

宏病毒处理思路
破坏宏标志
宏清除脚本
手工清理宏（针对*.DOCX和*.XLSM文档）
替换宏代码
说明

在前一章我们解析了宏病毒的二进制格式，本篇紧跟上文，利用宏病毒的二进制格式清除宏病毒。

宏病毒处理思路

破坏宏标志

在解析OLE文件时，我们介绍过Directory，其中其偏移0x42H这个字节的表示DirectoryEntry的类型Type。0为非法，1为目录（storage），2为节点（Stream），5为根节点。

我们可以定位宏工程所使用的Directory，将Type位修改为0，即非法Directory，这样Office办公软件在解析VBA工程时，会因为非法Directory而导致解析错误。

如图所示，表示该扇区类型的字节记录在偏移0x42处，该处值为1，表示该扇区是目录扇区，该扇区记录着Macros文件的地址。将偏移0x42处值修改为0，表示该扇区是非法扇区，office办公软件解析文档是就会忽略该扇区，也就无法找到存储Macros的扇区了。

0001C700	4D 00 61 00 63 00 72 00 6F 00 73 00 00 00 00 00	M.a.c.r.o.s.....
0001C710	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001C720	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001C730	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001C740	0E 00 01 01 02 00 00 00 11 00 00 00 0F 00 00 00
0001C750	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001C760	00 00 00 00 30 68 D7 85 3C 36 D2 01 30 68 D7 850h*.<60.0h*.
0001C770	3C 36 D2 01 00 00 00 00 00 00 00 00 00 00 00 00	<60.....

在上图中，我们值修改了名为“宏”的这个目录中。实际上，用于宏的指南还有两个，分别名为“VBA”，和“_VAB_PROJECT_CUR”，分别是宏，VBA和的VBAProject扇区，

宏清除脚本

根据如上原理，笔者写了一个简单的python处理脚本，源码如下：

```
import sys
import os.path

#read bytes
def readfile (filename, address, size) :
    f = open (filename, 'rb')
    f.seek (地址, 0)
    cont = f.read (size)
    f.close ()
    返回cont
```

```

def writefile (文件名, 地址, 大小) :
f = open (filename, 'wb')
f.seek (address, 0)
cont = f .write (0x0)
f.close ()
返回cont

#classify: doc, docm或其他?
def classify (filename) :
filehead = readfile (filename, 0,8)
if (filehead [0] == chr (0xD0) and filehead [1] == chr (0xCF) ) :
#print“这个文件是doc文件”
return 1
elif (filehead [0] == chr (0x50) 和filehead [1] == chr (0x4B) ) :
#print“此文件是docm / docx文件”
return 2
else:
#print“此文件不是word文件”
return 0

#repairfile
def repairedoc (filename) :
f = open (filename, 'rb')
ff = open (newfile, 'wb')
cont = f.read ()
dirsect = readfile (filename, 0x30,4)
diraddr = (0x1000000 * ord (dirsect [3]) + 0x10000 * ord (dirsect [2]) + 0x100 * ord (dirsect [1]) + ord (dirsect [0]) + 1) * 0x200
flag = 0
for i in range (0,16) :
dirname = readfile (filename, diraddr + 0x80 * i, 0x10)
if dirname == CHR (送出0x4d) + CHR (0x00) + CHR (0x61) + CHR (0x00) + CHR (0x63) +
CHR (0x00) + CHR (0x72) + CHR (0x00) + CHR (0x6f) + CHR (0x00) + chr (0x73) + chr (0x00) +
chr (0x00) + chr (0x00) + chr (0x00) + chr (0x00) :
#print“宏”
标志= 1
typeaddr = diraddr + i * 0x80 + 0x42
cont = cont [: typeaddr] + chr (0x00) + cont [typeaddr + 1: ]
elif dirname == chr (0x56) + chr (0x00) + chr (0x42) + chr ( 0x00) + CHR (0x41) + CHR (0x00) +
CHR (0x00) + CHR (0x00) + CHR (0x00) + CHR (0x00) + CHR (0x00) + CHR (0x00) + CHR (0x00) + CHR (
0x00) + chr (0x00) + chr (0x00) :
#print“VBA”
flag = 1
typeaddr = diraddr + i * 0x80 + 0x42
cont = cont [: typeaddr] + chr (0x00) + cont [typeaddr + 1: ]
#cont [typeaddr] = chr (0x00)
elif dirname == chr (0x5f) + chr (0x00) + chr (0x56) + chr (0x00) + chr (0x42) + chr (0x00) +
chr (0x41) + chr (0x00) + chr (0x5f) + chr (0x00) + chr (0x50) + chr (0x00) + chr (0x52) +
chr (0x00) + chr (0x4f) + chr (0x00) :
#print“_VAB_PROJECT_CUR”, 只需检查“_VAB_PRO”
flag = 1
typeaddr = diraddr + i * 0x80 + 0x42
cont = cont [: typeaddr] + chr (0x00) + cont [typeaddr + 1: ]

```

```

#cont [typeaddr] = chr (0x00)
ff.write (cont)
f.close ()
ff.close ()
if flag == 0:
print“我没找到宏!”
#print hex (diraddr)

def main (filename) :
filetype = classify (filename)
if (filetype == 1) : #doc file
repairdoc (filename) print'DOC
repair done! '
elif (filetype == 2) : #docm / x file
print“这个文件是高版本的olefile, 请自行修复!”
else:
print“这个文件不是olefile。”
if len (sys.argv) != 2:
print“USAGE: deletemacro.py filename”
print“eg: deletemacro.py D: \ 1.doc”
else:
filename = sys.argv [1]
newfile = 'new _'+ os.path中。

```

。上述脚本只能处理的*.doc, *.xls的文档, 处理说明:

.DOCX和*.xlsx的文件不含宏, 所以不需要处理。处理之后打开文件, 有时会触发“宏告警”或“文件损坏告警”, 但不会影响正常使用, 文件中的宏不会运行, 文字内容也可以正常查看。

分析运行结果:

·如果返回“干净完成!”, 说明宏清理工作已经完成, 新生成的文件名为new_filename, 本例中为new_1234.doc, 该文件中的宏已经被清理。

```

C:\Users\14215\Desktop\1>deletemacro_v03.py 1234.doc
Clean done!

```

如果返回“这个文件是高版本的olefile, 请自行修复!”, 说明该文档是*.docm或*.xlsm文档, 需要手工清理宏

```

C:\Users\14215\Desktop\1>deletemacro_v03.py 123456.doc
This file is a high version olefile, please repair it by yourself!

```

·如果返回“此文件不是olefile”, 说明该文件不是*.doc, *.docm, *.xls, *.xlsm文件。



```

C:\Users\14215\Desktop\1>deletemacro_v03.py C:\Windows\notepad.exe
This file isn't an olefile.

```

手工清理宏 (针对*.DOCM和*.XLSM文档)

·首先拷贝文件, 生成文件副本, 将文件后缀名修改为ZIP

 123456 - 副本.zip	2016/11/12 14:24	WinRAR ZIP 压缩...	31 KB
 123456.doc	2016/11/12 14:24	Microsoft Word ...	31 KB

打开拉链文件（不要解压），下面两个图中，上图是*.DOCM的情况，下图是*.XLSM的情况

名称	大小	压缩后大小	类型	修改时间	CRC32
..			Folder		
_rels			Folder		
docProps			Folder		
word			Folder		
[Content_Type...	1,563	419	XML Document	1980/1/1 0:00	4DC12E...

名称	大小	压缩后大小	类型	修改时间	CRC32
..			Folder		
_rels			Folder		
docProps			Folder		
xl			Folder		
[Content_Type...	1,218	388	XML Document	1980/1/1 0:00	0E351B74

打开word (xl) 文件夹，删除vbaProject.bin文件

..			Folder		
_rels			Folder		
theme			Folder		
document.xml	1,517	564	XML Document	1980/1/1 0:00	B18A10E3
fontTable.xml	1,186	454	XML Document	1980/1/1 0:00	DE9395...
settings.xml	5,215	1,542	XML Document	1980/1/1 0:00	F50878FA
styles.xml	14,921	1,838	XML Document	1980/1/1 0:00	A727715F
stylesWithEffec...	15,674	1,967	XML Document	1980/1/1 0:00	0A365A39
vbaData.xml	1,298	467	XML Document	1980/1/1 0:00	5E528AC0
vbaProject.bin	35,840	16,867	XML Document	1980/1/1 0:00	30F22C...
webSettings.xml	428	251	XML Document	1980/1/1 0:00	A017

全部选定(A)

Ctrl+A

添加文件到压缩文件中(O)

Alt+A

解压到指定文件夹(E)

Alt+E

测试压缩的文件(T)

Alt+T

查看文件(V)

Alt+V

删除文件(D)

Alt+D

修复压缩文件(R)

Alt+R

无须确认直接解压缩(W)

Alt+W

显示信息(I)

Alt+I

·将ZIP文件后缀名修改为.DOC或.xls的。文件可以正常使用。

实际上*.DOCM和*.XLSM文件使用的标准的压缩算法，稍微改造上述脚本，就可以利用脚本实现对*.DOCM，*.XLSM的处理。

替换宏代码

上述方式修改标志位，对文档中的宏代码数据没有做丝毫改变，这样处理之后杀毒软件仍然会查杀。

我们已经可以解析宏数据，那么我们就可以将宏代码进行修改和替换，如使用字符'a'替换宏代码中的每一个字符。例如MsgBox"hello"，就会被修改为aaaaaaaaaaaa。这样修改之后整个宏都已经被修改和破坏掉，杀软将不会查杀，文档也可以正常使用。

说明

- 本文并非原创，乃是征得作者同意后的转载 原作者为狐狸先生 未经允许,禁止转载
- 需要相关文件可以到我的Github下载:<https://github.com/TonyChen56/Virus-Analysis>
- 应作者要求 贴上知识星球图片 主要分享病毒分析和逆向破解技术，文章质量很高 我也在这个星球里 大家可以积极加入



星主：crazyman

逆向/破解/病毒分析

 知识星球

长按扫码预览社群内容
和星主关系更进一步

