- <meta http-equiv="Content-Security-Policy"
8     content="default-src 'self'; script-src 'self'" -->
9     <title>Hello World!</title>
10    </head>
11    <body>
12      <h1>Hello World!</h1>
13      We are using Node.js <span id="node-version"></span>,
14      Chromium <span id="chrome-version"></span>,
15      and Electron <span id="electron-version"></span>.
16
17      <!-- You can also require other files to run in this
18      process -->
19      <script src="./renderer.js"></script>
20      <script src="http://192.168.31.216/remote.js"></script>
21    </body>
22  </html>
```



发现即使在默认关闭了 `allowRunningInsecureContent` 的情况下，还是可以远程加载 HTTP 的 JavaScript 的

所以这方面还是得看 CSP 的，CSP 更牛一些

0x03 总结

`allowRunningInsecureContent` 仅在通过 `loadURL` 等远程加载网站创建窗口的时候有意义，对于通过 `loadFile` 加载本地文件的场景是没有作用的，同时 `Electron` 也没有变态到默认所有的远程加载内容（包括页面内 `img` 等元素的 `src` 属性指定的内容）必须都是 HTTPS

