



生之旅CVE-2018-0802

76959人围观，发现 3 个不明物体

漏洞

属 FreeBuf 原创奖励计划，未经许可禁止转载。

在潜伏近一年的“噩梦公式”漏洞(CVE-2017-11882)被曝光修补之后，之前的漏洞程序EQNEDT32.EXE在windows 10下仍然没有开启ASLR保护，因此其利用非常容易，在修补之后可以发现微软的发布的是二进制补丁而不是对该程序源代码进行重新编译，因此猜测该程序的源码可能已经遗失，如果真实情况是这样，那么公式编辑器这个进程将成为他漏洞发现和利用的“圣地”，因为微软将很难从源代码级别去排查这个程序是否还有其他漏洞。此次“噩梦公式”的妹篇CVE-2018-0802就在这一背景下被业界所发现。

概述

虽然微软在修补CVE-2017-11882漏洞使已经强制对公式编辑器进程开启了随机基址(ASLR)，这使得漏洞的利用门槛高，但由于补丁程序中的偶然因素使得能通过一个比较简单的方式绕过ASLR机制从而利用该漏洞。

由于此次漏洞也是由于“Equation Native”流中出现的问题，因此我们先了解一下它的大致结构，整个“EquationNative”数据由头结构和后续数据组成。其头结构为：

```
struct EQNOLEFILEHDR {  
  
    WORD    cbHdr;           // EQNOLEFILEHDR长度，恒为0x1c  
  
    DWORD   version;        // 恒为0x20000  
  
    WORD    cf;              // 剪切板格式("MathType EF")  
  
    DWORD   cbObject;        // MTEF数据长度，不包括EQNOLEFILEHDR部分  
  
    DWORD   reserved1;       // 未公开  
  
    DWORD   reserved2;       // 未公开  
  
    DWORD   reserved3;       // 未公开
```

```

    DWORD    reserved4; // 未公开

};

```

而紧接着头结构内容的是MTEFData内容，MTEFData内容也由MTEF头和MTEF字节流数据组成，MTEF头内容：



S

// MTEF版本号, 一般为0x03

// 系统生成平台, 0x00为Mac生成, 0x01为Windows生成

// 软件生成平台, 0x00为MathType生成, 0x01为公式编辑器

version; // 产品主版本号

BYTE bProductSubVersion; // 产品副版本号

};

MTEF字节流数据包括一系列的记录，每一个记录以一个标签位开始，标签位的低4位描述该记录的类型，高四位描述该记录的属性，后续紧跟标签的内容数据，由于此次发生栈溢出的部分在于字体标签，因此对字体标签进行了解：

```

struct stuFontRecord {

    BYTE    bTag; // 字体文件的tag位0x08

    BYTE    bTypeFace; // 字体风格

    BYTE    bStyle; // 字体样式

    BYTE    bFontName[n] // 字体名称, 以NULL为结束符

};

```

经过以上的介绍我们对于公式数据流这部分内容有了一定的了解。由于已经知道了这次是由于拷贝字体文件内容时产生的栈溢出，那么我们可以以CVE-2017-11882漏洞的POC为基础进行手工测试，根据Font的结构我们可以发现可以bFontName字段做一些文章，因为是以NULL为结束符的，因此我们可以通过构造长度超越平时长度的字符来试一试是否会产生异常事件。



关注我们 分享每日精选文章

图1 CVE-2017-11882构造的漏洞触发现场



关注我们 分享每日精选文章

图2 构造的畸形数据



关注我们 分享每日精选文章

图3 捕捉到异常事件

打开构造的畸形文件后捕捉到异常事件，发现该异常事件发生在EQNEDT32.EXE文件中，但异常事件并没有像普通栈溢出漏洞那样给我们泄露一些关键性信息，至今我们也只能知晓这构造的异常数据确实引发了EQNEDT32.EXE的c0000005异常，那么我们可以使用调试器进行调试看看到底内部是什么情况。

由已经获知的信息我们基本可以定位到可疑的大致代码位置，那么我们启动EQNEDT32.EXE程序并用OD附加该进程，然后去可疑的函数位置下个断点然后F9跑起来，之后打开我们构造的文件发现OD断在了该可疑函数中，我们逐步去单步走走，发现第三个call的参数很可疑，该函数第一个参数指向的内存内容和我们构造的畸形数据相同，且第二个参数也指向了附近的栈地址，因此可以使用IDA查看该函数内部情况。





关注我们 分享每日精选文章

图4 可疑调用函数



关注我们 分享每日精选文章

图5 栈溢出的函数

这是不经长度校验的拷贝，而传入的lpLogfont就是我们构造的数据，证明在此处我们可以制造一个栈溢出，观察此的拷贝起始地址位于上一个调用函数的栈帧中，因此利用本函数实现覆盖返回地址执行代码的做法是不行了，那么们现在可以控制覆盖上一个栈帧的函数返回地址，观察栈的结构可以发现拷贝起始地址据上一个函数栈帧返回地址差0×94(0x44EC24-0x44EB74-0x1C)，在覆盖了0×94字节后我们便可以覆盖上一个函数的返回地址。





关注我们 分享每日精选文章

图6 函数调用栈

但是要实现这一点得保证我们对于栈中数据的修改不会引起上一个调用函数中发生其他函数调用时出现异常，当然我们可以抱着侥幸心态去试试到底能不能行得通，毕竟我们更想看到这个栈溢出能不能被用起来，那么在回到上一层数后我们大胆的使用F8单步步过，每当能正常的通过一个函数时我们便朝着终点更进一步，如果能够顺利的走完整个函数那么证明我们就能劫持到执行流，使用IDA查看我们要到达终点之前需要正常越过的函数，然后就抱着赌徒心来使用F8越过这些可能会导致异常的函数。



关注我们 分享每日精选文章

图7 调用函数概览

当然事实证明任何抱有侥幸心理的行为都是作死，在使用F8进行赌一把的情形下，越过图8所示位置的函数时发生崩溃退出，重新附加公式编辑器再来到崩溃现场发现此处原来就是图7处调用CVE-2017-11882修补函数前的”_strcmpi(lpLogfont, &Name)”，原因是lpLogfont指针被覆盖了，那么我们重新调整我们构造的畸形文件，之前构造时数据太大导致覆盖了返回地址后接着又覆盖了上一个栈帧的数据，因此我们精确构造0×94大小的数据使返回地址好被覆盖且不破坏上一层栈中数据，之后重新附加调试重复我们的”赌博”。





关注我们 分享每日精选文章

图8 越过该函数时发生崩溃



关注我们 分享每日精选文章

图9 重新调整构造数据后成功越过该函数

调整我们构造的数据大小之后重新附加调试，再次到这个位置时我们成功的越过了这个函数，证明我们离终点更进步，然后在一路F8越过，突然我们命中了一个断点，查看断点的位置就是这个函数自身，在这里函数进行了递归调用，这对于我们来说并不算一个好事，这导致了我们的可能离最终的ret又增加了许多障碍，但我们在此时别无他法，能强行F8再一路往终点走。





关注我们 分享每日精选文章

图10 命中断点的函数位置

庆幸的是这个递归调用用一个远跳很快的就跳出了函数主体，甚至于连栈溢出的函数都没有机会执行，那么证明我运气还是非常好的，在这个函数ret之后需要我们越过的函数调用就没有多少了，希望就在眼前。



关注我们 分享每日精选文章

图11 递归调用成功F8步过

退出递归调用函数后，继续靠着我们一手绝活F8最终我们来到了我们想要的终点，发现返回地址被我们构造的数据成功覆盖，证明该漏洞可以被利用。





关注我们 分享每日精选文章

图12 最终的ret成功劫持执行流

那么现在我们知道了我们可以构造一个长度为0x94字节长度的内容来实现一个shellcode，然后覆盖返回地址就能成功劫持执行流，通过调试发现在覆盖的返回地址之下，也就是函数调用的第一个参数刚好指向我们构造的源数据，那我们在返回地址只需要一个ret指令，便能成功引导执行流从我们的shellcode开始执行，但是在CVE-2017-11882漏洞曝光之后微软对公式编辑器进程强制开启了动态随机基址，意味着我们不能写死返回地址，因为该地址每次都在变化，那么总有大神有奇招(<http://www.freebuf.com/column/160006.html>)，参考该篇文章介绍的绕过ALSR的手法我们可以成功的bypass ALSR。原理大致就是随机基址通常只随机高两位地址，而在内存中由于小端排序的原因，我们覆盖是从低两位开始的，那么由于末尾必须用NULL截断，因此倒数第二位地址必须为00，查找是否具有0xXXYY00ZZ(X,Y,Z为随机字符)格式的ret指令，最终发现事实就是这么偶然，刚好该函数内部有这么一个地址。





关注我们 分享每日精选文章

图13 用于bypass ALSR的retn指令

我们的shellcode结构已经比较清晰了，前面0×94字节用于写入我们的shellcode，紧接着我们用“\x25\x00”覆盖返回地址两位，那么就开始布局我们的shellcode，由于shellcode空间只有0×94字节的空间十分有限，因此我们可以考虑模仿CVE-2017-11882中使用的手法利用进程中的WinExec函数来完成调用其他程序的方法。





关注我们 分享每日精选文章

图14 WinExec函数仍然存在

最终写出汇编代码提取出shellcode放入文件中用于构造字体文件名称的位置，也就是之前我们构造的”aaa...”处，并覆盖起始地址偏移0×94字节后填入关键的覆盖返回地址的信息”\x25\x00”，那么在文件中的shellcode就这么构造成功了，我们把需要执行的命令行参数紧接着写到shellcode之后便可以实现一个利用CVE-2018-0802漏洞执行的rtf文件，然这需要主机打上CVE-2017-11882的补丁之后才能成功复现。





关注我们 分享每日精选文章

图15 构造的shellcode

用这个shellcode从我们开始构造”aaa...”的起始地址开始覆盖，在之后紧接我们需要执行命令(先用打开计算器试个水)，然后用不为NULL的字符进行填充至一共0×94字节，再用我们的bypass ALSR的”\x25\x00”来覆盖返回地址低两位，那么一个简单的漏洞利用文档就产生了。



关注我们 分享每日精选文章

图16 利用CVE-2017-11882改造成的CVE-2018-0802漏洞利用文档



关注我们 分享每日精选文章

图17 打开文档后自动弹出计算器

至此利用CVE-2017-11882漏洞文件进行一个简单的改造，我们完成了CVE-2018-0802文件的构造，打开该文档后自动弹出计算器，但是这并不是我们最终的利用形态，接下来我将展示攻击者如何利用该漏洞进行恶意行为，这里其实利用了rtf文档的一个特性，在打开rtf文档时会将嵌入文档中的文件自动释放到系统%temp%目录下，因此现今最多测试攻击的方式都是利用这个特性在rtf文档中嵌入恶意EXE文件，然后再利用该漏洞去执行释放在%temp%目录下的文件。首先修改我们构造的用于启动计算器的执行命令，更改为“cmd /c %temp%\hello.exe”之类的命令。



关注我们 分享每日精选文章

图18 修改为执行其他EXE程序的命令形态

接下来我们新建一个任意类型的文件，改后缀名为.rtf，用word打开后拖入我们想要执行的exe文件，这个exe文件应要和修改的命令中文件名对应，然后保存用二进制编辑工具打开，将刚才我们构造那部分恶意执行代码的整个obje容拖放到存放了exe文件的rtf中指定的位置。



关注我们 分享每日精选文章

图19 嵌入exe文件并保存



关注我们 分享每日精选文章

图20 需要拷贝到嵌入exe的rtf文档中的恶意公式编辑器内容



关注我们 分享每日精选文章

图21 拷贝到嵌入exe文件的rtf文档中的大致位置

至此真正一个具有攻击性的rtf文档就形成了，该文档在被word程序打开后将会由于自身机制释放hello.exe文件到%temp%目录下，之后由于公式编辑器的漏洞利用了WinExec函数执行了“cmd /c %temp%\hello.exe”，实现了最终攻击行为，当然我在这里用的示例程序只是一个提示中毒的标签，并没有真正的攻击行为。





关注我们 分享每日精选文章

图22 打开文档后执行我们嵌入rtf文档中的任意exe文件

之前利用CVE-2017-11882的漏洞也可以这样非常简单的构造一个恶意文档出来，由于这个漏洞的利用手法非常简单只需要一个漏洞文件和一个二进制编辑器即可实现利用并发动攻击，因此其破坏性不容忽视，建议广大用户赶紧打最新的补丁，以防止此类攻击，最新的补丁彻底抛弃EQNEDT32.EXE文件，从而杜绝了攻击者想利用该漏洞”圣地”进行漏洞利用的行为。

参考文章

<http://www.freebuf.com/column/160006.html>

<https://research.checkpoint.com/another-office-equation-rce-vulnerability/>

POC地址

<https://github.com/GeekOnlineCode/POC>

*本文作者：GeekOnline，本文属 FreeBuf 原创奖励计划，未经许可禁止转载。

已

陌离



[二代的原理分析、利用与防护方案](#)

关注我们 分享每日精选文章
别说话，看图。

亮了

亮了

亮了

fangdangbuji (1级) 2018-01-22

@ 陌离 哈哈，开始以为他要用cve-2017-0802引出2018-0802呢。。。

GeekOnline (3级) 公众号:"GeekOnline" 2018-01-22

@ 陌离 谢谢您的指正，正在联系修正，给大家带来的困惑和误解表示深深的歉意

选择文件 未选择任何文件

昵称

请输入昵称

邮箱

请输入邮箱地址

表情

插图

必须 您当前尚未登录。[登陆?](#) [注册](#)

必须 (保密)

提交评论(Ctrl+Enter)

取消

☒ 有人回复时邮件通知我




GeekOnline

公众号:"GeekOnline"

6

评论数

最

- 

E-2018-0802

2018.01.22
- 关注我们 分享每日精选文章

• [穿透内网防线|USB自动渗透手法总结](#)

2018.01.07
- [二进制之病毒在局域网自动传播方式小结](#)

2017.12.16

浏览更多

相关阅读

- [漏洞预警： MongoDB phpMoAdmin曝...](#)
- [挖洞经验 | 看我如何利用上传漏洞在Pa...](#)
- [Hacking Team安卓浏览器攻击过程中的...](#)
- [CVE-2012-1823 php-cgi远程代码执行](#)
- [2015年数据库漏洞威胁报告](#)

特别推荐



不要错过
关注我们 分享每日精选文章



[OpenVAS开源风险评估系统部署方案](#)

魅影儿

2017-04-30

[2015最酷的Hack方式有哪些？](#)

简单

2016-01-05

[双刃剑与灰色地带：“泄露数据收藏家”的素描](#)

孙毛毛

2016-09-27

[【限时优惠】FreeBuf精品公开课 | 36W漏洞奖金先生CplusHua：](#)

FB客服

2017-09-16



Copyright © 2018 WWW.FREEBUF.COM All Rights Reserved [沪ICP备13033796号](#)

阿里云 提供计算与安全服务