Features of Netcat

Jeffrey Kurcz
School of Computer Science
University of Windsor
kurcz@uwindsor.ca

Abstract

This paper discusses the many uses that Netcat can perform for many different tasks that need to be done on a network. Netcat is network tool that allows a user or administrator write or read data over the network. This can be done by using both the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) protocols. This tool was rated #4 on sectools.org top 100 Network Security Tools by insecure.org. You will see why it is considered the Swiss-Army Knife of TCP/IP because of its high recognition and usefulness of the abilities it can perform.

Keywords

Netcat, Local Area Network (LAN), Wide Area Network (WAN), Remote Access, Port Scanning, Banner Grabbing, Server-Client Networking

I. Introduction

As newer programs, applications, and systems are released for the world, they are becoming larger in size, scale and complexity. This issue creates potential for many small bugs and glitches in programs because they are so large that a developer may miss key issues, or for testing purposes create a bug to be used then late for get to take it out when publishing. Since so many vulnerabilities are appearing there are many tools to help a hacker or penetration tester assess and validate the security of a vulnerable or invulnerable system.

Netcat is one of these tools. It offers such a diverse scale of features that it is capable of taking hold of a system from the beginning, starting with scanning and looking for open ports that services may be running on a system. With the next step of banner grabbing which is connecting to this system on an open port and getting a response from the server on which services are running along with which version and sometimes the Operating System in which it is running on.

Once the Hacker or Penetration Tester has this information they can find vulnerabilities or use a well known exploit to gain access to the system they want. Whether they exploit a vulnerable FTP server or a vulnerable Web server doesn't matter as long as they can get access to the machine. From there the hacker will use that backdoor to gain access whenever they want, or upload a Trojan or create another backdoor of their own for their own access so that another hacker cannot gain access from the same vulnerable program that they just used.

Netcat allows the use for all of these, from port scanning, to banner grabbing to creating a backdoor on a host machine for allowing shell access to the hacker to take over the computer. This is probably why Netcat is noted as such a useful tool to many hackers, and a useful tool for network administrators troubleshooting or debugging their networks.

II. History

Netcat is a simple UNIX utility with its last stable build in 1996 by the creator named Hobbit. There has been many other successors to Netcat such as Socat, Cyrptcat and ncat. The name Netcat comes from the UNIX tool *cat* which reads and writes files from an operating system, where Netcat reads and writes data on a network communication using the TCP or UDP protocols.

The purpose of Netcat is for network administrators to troubleshoot or debug issues pertaining to their network either locally or remotely, while it also has those uses Black hats can use it for mischievous tasks. Netcat can be run as a simple standalone command line program or as backend for another program or script.

III. Features

3.0 Netcat Syntax

Here is the general syntax for Netcat:

nc [options] hostname port[s]

To make Netcat run as a server you must include the following two flag options:

-l: this option sets Netcat to listen for incoming connections.
-p: this option sets the port to listen on.

Other flag options for client or server are:

-h: help mode
-d: run in background mode
-e: inbound program to execute
-z: zero-I/O mode [used for scanning ports]
-n: numeric IP address only
-u: run as UDP
-r: randomize local ports
-w: timeouts for connects
-v: verbose mode

3.1 Port Scanning

Port Scanning is a technique which is used to scan a system to determine which ports a system has opened and closed. For the first 1024 ports there are standard programs that typically run on known ports. Such as port 80 is for Web servers, 21 for File Transfer Servers, 22 for secure shells, 110 for POP e-mail servers and the list continues.

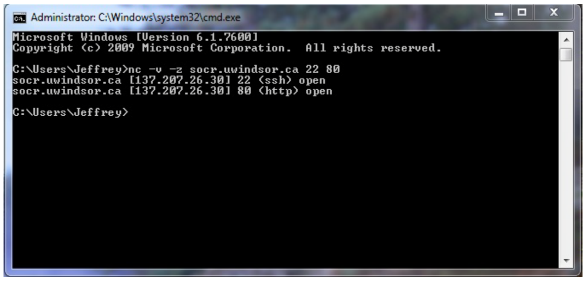
There are much better tools for port scanning such as Nmap which is a "network mapper" and will identify all hosts running on a network within a specified IP range as well as many of the services running on the hosts and other information about the Operating System and the versions of software running. Although Netcat allows for very similar and basic activity once you understand how to use Netcat it is just as powerful as other tools.

To run a port scan using Netcat you can use the following command:

nc –v –z hostname ports

For the host you can use an ip address or a dns name, and netcat will perform a dns lookup. Also for the ports you can list an individual port, a range of ports such as 80-90, or multiple different ports such as 22, 80.

In the example below we scan socr.uwindsor.ca to see if ports 22 (SSH) and 80 (Web Server) are open.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc -v -z socr.uwindsor.ca 22 80
socr.uwindsor.ca [137.207.26.30] 22 (ssh) open
socr.uwindsor.ca [137.207.26.30] 80 (http) open

C:\Users\Jeffrey>
```

As you can see it returns the hostname, IP address of the hostname, the ports scanned as well as the protocol their typically used for as well as the status of the port, being open or closed.

3.2 Banner Grabbing

Banner Grabbing is a technique used in order to determine which services are running on a specified port by requesting a connection to the server and the server returns basic information to let an application know which protocol to use to make a connection.

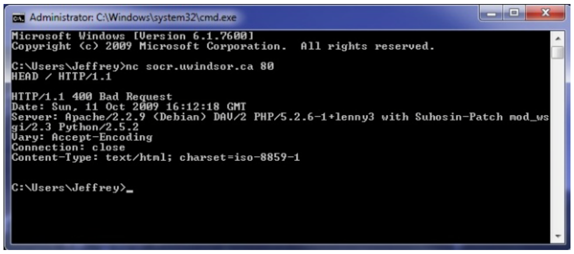
To do a simple banner grab simply try to connect to the server running the open port and send some information. For example to connect to a web server and see what is running you can type

nc socr.uwindsor.ca 80

Then once connected type:

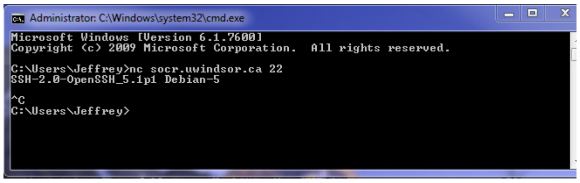
HEAD / HTTP/1.1

This will generate a simple request to a web server and return the web service running, the version of that service as well as any patches or add-ons. It will also return the host operating system it is running.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc socr.uwindsor.ca 80
HEAD / HTTP/1.1

HTTP/1.1 400 Bad Request
Date: Sun, 11 Oct 2009 16:12:18 GMT
Server: Apache/2.2.9 (Debian) DAV/2 PHP/5.2.6-1+lenny3 with Suhosin-Patch mod_ws
gi/2.3 Python/2.5.2
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=iso-8859-1


C:\Users\Jeffrey>_
```

The following example shows the information displayed when trying to connect to a secure shell running on the same host.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc socr.uwindsor.ca 22
SSH-2.0-OpenSSH_5.1p1 Debian-5

^C
C:\Users\Jeffrey>
```

Many other port scanning and vulnerability assessment tools also do a banner grab when doing a scan and inform you which services are running and which may have open vulnerabilities that can be exploited.

3.3 Basic Chat Interface

Netcat also has the ability to run as either the client or as the server, allowing it to play both roles. If a user wanted to they could run Netcat as a server on one machine and connect the client to it using the port that the server is running on. Once the client is connected to the server it can then communicate by using its own specific protocols if applied to an application. In our case we will just use it as a simple chat program.

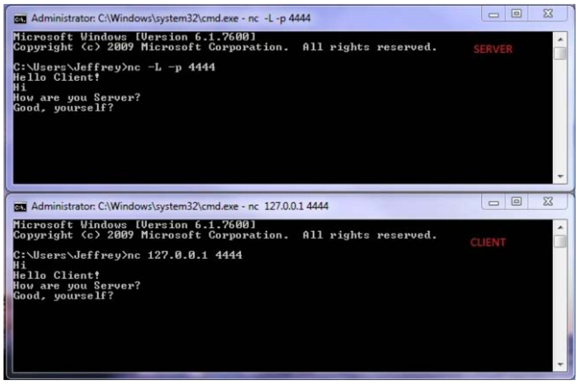
To run Netcat as a server type:

nc –L –p port

To run Netcat as a client and connect, type:

nc hostname port

The example below shows a server and client both running on the loopback interface of the same machine. Although this works the same way if the client and server are connecting remotely. In the image below the Server and Client are labeled, but the option flag –l or –L stand for listen, which will listen for connections, making it the server.


Administrator: C:\Windows\system32\cmd.exe - nc -L -p 4444
SERVER
Administrator: C:\Windows\system32\cmd.exe - nc 127.0.0.1 4444
CLIENT

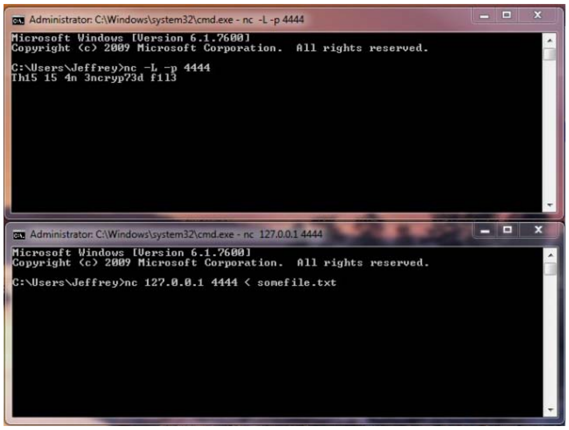
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc -L -p 4444
Hello Client!
Hi
How are you Server?
Good, yourself?
```

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc 127.0.0.1 4444
Hi
Hello Client!
How are you Server?
Good, yourself?
```

3.4 File Transfer

Since Netcat allows data to be transferred back and forth on the same connection, it doesn’t matter if it is plain text data or a file. They will both work the same way. The only difference is how you start the server and data. You can use the Input and Output flags on either the client or the server. Generally if you want to transfer a file from the client to the server you will use input on the client and select the filename to be transferred. While on the server if you know there’s a file being sent you will choose the output to be of the same filename or file type as the client is sending.

The following image depicts a client sending somefile.txt as the input to the server in this case just using the default output method. The server will display what the client has sent. If the server had run with the option > someotherfile.txt the content of the file being sent from the server would be saved into that file as the output method.


Administrator: C:\Windows\system32\cmd.exe - nc -L -p 4444
Administrator: C:\Windows\system32\cmd.exe - nc 127.0.0.1 4444

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc -L -p 4444
Th15 15 4n 3ncryp73d f113
```

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc 127.0.0.1 4444 < somefile.txt
```

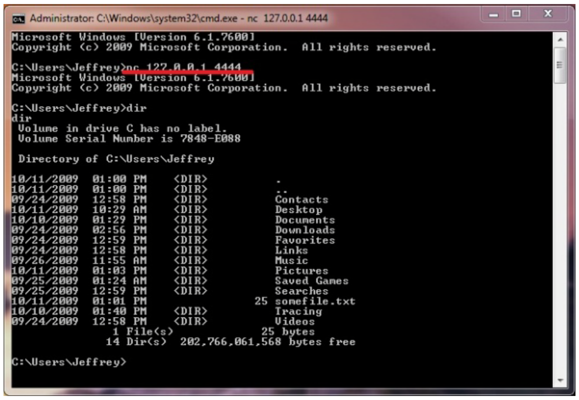
3.5 Creating a Backdoor

Backdoors are extremely useful to allow for remote access to a system because it opens up the ability to take control of the host you are connected to. Once a hacker connects to a backdoor it is important for them to create a new backdoor in which only they can connect to so that if another hacker tries to gain access they can’t take over the first hacker that connected. Usually a hacker will plant a Trojan and another method to allow unlimited access whenever they want as well as the use of a password so only they know it.

Netcat has the ability to run an external program upon creating a connection. This is incredible useful if one wants to load the Windows Command line as an external connection, or a Bash Shell to a UNIX environment, this will allow total control of the machine. To do this, on the server type:

nc –L –p 4444 –e cmd.exe

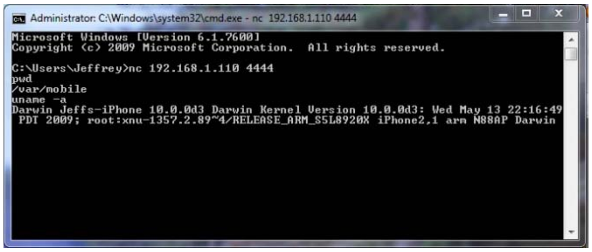
The –e flag is what will run the external program upon connecting to the server. For the client, just connect to the server with the port as normal and you will be given the windows prompt. The following image is what you would see connecting to a windows shell backdoor.


Administrator: C:\Windows\system32\cmd.exe - nc 127.0.0.1 4444

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc 127.0.0.1 4444
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is 7848-E088

 Directory of C:\Users\Jeffrey

10/11/2009  01:00 PM    <DIR>          .
10/11/2009  01:00 PM    <DIR>          ..
09/24/2009  12:58 PM    <DIR>          Contacts
10/11/2009  10:29 AM    <DIR>          Desktop
10/10/2009  01:29 PM    <DIR>          Documents
09/24/2009  02:56 PM    <DIR>          Downloads
09/24/2009  12:59 PM    <DIR>          Favorites
09/24/2009  12:58 PM    <DIR>          Links
09/26/2009  11:55 AM    <DIR>          Music
10/11/2009  01:03 PM    <DIR>          Pictures
09/25/2009  01:24 AM    <DIR>          Saved Games
09/25/2009  12:59 PM    <DIR>          Searches
10/11/2009  01:01 PM                25 somefile.txt
10/10/2009  01:40 PM    <DIR>          Tracing
09/24/2009  12:58 PM    <DIR>          Videos
               1 File(s)             25 bytes
              14 Dir(s)  202,766,061,568 bytes free

C:\Users\Jeffrey>
```

You can see after connecting to the server, you are prompted with a windows command line interface. Now you can perform windows commands and have expected output much like in the image above. Note, this client is connected to the same machine as the server, but it works similar if they are remote connections.

Take a look at the next image, where the Netcat server is now running on a UNIX environment on my phone, which is not a local connection, but now remote. When you connect to a UNIX system you are not given output like in a windows environment. By typing UNIX commands we get a response and that is how we know we are successfully connected and it is a UNIX system.


Administrator: C:\Windows\system32\cmd.exe - nc 192.168.1.110 4444

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Jeffrey>nc 192.168.1.110 4444
pwd
/var/mobile
uname -a
Darwin Jeffs-iPhone 10.0.0d3 Darwin Kernel Version 10.0.0d3: Wed May 13 22:16:49
 PDT 2009; root:xnu-1357.2.89~4/RELEASE_ARM_S5L8920X iPhone2,1 arm N88AP Darwin
```

As you can see by typing ‘*pwd*’ it returns your current working directory, in this case it is /var/mobile. Now when we type ‘*uname –a*’ it will print our kernel information:

Darwin Jeffs-iPhone 10.0.0d3 Darwin Kernel Version 10.0.0d3: Wed May 13 22:16:49
PDT 2009; root:xnu-1357.2.89~4/RELEASE_ARM_S5L8920X iPhone2,1 arm N88AP Darwin

IV. Experiment

In the experiment we will look more into some of the features that have been explained previously and compare them to other tools that offer the same functionality. As well as try these tools on different platforms.

I will compare port scanning against nmap which is designed for specifically scanning hosts on a network. I will test speed of each program for port scanning as well as information retrieved by both. I will demonstrate the use of these programs from the installation to the advanced techniques available.

Some of the different test environments I have access to locally attempt these tests on are:

Laptop:
AMD Turion 64 x2 2.00 GHz, 3.00 GB RAM
Windows 7 Professional

Desktop:
AMD Sempron 1.60 GHz, 960 MB RAM
Windows XP Professional, SP2

iPhone:
ARM Cortex A8 600 MHz, 256 MB RAM
iPhone OS

I will explore more in-depth features of Netcat as well as try and find many more uses to the already mentioned features.

V. Conclusion

In Conclusion you can see how useful Netcat is an many of its different unique features making it extremely powerful. It is listed as one of the best tools to use for Network Troubleshooting or debugging, but can also be used for a hacker to compromise a system from start to finish with just this tool alone. Any Network or Security Administrator should include this tool in their toolbox. In our experiment we will compare the benefit of using this tool compared to other tools out there that are similar.

VI. References

[1] Brian Baskin, *Netcat Power Tools*, Syngress Publishing Inc, Burlington, MA, 2008.

[2] Netcat: the TCP/IP swiss army: http://nc110.sourceforge.net

[3] Netcat. From Wikipedia: http://en.wikipedia.org/wiki/Netcat

[4] Ncat Users' Guide: http://nmap.org/ncat/guide/index.html

[5] Mati Aharoni, Netcat Security: http://www.webpronews.com/topnews/2003/10/20/netcat-security

[6] Nmap - Free Security Scanner For Network Exploration & Security Audits: http://nmap.org/

[7] Nmap. From Wikipedia: http://en.wikipedia.org/wiki/Nmap

[8] NetCat Tutorial: http://www.securitydocs.com/library/3376