CVE-2018-15421溢出漏洞以及使用Hackvertor绕过WAF

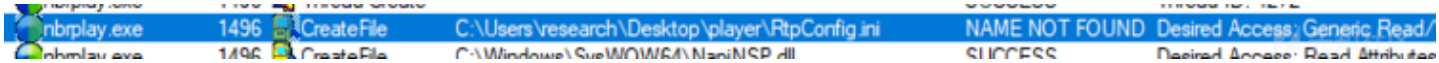
风吹花 / 2018-11-08 07:01:00 / 浏览数 4246 技术文章 技术文章 顶(0) 踩(0)

前言

在过去的一年中，几位研究人员在Cisco
Webex程序套件中提交了漏洞报告。然而，在他们提交的40多个报告中，他们忽视了某些琐碎的东西，包括一些基于堆栈的缓存溢出。这篇文章(https://www.zerodayiniti
)详细地介绍了ZDI-18-1078，这是Cisco Webex网络录制播放器31.23.2.58(现已达到EOL)版本中的一个漏洞，它会使得远程代码被执行。

漏洞

读取Advanced
Recording■.arf■文件时，播放器会尝试访问当前目录中的RtpConfig.ini文件，这个操作还没有被记录下来。这个.ini文件包含一种配置，该配置可能是一种实时传



上图展示了nbrplay.exe正在查找RtpConfig.ini的过程

这个错误发生在nmvc.dll中，它存在于名为sub_1001F479的程序中，这个程序可以用来解析RtpConfig.ini，并且提取它的属性。以下代码段展示了如何设置MinLos

```
1001F479
1001F479
1001F479 ; Attributes: bp-based frame
1001F479
1001F479 ; int __stdcall sub_1001F479(char *Filename)
1001F479 sub_1001F479 proc near
1001F479
1001F479 var_920= byte ptr -920h
1001F479 var_820= dword ptr -820h
1001F479 var_81C= dword ptr -81Ch
1001F479 var_818= dword ptr -818h
1001F479 var_814= dword ptr -814h
1001F479 var_810= byte ptr -810h
1001F479 Dest= byte ptr -710h
1001F479 Buf= byte ptr -610h
1001F479 Dst= byte ptr -60Fh
1001F479 Source= byte ptr -210h
1001F479 Str1= byte ptr -110h
1001F479 var_4= dword ptr -4
1001F479 Filename= dword ptr  8
1001F479 arg_4= dword ptr  0Ch
1001F479
1001F479 ; FUNCTION CHUNK AT 1003C999 SIZE 00000033 BYTES
1001F479
1001F479 ; __unwind { // loc_1003C9A4
1001F479 push    914h
1001F47E mov     eax, offset loc_1003C9A4
1001F483 call    __EH_prolog3_GS
1001F488 mov     esi, ecx
1001F48A mov     [ebp+var_814], esi
1001F490 mov     ebx, [ebp+Filename]
1001F493 and     dword ptr [esi], 0
1001F496 and     dword ptr [esi+4], 0
1001F49A mov     [ebp+var_820], esi
1001F4A0 call    sub_1001FAED
1001F4A5 mov     [esi], eax
1001F4A7 ;   try {
1001F4A7 and     [ebp+var_4], 0
1001F4AB lea     eax, [ebp+Dest]
1001F4B1 push    offset Source   ; "0.005"
1001F4B6 push    eax             ; Dest
1001F4B7 call    strcpy
1001F4BC lea     eax, [ebp+Str1]
1001F4C2 push    offset aMinlostrate ; "MinLostRate"
1001F4C7 push    eax             ; Dest
1001F4C8 call    strcpy
```


这里的罪魁祸首是对sscanf函数的调用，这是一个被微软禁用的函数。该sscanf函数解析.ini文件并读取属性值，以便将它们与一组硬编码参数进行匹配。使用的格式是：

```
%[^ \t#]%*[ \t]%[^ \t#]%n
```

这个格式可以写入三个参数。第一个和第三个说明符(%[^
\t#])在%和[之间不使用任何值,这意味着在它到空格之前，它将读取每个字符。忽略它们的大小的话，这将可以写入Str1和Source参数，如果输入足够大，可能会导致溢出

```
1001F81C lea     eax, [ebp+var_818]
1001F822 push    eax
1001F823 lea     eax, [ebp+Source]
1001F829 push    eax
1001F82A lea     eax, [ebp+Str1]
1001F830 push    eax
1001F831 mov     eax, [ebp+var_81C]
1001F837 push    offset aN_0     ; "%[^ \t#]%*[ \t]%[^ \t#]%n"
1001F83C lea     eax, [ebp+eax+Buf]
1001F843 push    eax             ; Src
1001F844 call    ds:sscanf
```

这个.ini文件在0x3FF字节块中被读取，并且因为这两个连续变量Source和Str1的空间分别为0x100和0x106字节，因此可发生溢出从而导致堆栈被损坏。

```
1001F7DA push    ebx             ; File
1001F7DB lea     eax, [ebp+Buf]
1001F7E1 push    400h            ; MaxCount
1001F7E6 push    eax             ; Buf
1001F7E7 call    ds:fgets
```

```
0:000> kv
 # ChildEBP RetAddr  Args to Child
WARNING: Stack unwind information not available. Following frames may be wrong.
00 00efb740 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a nmvc!ITransportSink::operator=+0x5146
01 00efb744 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
02 00efb748 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
03 00efb74c 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
04 00efb750 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
05 00efb754 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
06 00efb758 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
07 00efb75c 5a5a5a5a 5a5a5a5a 5a5a5a5a 5a5a5a5a 0x5a5a5a5a
```

更多

思科公司通过咨询cisco-sa-20180919-webex修补了这几个漏洞。让人高兴的是，随着越来越多相类似的错误被提交，这些版本即将达到他们的EOL，这些新版本有希望

你可以在Twitter@ziadrb上找到我，并跟随我的团队寻找最新的漏洞利用技术和安全补丁。

用Hackvertor绕过WAFS和破解XOR

我最近一直在致力于研究如何扩展Hackvertor，因为它具有基于标签的转换功能，这比Burp中的内置解码器要厉害得多。在这样一个功能的背后，我的想法是，标签可以转

例如，要将一个字符串编码为base64，只需使用base64进行标记：

```
<@base64_0>test<@/base64_0>
```

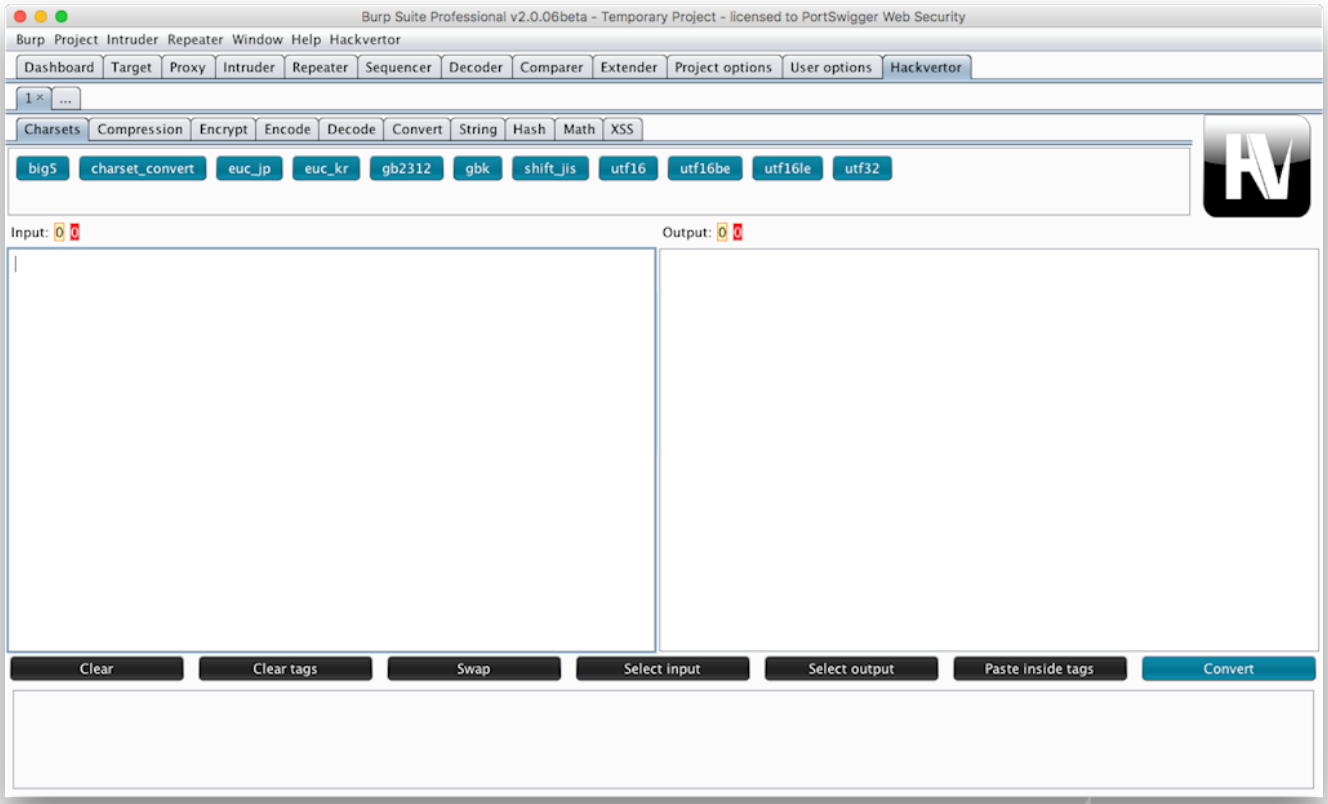
您也可以进行多级编码，例如假设您想将字符串转换为十六进制，然后对其进行base64编码，您首先要使用十六进制标记，然后使用base64标记：

```
<@base64_1><@hex_0(" ")>test<@/hex_0><@/base64_1>
```

这个十六进制标记有一个分隔符参数，用于分隔每个十六进制字符串，即'test'，它将作为分隔符连同空格一起传递给标记函数。

使用Hackvertor

当扩展部分在加载时，它将在Burp中创建一个名为Hackvertor的新Tab。在选项卡中有一个输入框和一个输出框，您在输入框中输入要转换的文本，将它选中，然后单击，



绕过Cloudflare WAF

最近在诸如repeater等工具中，我制作了Hackvertor标签，您只需右键单击repeat请求，并点击Hackvertor菜单然后在请求中添加标签，它将会在发送请求之前自动运行转

现在，我将向您展示如何在repeater模块中使用标签从而绕过Cloudflare WAF。首先要将以下网址发送给repeater：

```
https■//waf.party/xss/xss.php■x =
```

然后在等号之后输入以下代码：

```
<img/src/onerror=alert(1)>
```

在repeater请求中选择alert(1)，然后启用Hackvertor
BApp，右键单击所选文本，接着单击Hackvertor，然后单击XSS，最后单击throw_eval。这会将标记添加到请求中，如果你点击go，你会看到这样的回复：

```
<img/src/onerror=window.onerror=eval;throw'=alert\x281\x29'>
```

如果你想检查它是否实际地在工作，你只需要右键单击请求编辑器并从菜单和Copy
URL中选择Hackvertor，然后在转换所有标签后生成一个URL。如果您使用Burp's copy URL命令，那么这将仅复制带有标记的URL。

解码rotN

这一切都源于我的女儿。我从2016年开始穿着一件带有'Sides
Manchester'标记的T恤，前面有一些二进制字符，她问"爸爸这些数字是什么意思？"。我告诉她这是二进制代码，并询问她是否要将它解码。然后我们开始在Hackvertor中

它必须从一堆类似于胡言乱语的话当中识别出像英语那样的单词，因此我开始创建一个is_like_english标签，起初我以为可以使用二元语法和三元语法模型，并且只需要在它

下一步是改进自动解码器。自动解码器是一种标签，它可以自动尝试确定字符串的编码方式，并且对其进行多次解码。我添加了一个简单的正则表达式，用于查找更多的a-z

```
<@auto_decode_0>01010111 01101101 00110101 01101000 01100011 01001000 01010110 01111001 01011010 01101101 01100100 01111001 01
```

James还有一个带有扬声器的衬衫，上面有一些不同的代码，所以我进入了Hackvertor，看看它是否会将它们自动解码。事实证明它是很有效的，你自己可以尝试着将以下内

```
<@auto_decode_10>01011010 01101110 01100001 01110000 01110101 01110010 01100110 01100111 01110010 01100101 00101100 00100000 0
```

毋庸置疑，如果自动rotN可以破解任何代码，这将会变得非常有趣。

重复密钥以解密XOR

我本来打算结束我的文章，但是James向我发起挑战，要我解码重复加密的XOR。我使用了加密站点Practical
cryptography，并学习了所有关于XOR和频率分析的知识。第一步是确定密钥长度，您可以为每个关键候选项使用频率分析来执行此操作，我将30作为最大密钥长度。我将

我花了很多时间来尝试提高关键猜测的准确性，并且多次重写了代码。Trusted signal
blog博客证实，你可以通过使用前5-6个候选结果当中的最大公分母来提高确定的键长的准确性，但是在我的测试中，我无法提高准确性。无论如何，一旦你有了密钥长度，
Hellman 的cool python utility中。

最后，无论转换成功与否，我将重新使用我的is_like_english函数来确定文本的分数。这适用于小块文本，更大的文本就不行了，这是因为你输入越多的文本，你获得的ngra

为了演示自动解码，我用一个密钥进行了XOR，然后对其进行了十六进制编码。当您在输入框中输入时，Hackvertor将自动解码十六进制，预估密钥长度，然后自动解密XO

```
<@auto_decode_8>1C090C1E05041C101C523D296324212F000D020C04061D001C216F3638366823161906452101060637 6F3724732E080D0F561617171A00
```

重复的XOR时不时会被使用，所以希望这个功能能够防止某些应用程序逃脱看似合法的加密。

■■■■■

```
https://www.zerodayinitiative.com/blog/2018/9/27/cve-2018-15421-examining-a-stack-based-overflow-in-the-cisco-webex-network-re
```

```
https://portswigger.net/blog/bypassing-wafs-and-cracking-xor-with-hackvertor
```

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