Testing Port Connectivity

Grow with Google: IT Support Certificate

Sections

---

netcat

The following commands are used on Linux and MacOS devices. The letters `nc` are used for the netcat command along with a host and a port. Net stands for networking. Cat comes from the Unix command line program cat, short for concatenate, which means to link things together in a chain or series.

`nc [options] <host> <port>`
`Example command: nc google.com 80`
This command tries to establish a transmission control protocol (TCP) connection to the host <google.com> on the specified port <port 80>.

The host can be a website or a targeted IP address.
The port can be a specific port or a range of port numbers.

Listed below are some of the options available for you to use in testing port connectivity.

> `nc -u <host> <port>`
> Tells netcat to open a user datagram protocol (UDP) connection, instead of a TCP connection.

Example: Certain network protocols use UDP for speed or efficiency purposes. To test them, you will open a UDP connection by using nc -u and identify the host and port the data needs to be sent to.

`nc -z <host> <port>`
Stands for zero input/output and tells it to scan for open ports.

Example: You can use nc -z to scan for unneeded services that could be listening into a network without sending any data to them. This can be combined with using a range of port numbers instead of just a single port.

`nc -v <host> <port>`
Stands for verbose and gives extended output text important for debugging and troubleshooting.

Example: You want to run a scan looking for listening devices and adds -v so that it will return lists of ports and statuses on the network or website being scanned.

`nc -vv <host> <port>`
Stands for very verbose and gives more output text than just verbose

Example: The IT administrator has asked for a very detailed report of all ports and their statuses within the company network. A netcat command can be run with -vv to get the robust information on all ports and their statuses.

`nc -p <localport> <host> <port>`
Refers to a local port for a connection. Some protocols require a specific source port to work properly, this lets you specify what port to connect from.

`nc -e <program> <host> <port>`
Executes a program after connection established. This option is not supported by all version of netcat, but you can also use standard unix command line pipelines to pass network input to or from other programs.

`nc -n <addr> <port>`
prevents domain name server (DNS) lookup. Use this when you have an IP address and numeric port to use for the connection and you want to avoid the overhead of DNS or if it is not working properly.

These command-line options can be used independently or combined with one another.

Example: An IT administrator wants to evaluate the network for open doors or weak connections that would allow someone to hack into the network. To discover this you run nc -v -z google.com 80 to determine if a connection to the port 80 is possible to google.com.

Test-NetConnection

The following commands are used on Windows PowerShell devices. The command, Test-NetConnection is case sensitive and uses capitals unlike netcat.

`Test-NetConnection <host> <port>`
`Example command: Test-NetConnection -ComputerName google.com -Port 80`
Tests ping connectivity and displays diagnostic information for a connection from the host google.com on port 80

`Test-NetConnection -InformationLevel "Detailed"`
Tests ping connectivity with detailed results.

Example: A data transfer on the network is moving very slowly. In order to check the quality of the connection an you runs Test-NetConnection -InformationLevel “Detailed” to view details about the connection. This will connect to a default address from microsoft.

`Test-NetConnection -ComputerName [remote host]`
Tests a connection to a remote host.

`Test-NetConnection -ComputerName [remote host] -Port [port number]`
Tests TCP connectivity to a specific host and port. This can be combined with the display detailed results option:
Example:

```
PS C:\> Test-NetConnection -ComputerName www.google.com
-Port 80 -InformationLevel Detailed
```

```
Test-NetConnection -ComputerName [remote host]
-DiagnoseRouting
```
Performs route diagnostics to connect to a remote host. This can require administrator privileges, so you may have to run your powershell window as administrator.

```
Test-NetConnection -ComputerName [remote host]
-constrainInterface [interface number] -DiagnoseRouting
-InformationLevel "Detailed"
```
Performs route diagnostics to connect to a remote host with routing constraints.

Example: An employee is having trouble connecting to a specific website from their computer, but other sites are loading fine in the browser. You can try connecting to the website directly with:

```
Test-NetConnection -ComputerName www.example.com -Port 80
-InformationLevel Detailed
```