CompTIA Network+® Lab Series
Network Concepts

Lab 3: TCP/IP Utilities

Objective 1.5: Identify common TCP and UDP default ports
Objective 1.6: Explain the function of common networking protocols
Objective 1.7: Summarize DNS concepts and its components
Objective 4.3: Given a scenario, use appropriate software tools to troubleshoot connectivity issues

Document Version: 2015-09-18

Development was funded by the Department of Labor (DOL) Trade Adjustment Assistance Community College and Career Training (TAACCCT) Grant No. TC-22525-11-60-A-48; The National Information Security, Geospatial Technologies Consortium (NISGTC) is an entity of Collin College of Texas, Bellevue College of Washington, Bunker Hill Community College of Massachusetts, Del Mar College of Texas, Moraine Valley Community College of Illinois, Rio Salado College of Arizona, and Salt Lake Community College of Utah.

This workforce solution was funded by a grant awarded by the U.S. Department of Labor's Employment and Training Administration. The solution was created by the grantee and does not necessarily reflect the official position of the U.S. Department of Labor. The Department of Labor makes no guarantees, warranties or assurances of any kind, express or implied, with respect to such information, including any information on linked sites, and including, but not limited to accuracy of the information or its completeness, timeliness, usefulness, adequacy, continued availability or ownership.

Contents

Introduction

This lab is part of a series of lab exercises designed to supplement coursework and provide students with a hands-on training experience based on real world applications. This series of lab exercises is intended to support courseware for CompTIA Network+® certification.

This lab will identify common commands used to gather information about nodes on a network. Students will execute these commands in both Windows and Linux environments to compare and contrast their outputs.

This lab includes the following tasks:

1. Displaying Computer Information Using the CLI
2. Displaying IP Information Using the CLI
3. Displaying DNS Information Using the CLI
4. Displaying Network Connections Using the CLI
5. Using Commands to Test Network Connectivity
6. Observing the ARP process using Wireshark®

Objective: Using CLI Tools to Gather Network Information

Troubleshooting a network involves gathering information about the nodes on the network. Many of the tools used for this purpose are run via the command line interface (CLI).

Key terms for this lab:

Cat – a Linux utility that concatenates and lists files

Man pages – *Manual Page*, a form of software documentation found on Linux machines used to provide help with concepts such as programs or command syntax

Domain Name System (DNS) – the protocol used to map hostnames and domain names into IP address on the Internet. DNS uses UDP port 53 for initiating requests

Fully Qualified Domain Name (FQDN) – the domain name that specifies the exact location of the specified node in the DNS hierarchy

Authoritative DNS Server – the master DNS server that hosts a specified domain

Non-authoritative DNS Server – a secondary DNS server that responds to DNS queries using cached DNS information

Alias – a secondary name assigned to a host within DNS – allows an administrator to provide multiple names that the same host can respond to

in-addr.arpa – the reverse lookup zone used by IPv4 to map IP addresses to DNS names
Socket – the combination of an IP address and a TCP or UDP port number separated by a colon (ex. 192.168.12.10:53)

Internet Control Message Protocol (ICMP) – a protocol within the TCP/IP suite that resides at the OSI Network Layer (Layer 3) used to send query or error messages to network nodes

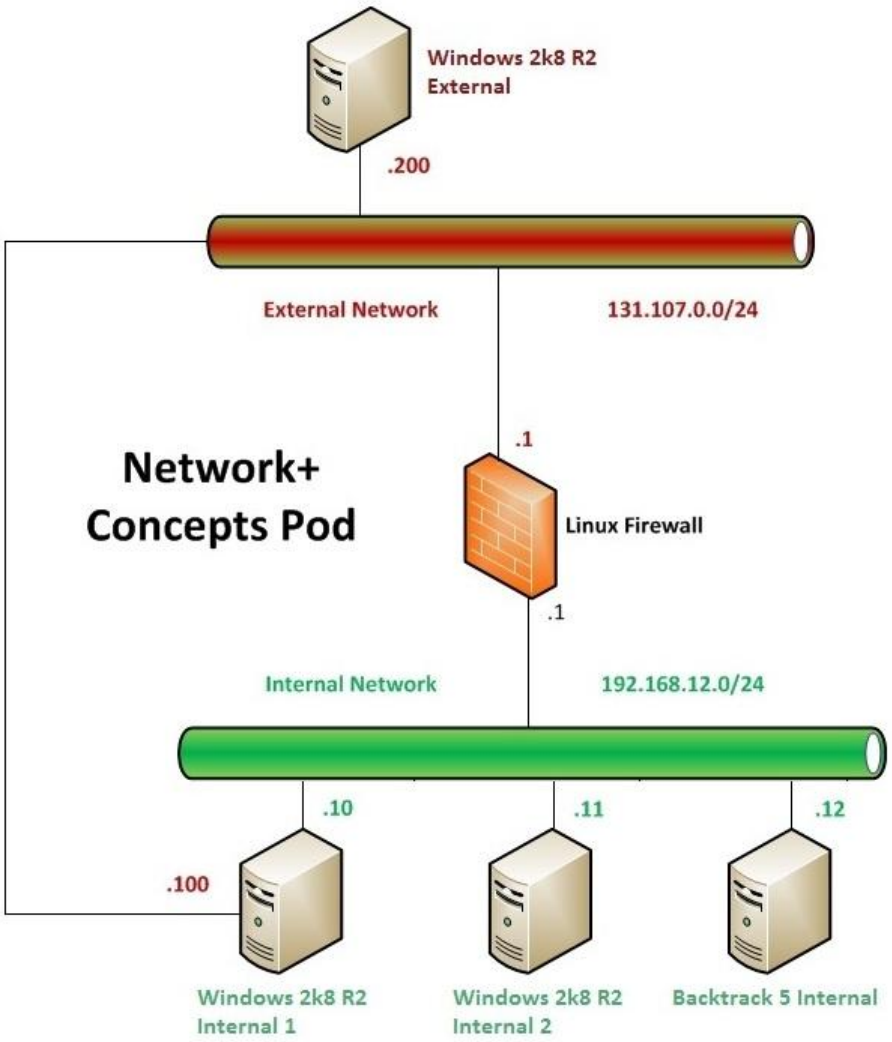
Time to Live (TTL) – a mechanism to specify the lifetime of data on a network

Address Resolution Protocol (ARP) – a protocol within the TCP/IP suite that resides at the OSI Network Layer (Layer 3) used to resolve network layer addresses (IP addresses) into link layer addresses (MAC addresses)

Media Access Control (MAC) address – the physical address burned into the ROM of an Ethernet network card; used by switches at the Data Link layer of the OSI model to move information between nodes on the same network

Wireshark - "is a network protocol analyzer. It lets you capture and interactively browse the traffic running on a computer network. It has a rich and powerful feature set and is world's most popular tool of its kind. It runs on most computing platforms including Windows, OS X, Linux, and UNIX. Network professionals, security experts, developers, and educators around the world use it regularly. It is freely available as open source, and is released under the GNU General Public License version 2."
Reference: http://www.wireshark.org

Pod Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

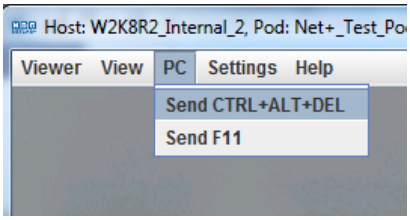
Required Virtual Machines and Applications

Log in to the following virtual machines before starting the tasks in this lab:

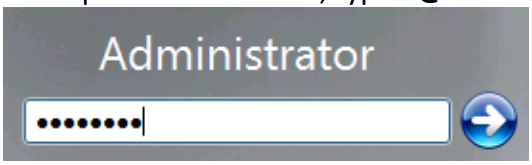
Windows 2k8 R2 Internal 1	192.168.12.10
Windows 2k8 R2 Internal 1 password	P@ssw0rd
Backtrack 5 Internal	192.168.12.12
Backtrack 5 Internal username/password	root/toor

Windows 2k8 R2 Login (applies to all Windows machines)

1. Click on the Windows 2k8 R2 icon on the topology that corresponds to the machine you wish to log in to.
2. Use the PC menu in the NETLAB+ Remote PC Viewer to send a **Ctrl-Alt-Del** (version 2 viewer), or click the **Send Ctrl-Alt-Del** link in the bottom right corner of the viewer window (version 1 viewer).



3. In the password text box, type **P@ssw0rd** and press enter to log in.



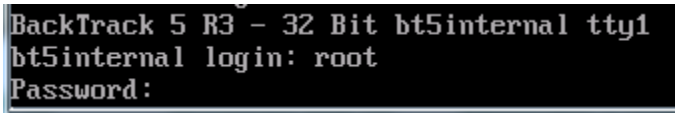
4. If the Initial Configuration Tasks and/or Server Manager windows appear, close them by clicking on the "**X**" in the top-right corner of the window

Backtrack 5 Internal Login

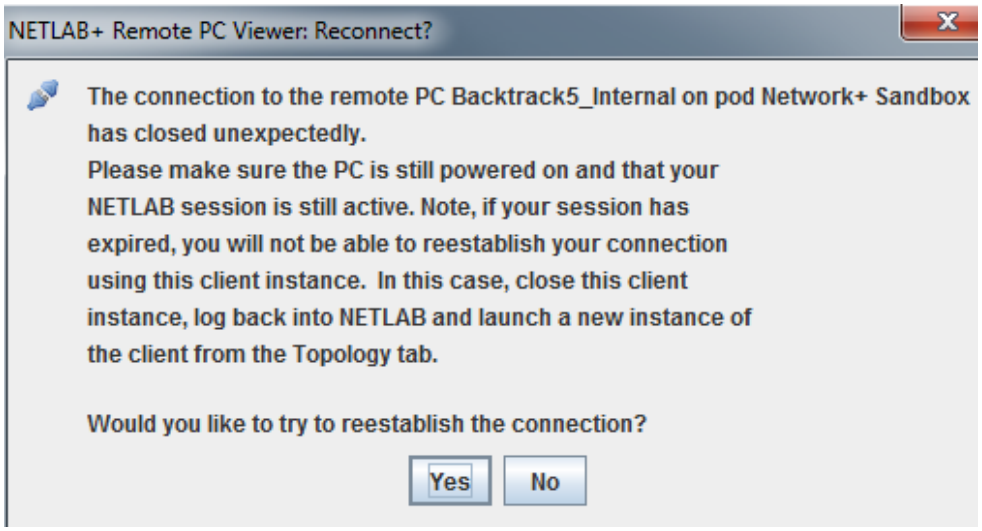
1. Click on the Backtrack 5 Internal icon on the topology.
2. At the **bt5internal login**: prompt, type the username **root** and press **Enter**.

3. At the **Password:** prompt, type the password **toor** and press **Enter**.

The password will not be displayed as you type into the prompt.



4. Once you have successfully logged in, type **startx** at the **root@bt5internal:~#** prompt and press **Enter**. This will start the GUI (Note: if you are disconnected after typing startx, click yes on the popup message to reconnect).



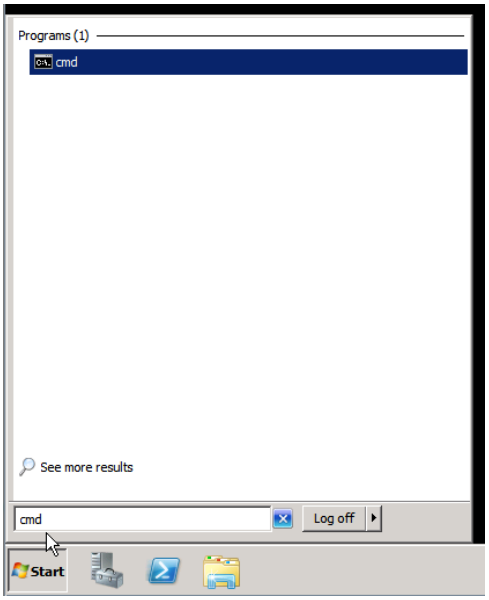
1 Displaying Computer Information

Knowing how to navigate in the command line is an essential part of troubleshooting for any network technician. There are a plethora of commands available in both Windows and Linux to help gather information about the system a user is on. Some commands even include switches (or command extensions) that can give more detailed information or even information about a remote computer. This section focuses on commands that help gather information about the local machine.

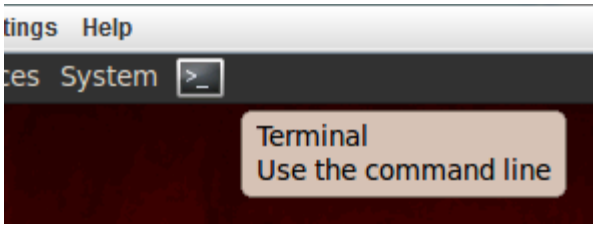
1.1 Displaying Computer Information Using the CLI

Keep in mind that **Linux commands are case sensitive**. The Linux commands below must be entered exactly as shown.

1. Use the instructions in the Lab Settings section to log into the Windows 2k8 R2 Internal 1 and Backtrack 5 Internal machines, if you are not logged in already.
2. For best results, arrange the NETLAB+ viewer windows so they are side-by-side on your computer screen.
3. On the Windows 2k8 R2 Internal 1 machine, click the **Start** menu. In the **Search programs and files** dialog box, type **cmd** and press **Enter** to gain access to the command prompt.

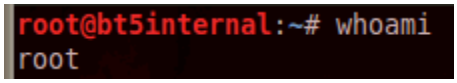


4. On the Backtrack 5 Internal machine, click the icon to the right of the **System** menu to gain access to the terminal window.

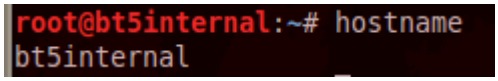


5. Sometimes when troubleshooting a machine, a user may need to know who they are logged in as. This is useful information, especially when attempting to troubleshoot permission issues. Interestingly, the command is the same in both Windows and Linux. In the command line interface on both machines, type the command **whoami**. On the Windows 2k8 R2 Internal 1 machine, notice this command gives the full login context (i.e. computername\username – if the user was logged into a domain, the context would be presented as domainname\username).

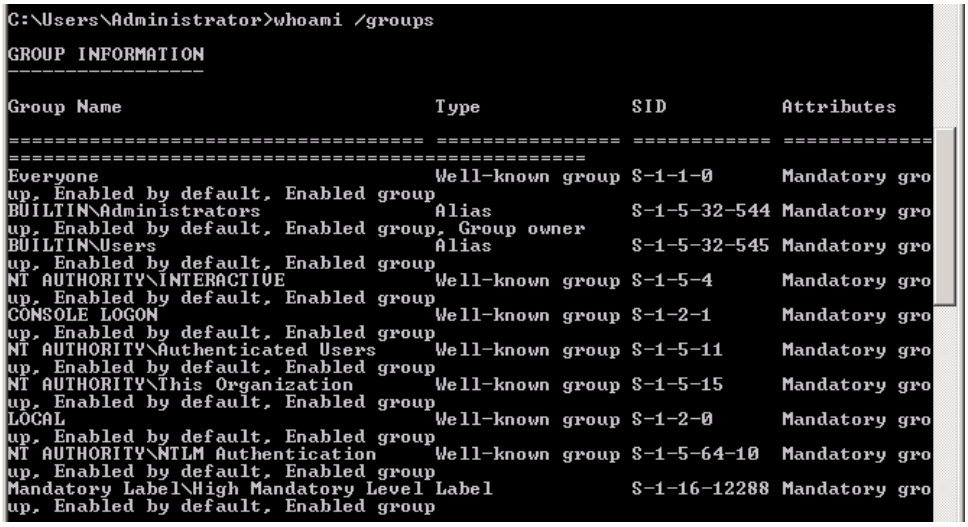
6. On the Backtrack 5 Internal machine, type the same command, **whoami**, into the terminal window. Notice that this command only gives the current username.



7. A second command can be used on either machine to determine the hostname of the machine (if it is not readily apparent). This command is simply **hostname**. Type this command into both command-line interfaces and press **Enter**.



8. Many, but not all, commands can be altered using switches, or command extensions, that modify the command to give a different output. One example is the **whoami** command in Windows. Adding the **/groups** switch to the command will display all of the groups the current user belongs to.



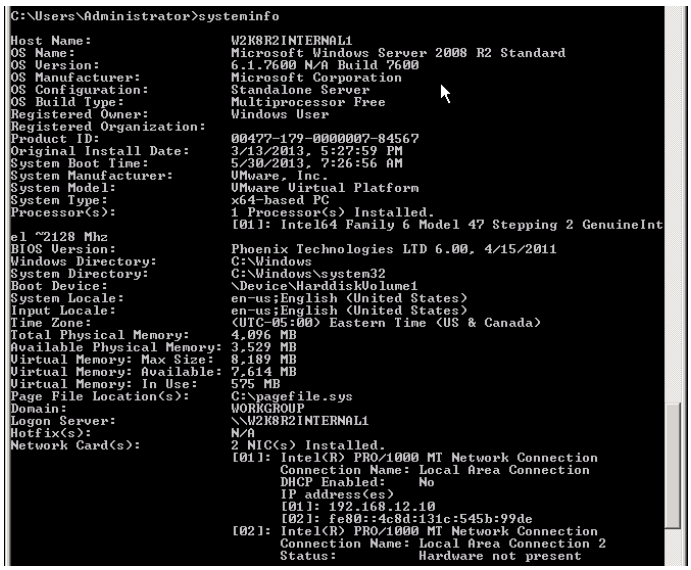
To see all of the switches that are available for a command in Windows, **/?** can typically be added after the command. This will also give the syntax for how to use the command and sometimes even a brief description of what the command does. Try this for the whoami command.

1. *How many switches are available for this command in Windows?* *10 (including /?)*

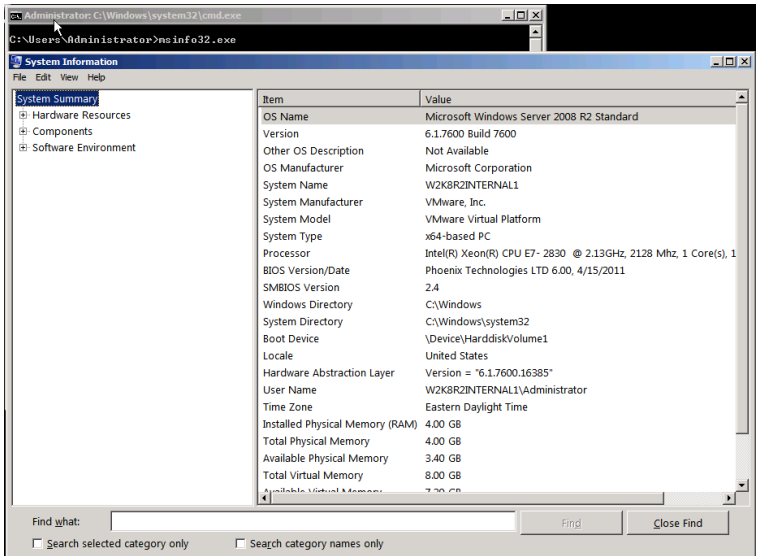
9. In Linux, adding the switch **--help** will usually display similar information for its commands. Try this for the **hostname** command. Notice that this command in Linux can also be used to set the hostname for the machine, whereas in Windows it can only view it.

10. To find more detailed information about a system, Windows and Linux commands start to vary greatly. Typically in Windows, the command that is run is an applet or executable that gathers the information. Typically in Linux, a user is looking directly at the file associated with the information they are attempting to gather.

a. Windows has two commands that can be used to gather more detailed system information. The **systeminfo** command run directly in the command line interface and displays information such as hostname, OS version, installation date and hotfixes applied (and a lot more). Type this command into the command line interface on the Windows 2k8 R2 Internal 1 machine and view its output.



b. The GUI version of the command can be launched by using the msinfo32.exe command. This command launches the System Information applet that displays much of the same information. This applet also includes a search feature to help find information more quickly. Search for the phrase "virtual memory". How many entries does it find? (NOTE: You may have to click the Find Next button a few times!) Close the System Information applet once completed.



c. The majority of hardware information can be extracted from the /proc filesystem in Linux. The command **cat /proc/cpuinfo** displays information about the CPU. Type this command into the terminal window on the Backtrack 5 Internal 1 machine and view its output.

Be sure to type a space between cat and /proc to use the command correctly. Syntax is very important when using the command line.

d. The command **cat /proc/meminfo** displays information about the memory currently available to the Linux system. Type this command into the terminal window to view its output.

```
root@bt5internal:~# cat /proc/meminfo
MemTotal:        2062204 kB
MemFree:         1823556 kB
Buffers:           30524 kB
Cached:           116452 kB
SwapCached:            0 kB
Active:            95632 kB
Inactive:         116660 kB
Active(anon):      65832 kB
Inactive(anon):     6472 kB
Active(file):      29800 kB
Inactive(file):   110188 kB
Unevictable:           0 kB
Mlocked:               0 kB
HighTotal:       1187784 kB
HighFree:        1002280 kB
LowTotal:         874420 kB
LowFree:          821276 kB
SwapTotal:       1764348 kB
SwapFree:        1764348 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:         65332 kB
Mapped:            38884 kB
Shmem:              6992 kB
Slab:              16664 kB
SReclaimable:       8828 kB
SUnreclaim:         7836 kB
KernelStack:        1384 kB
PageTables:         1616 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     2795448 kB
Committed_AS:     288888 kB
VmallocTotal:     122880 kB
VmallocUsed:        7592 kB
VmallocChunk:     110972 kB
HardwareCorrupted:     0 kB
```

11. Keep all windows open to continue on to the next task section.

Many commands will work between the various Linux versions, but this is not always the case. There may also be times when there are multiple commands that will display nearly the same information. This is partly because of the fact that many Linux versions are open-sourced and the Linux community has developed multiple ways to get the same information. The man pages within Linux or the Internet can be used for help with issuing commands in Linux.

1.2 Conclusion

Knowing how to navigate in the command line is an essential part of troubleshooting for any network technician. There are a plethora of commands available in both Windows and Linux to help gather information about the system a user is on. Some commands even include switches (or command extensions) that can give more detailed information or even information about a remote computer.

1.3 Review Questions

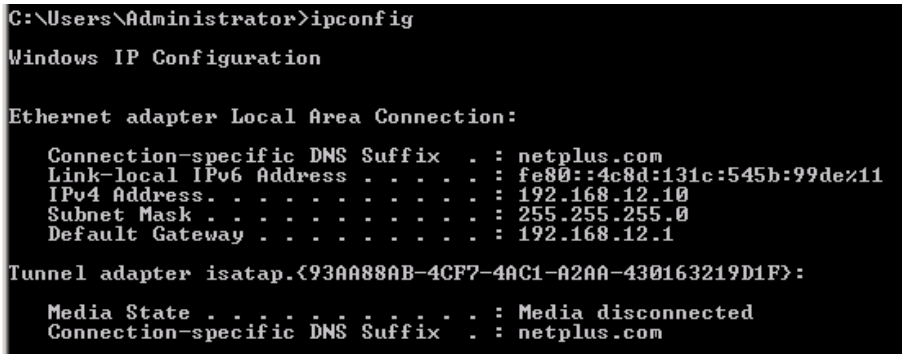
1. *What is the command used to display the current user in Windows and Linux?*
2. *What two commands can be used to obtain system information win Windows?*
3. *What command can be used to display CPU information in Linux?*

2 Displaying IP Information

It is often necessary when troubleshooting that a user needs to gather the IP information from the machine they are working with. The commands within Windows and Linux are similar, but each has their own syntax. Becoming familiar with these commands and their available switches will greatly assist in troubleshooting.

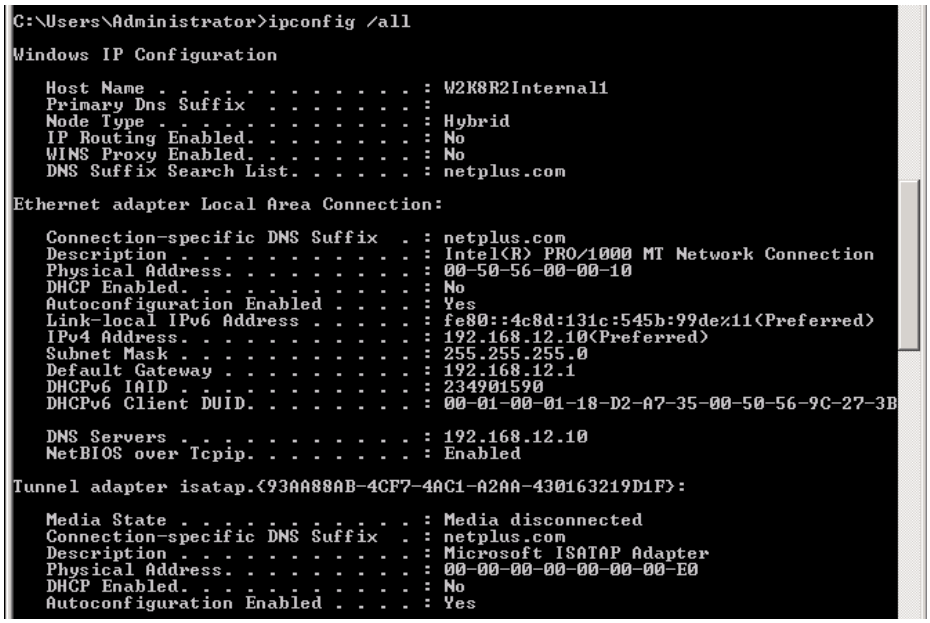
2.1 Displaying IP Information Using the CLI

1. Using the Windows 2k8 R2 Internal 1 machine, to display basic IP information in Windows, type the command **ipconfig** into the command prompt and press **Enter**.



Notice that this command provides the basic IP information such as IP address, subnet mask and default gateway.

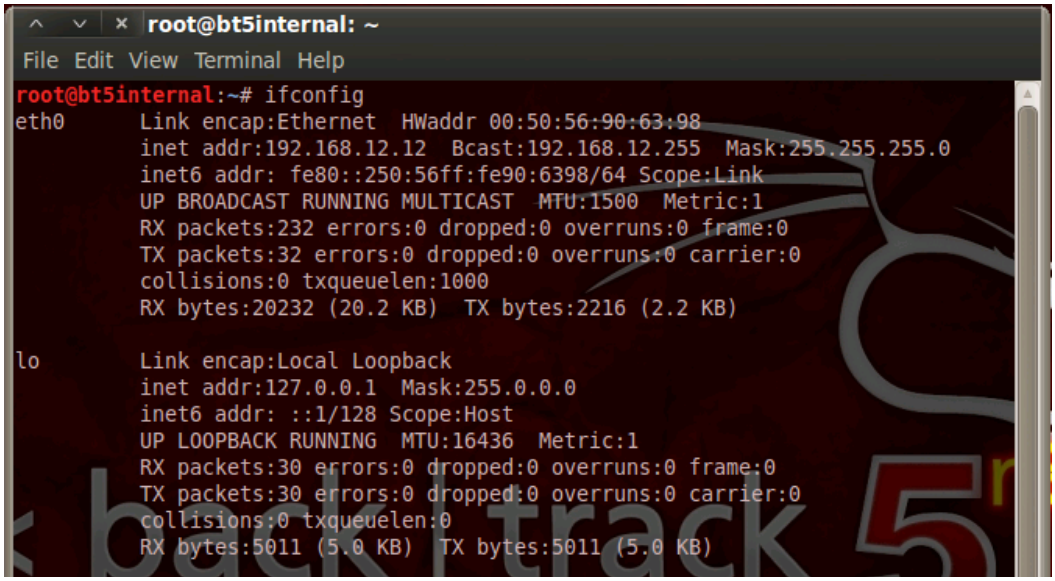
2. To provide detailed IP information, type the command **ipconfig /all** into the command prompt and press **Enter**.



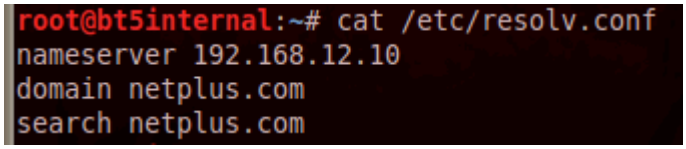
Notice some of the additional information that is displayed with the output of this command. Some examples include DHCP enabled and available DNS servers.

3. To display the available command switches available for the **ipconfig** command, type **ipconfig /?** Into the command prompt and press **Enter**.

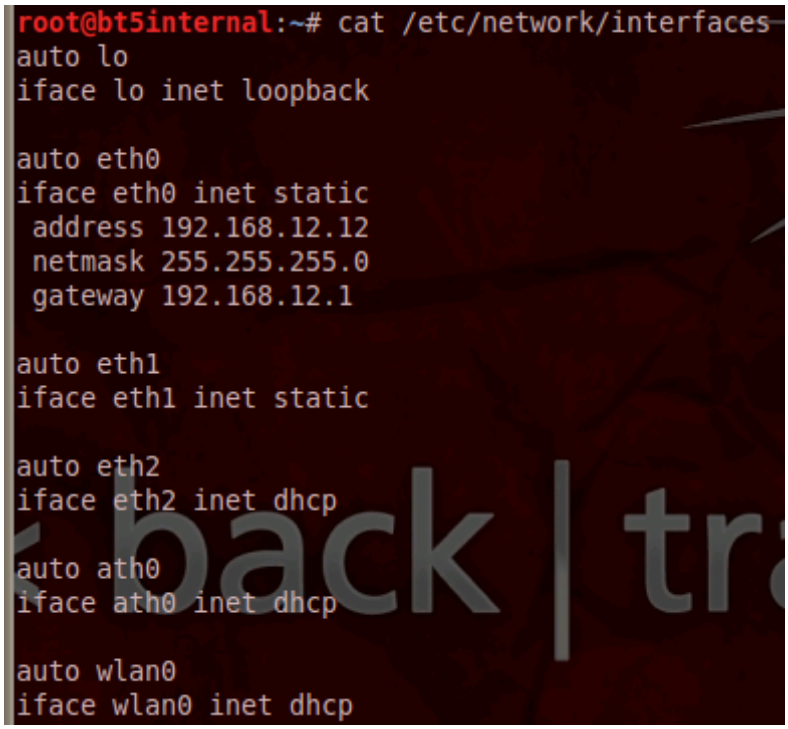
4. To display IP information in a Linux terminal, type the command **ifconfig** and press **Enter**.



Notice that this command displays the same basic information as the **ipconfig** command in Windows. However, there is no switch available (such as **/all**) to display more detailed information for the **ifconfig** command. For example, if a user wants to view the DNS servers configured on a Linux machine, they must view the configuration file associated with DNS. Type the command **cat /etc/resolv.conf** into the terminal window and press **Enter**. The DNS information for the Linux machine will be displayed.



5. To view the configuration file associated with the network card, type the command **cat /etc/network/interfaces** into the terminal window and press **Enter**. Changing the information in this file (such as the IP address) will make it persistent across reboots. Notice that the interface named **eth0** has its IP information set statically.



6. Keep all windows open to continue on to the next task section.

2.2 Conclusion

It is often necessary when troubleshooting that a user needs to gather the IP information from the machine they are working with. The commands within Windows and Linux are similar, but each has their own syntax. Becoming familiar with these commands and their available switches will greatly assist in troubleshooting.

2.3 Review Questions

1. *What command can be used to display the IP address, subnet mask and default gateway in Windows?*

2. *What switch can be added to the above command to also view IP information such as DNS and DHCP servers?*

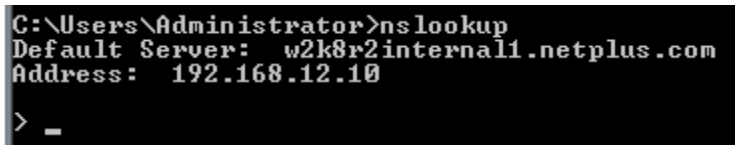
3. *What command can be used to display the IP address, subnet mask and default gateway in Linux?*

3 Displaying DNS Information

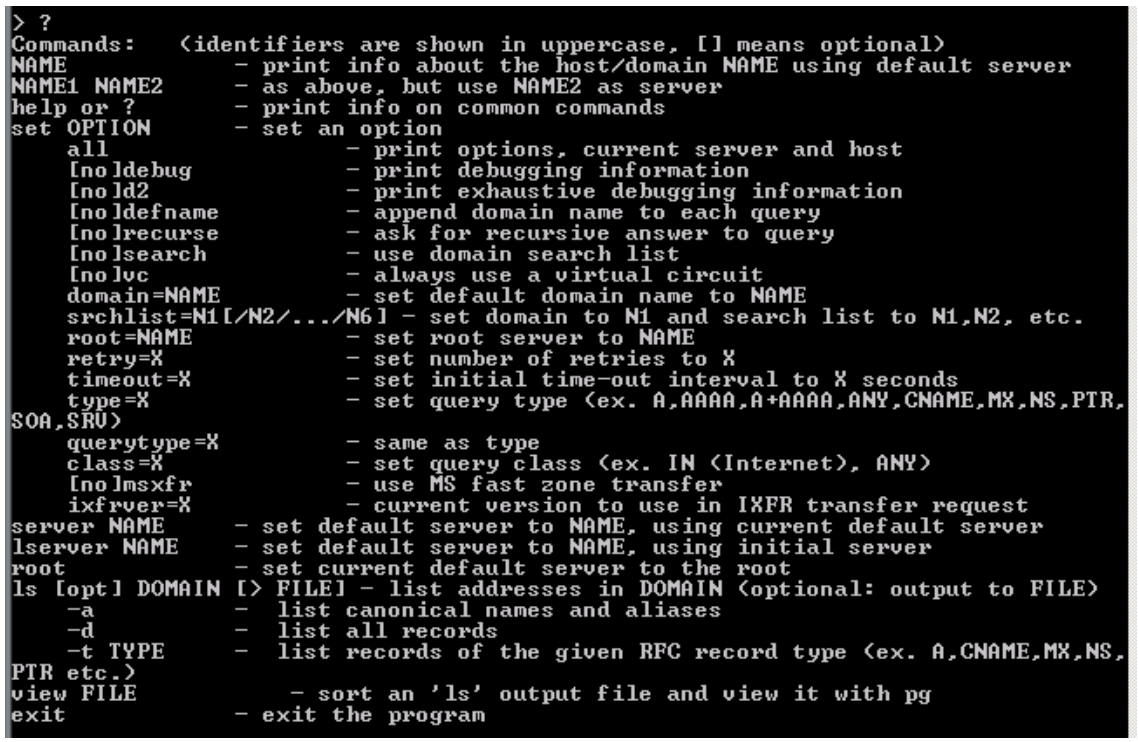
Looking up DNS information is similar between Windows and Linux using the nslookup command. This command can be used in two modes – interactive or non-interactive mode. In interactive mode, the user can query name servers about various hosts and domains as well as print a list of hosts. In non-interactive mode, the host or domain is specified in the command and only information pertaining to that host or domain is returned.

3.1 Displaying DNS Information Using the CLI

1. Using the Windows 2k8 R2 Internal 1 machine, to enter interactive mode for the **nslookup** command in Windows, type **nslookup** into the command prompt and press **Enter**. Notice the prompt changes to a **>** and the cursor sits blinking waiting on input. The default server that will be used for queries and its IP address are displayed as well.



2. To see a list of commands available in interactive mode, type **?** and press **Enter**.



3. To find DNS information for a host, only type the name or IP address of that host at the prompt.
    a. Type the IP address **192.168.12.12** into the command prompt window and press **Enter**.

```
> 192.168.12.12
Server:  w2k8r2internal1.netplus.com
Address:  192.168.12.10

Name:    bt5internal.netplus.com
Address:  192.168.12.12
```

Notice the two pieces of information from this output. The first section tells the DNS server's Fully Qualified Domain Name (FQDN) and IP address that responded to the query. The second portion is the answer to the query returned by the server giving the FQDN and IP address of the host.

    b. Type the FQDN **www.isp.com** into the command prompt window and press **Enter**.

```
> www.isp.com
Server:  w2k8r2internal1.netplus.com
Address:  192.168.12.10

Non-authoritative answer:
Name:    w2k8r2external.isp.com
Address:  131.107.0.200
Aliases:  www.isp.com
```

Notice that the same server returned the requested information. However, this time there is some additional information returned about the host. Look at the line Non-authoritative answer: This statement means that the server answering the query is doing so based upon the best information that it has about that host (i.e. that host is not in its local DNS database). This server could have found this information within its DNS cache or by querying another DNS server to find the information.

Next, look at the lines Name and Aliases. Notice that the name of the server is actually different than the query, but that it matches the alias. This is a common practice in DNS. The actual name of the physical server typically means more to the administrator for the domain the server belongs to than it does to anyone outside of that network. Therefore, an administrator can create an alias, or additional name, for that server to be able to respond to. It's like giving the server a nickname. In this case, this server also acts as a web server. Therefore, the administrator gave the server the alias www to allow the server to respond to this additional name. Nearly all web servers on the Internet that respond to the name www have been configured with this name as an alias.

c. If an administrator allows it, a list of all addresses in a domain can also be viewed. Type the command **ls netplus.com** into the command prompt windows and press **Enter**.

```
> ls netplus.com
[w2k8r2internal1.netplus.com]
 netplus.com.                   NS      server = w2k8r2internal1.netplus.com
 bt5internal                    A       192.168.12.12
 w2k8r2internal1                A       192.168.12.10
 w2k8r2internal2                A       192.168.12.11
>
```

The first column lists the names returned by the server. The second column lists the DNS record type. An **NS** record identifies a DNS server in the specified domain while an **A** record identifies a host in the specified domain. The last column is the information about the record, such as the DNS server name or the IP address of the host.

1. *Using the information from the table, what is the actual IP address of the DNS server for the **netplus.com** domain?*

d. To exit interactive mode for the **nslookup** command, type **exit** into the command prompt window and press **Enter**.

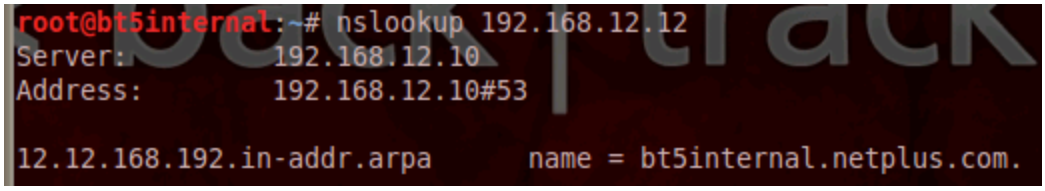
```
> exit

C:\Users\Administrator>
```

4. The **nslookup** command can be used in non-interactive mode to perform a lookup of a single domain, FQDN or IP address. On the Backtrack 5 Internal machine, type the command: **nslookup www.isp.com**.

```
root@bt5internal:~# nslookup www.isp.com
Server:         192.168.12.10
Address:        192.168.12.10#53

Non-authoritative answer:
www.isp.com     canonical name = w2k8r2external.isp.com.
Name:   w2k8r2external.isp.com
Address: 131.107.0.200
```

While the output looks slightly different in Linux, the information returned is still the same. One thing to notice is the **#53** on the end of the address of the server. This is the well-known port number assigned to DNS.

5. To perform a lookup by IP address, type the command **nslookup 192.168.12.12** into the terminal window and press **Enter**.



   Notice the last line in the output, specifically **12.12.168.192.in-addr.arpa**. Performing a DNS lookup by IP address is known as a reverse lookup. In this case, the command actually returns the specific record name returned by the DNS server. Notice that it is the IP address of the machine listed in reverse order followed by the special reverse-lookup domain name, **in-addr.arpa**.
6. Keep all windows open to continue with the next task section.

3.2 Conclusion

Looking up DNS information is similar between Windows and Linux using the nslookup command. This command can be used in two modes – interactive or non-interactive mode. In interactive mode, the user can query name servers about various hosts and domains as well as print a list of hosts. In non-interactive mode, the host or domain is specified in the command and only information pertaining to that host or domain is returned.

3.3 Review Questions

1. *What command is used to lookup DNS information in Windows and Linux?*

2. *What two modes can the above command be executed in?*

3. What special domain is associated with reverse DNS lookups?

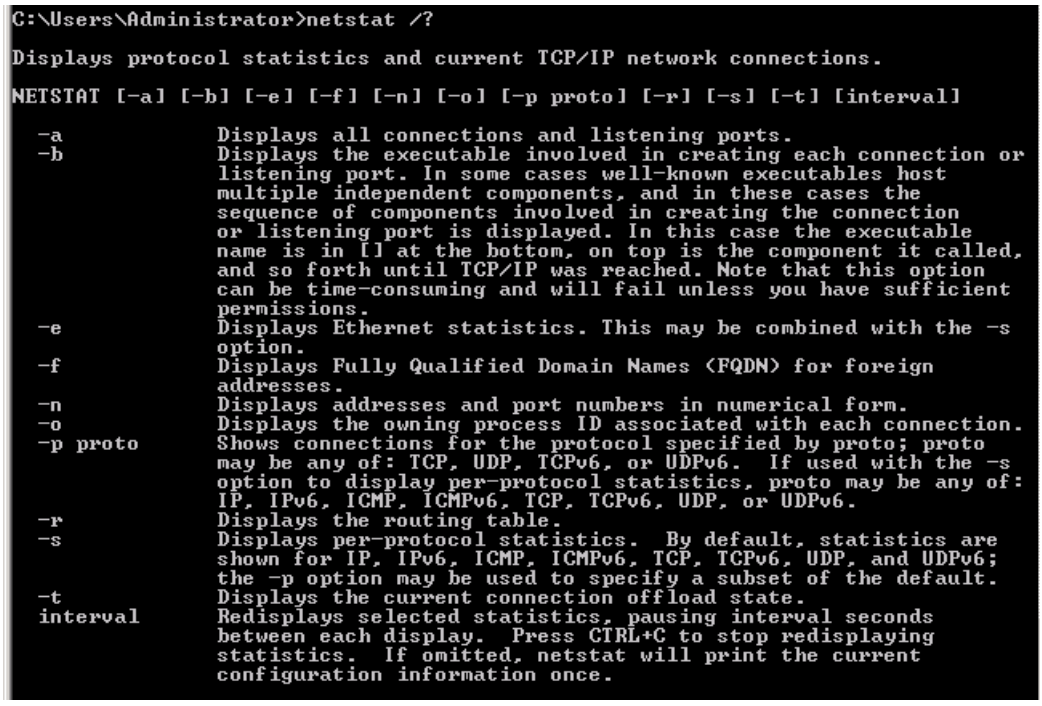
4 Displaying Network Connections

Netstat, or network statistics, is a command-line tool that displays active TCP connections (both incoming and outgoing) as well as other network statistics. When used with the appropriate command switches, it also displays ports on which the computer is listening, the IP routing table, Ethernet and IP statistics. The utility is available in both Windows and Linux.

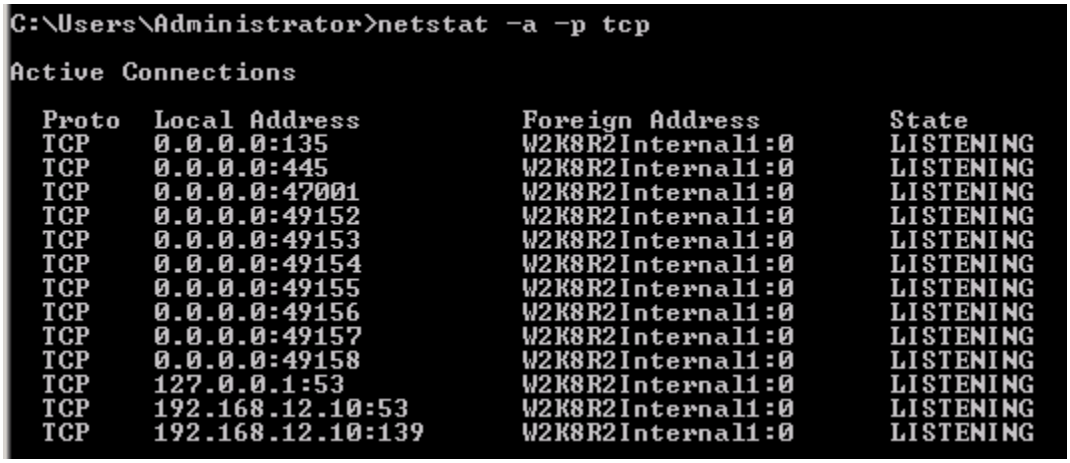
4.1 Displaying Network Connections Using the CLI

1. Use the instructions in the Lab Settings section to log into the Windows 2k8 R2 Internal 1 and Backtrack 5 Internal machines, if you are not logged in already.
2. On the Windows 2k8 R2 Internal 1machine, running the **netstat** command in Windows without any switches defaults to displaying only active connections. As such, running this command on the Windows machine will probably not display any information.

3. Therefore, we need to determine what command switches are available to us to display usable information. Type **netstat /?** into the command prompt on the Windows 2K8R2 Internal 1 machine and press **Enter**.



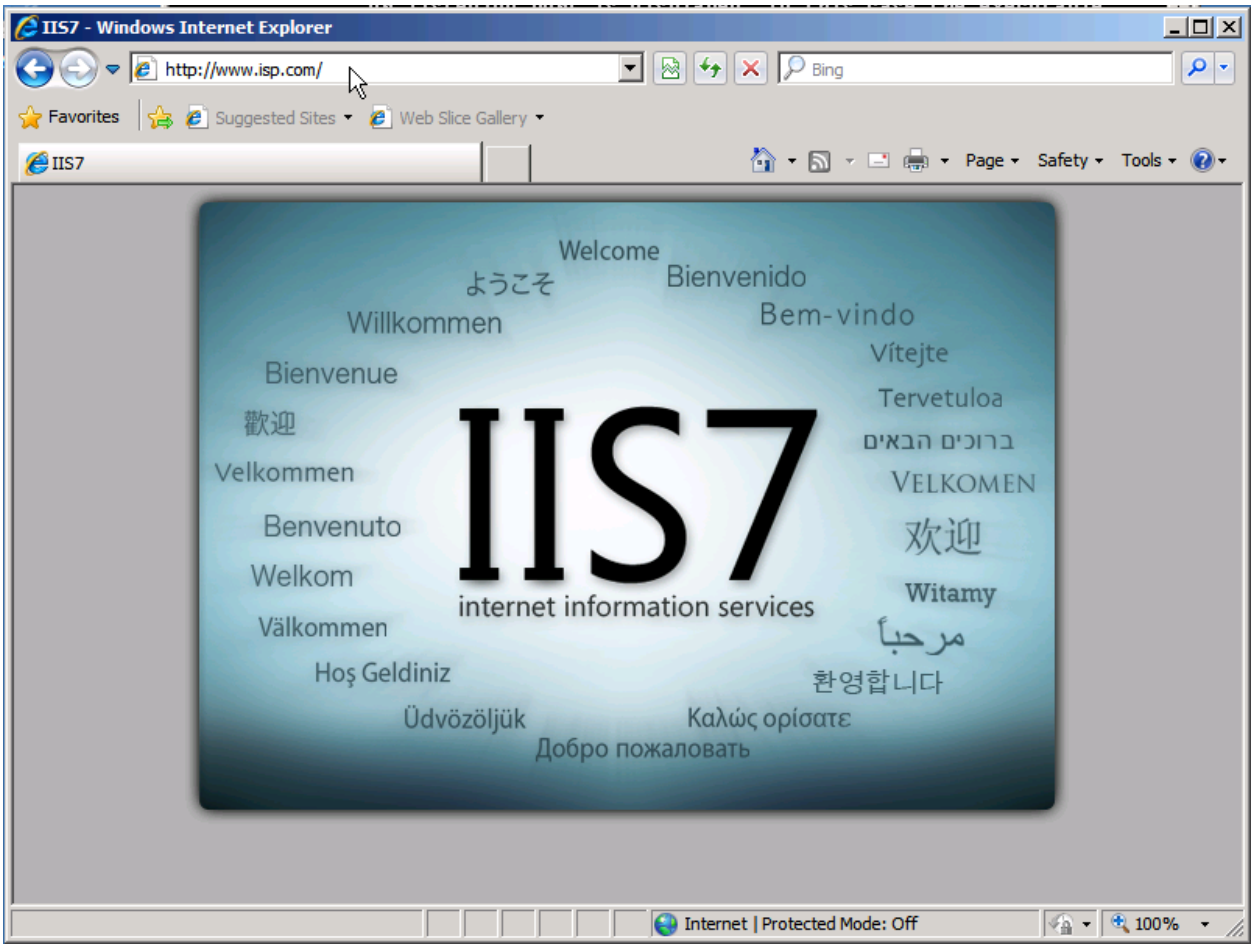
4. Review the output to examine the function of various switches. To list all TCP ports that are in the **Listening** state, type **netstat –a –p tcp** into the command prompt and press **Enter**. The **–a** command switch displays all connections and listening ports. The **–p tcp** command switch displays just those ports associated with the TCP protocol. This also helps make the output of the command more reasonable.



The first column is the protocol; the second column is the socket (IP address and port number) that the computer is listening on; the third column lists the remote machine of established connections (in this case it is the local machine since it is listening); the fourth column is the state of the connection. The listening state means the machine is ready to accept a connection while the established state means the connection is active.

Notice specifically the line with a local address of **192.168.12.10:53**. Remember from the previous task that port 53 was the well-known port number assigned to DNS. Therefore, we can conclude that this machine must be a DNS server as it is listening for requests on this port.

5. To establish a connection, we need to generate some network traffic. The easiest way to do this is by opening a web page. Leave the command prompt open and open a browser window by clicking **Start -> Internet Explorer**. In the address bar, type **www.isp.com** and press **Enter**. This is the web server running on the W2K8R2 External machine.



6. Leave the Internet Explorer window open and click on the command prompt window. Type the command **netstat** and press **Enter**. The command now displays the one active HTTP connection. Note that the port number attached to the Local Address is randomly generated and may vary from the figure.

```
C:\Users\Administrator>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.12.10:49161    w2k8r2external:http    ESTABLISHED
```

If the connection disappears, simply click on the Internet Explorer window and refresh the page. This will establish a new connection to the web server. This will also change the port number, but that does not affect the output of the command.

7. To display the FQDN of the remote server, type the command **netstat –f** and press **Enter**.

```
C:\Users\Administrator>netstat -f

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.12.10:49161    w2k8r2external.isp.com:http  ESTABLISHED
```

8. To display the IP address of the remote server, type the command **netstat –n** and press **Enter**.

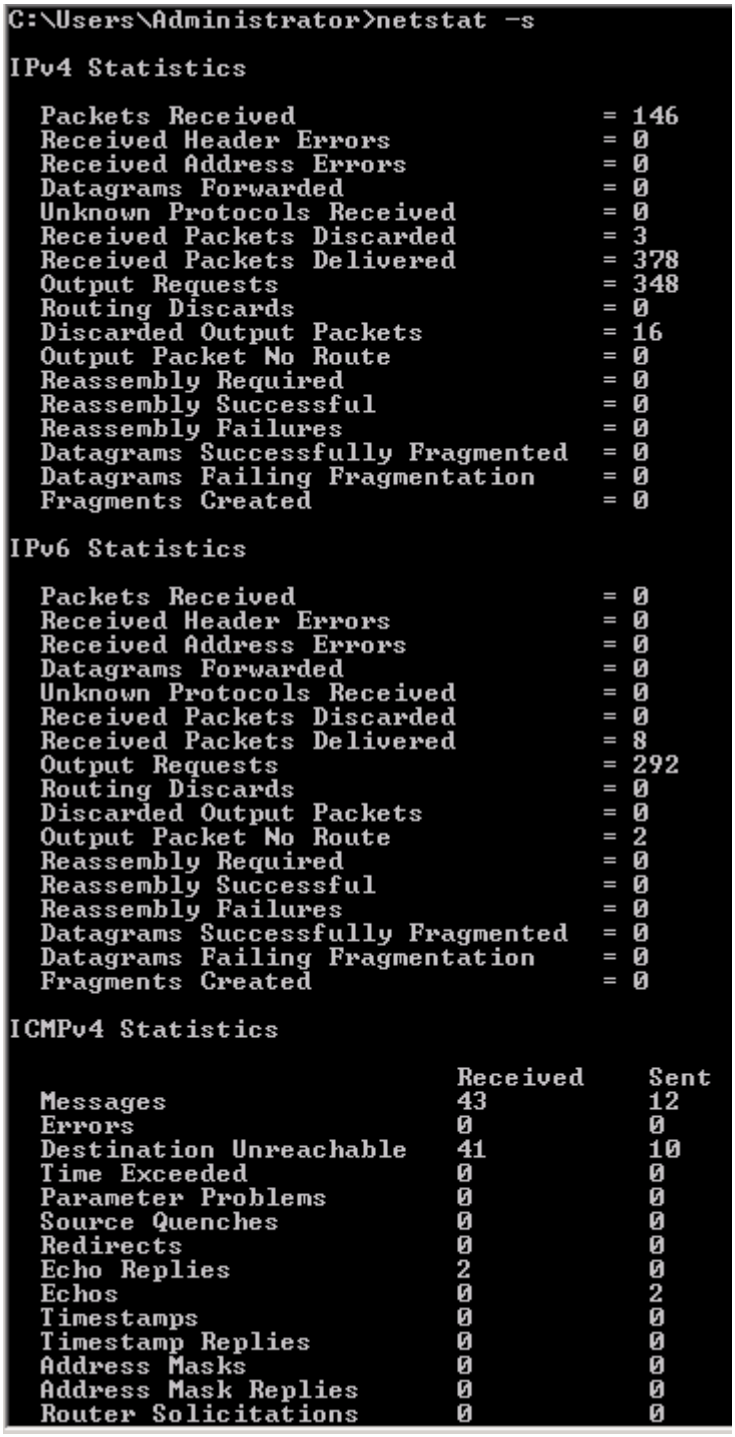
```
C:\Users\Administrator>netstat -n

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.12.10:49161    131.107.0.200:80       ESTABLISHED
```

9. To display Ethernet statistics, type the command **netstat –e** and press **Enter**.

```
C:\Users\Administrator>netstat -e
Interface Statistics

                           Received            Sent

Bytes                      49266          186114
Unicast packets              387            1272
Non-unicast packets            0               0
Discards                       0               0
Errors                         0               0
Unknown protocols              0
```

The output of this command is useful to see if errors are occurring on connections.

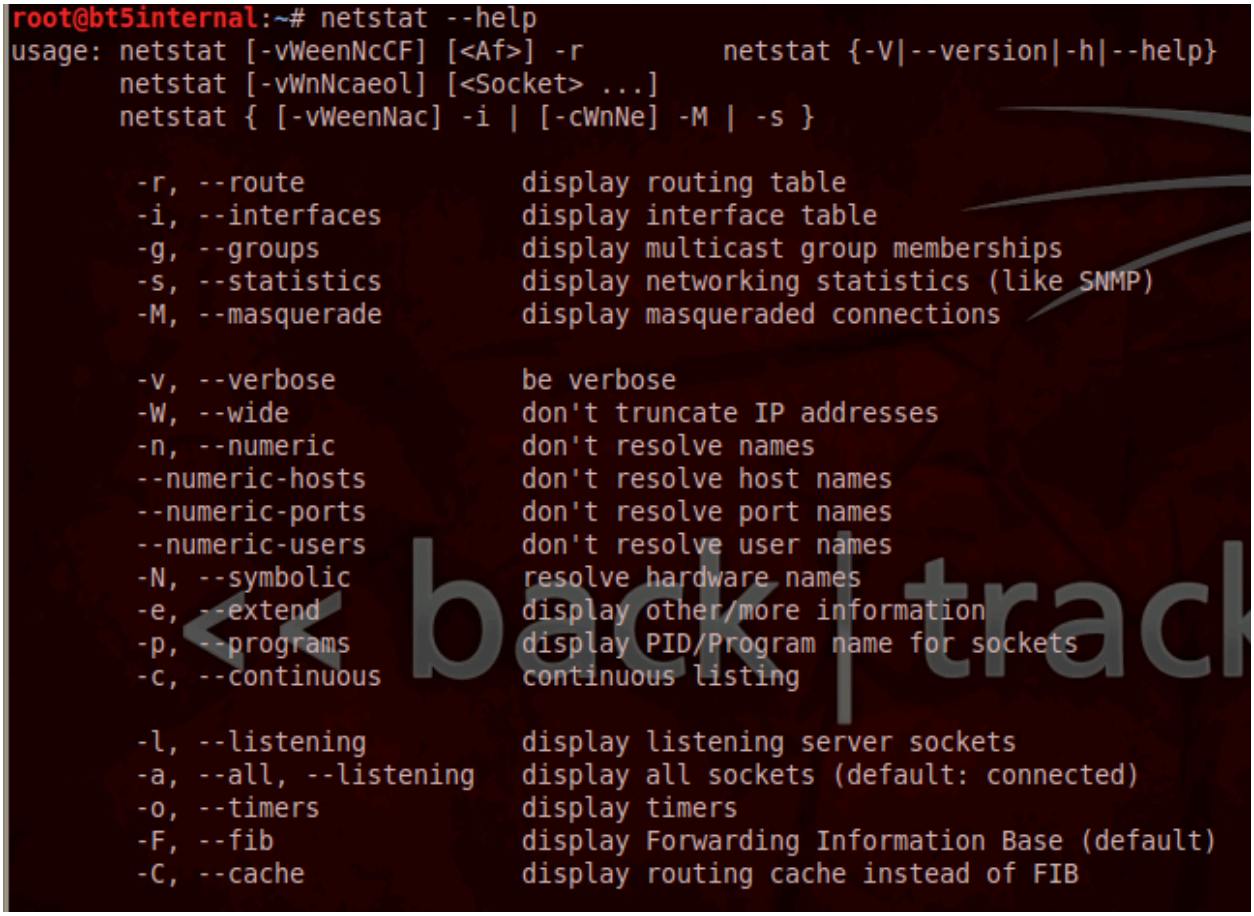
10. To display statistics about other protocols, type the command **netstat –s** and press **Enter**.



```
C:\Users\Administrator>netstat -s

IPv4 Statistics

  Packets Received                   = 146
  Received Header Errors             = 0
  Received Address Errors            = 0
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 3
  Received Packets Delivered         = 378
  Output Requests                    = 348
  Routing Discards                   = 0
  Discarded Output Packets           = 16
  Output Packet No Route             = 0
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0

IPv6 Statistics

  Packets Received                   = 0
  Received Header Errors             = 0
  Received Address Errors            = 0
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 0
  Received Packets Delivered         = 8
  Output Requests                    = 292
  Routing Discards                   = 0
  Discarded Output Packets           = 0
  Output Packet No Route             = 2
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0

ICMPv4 Statistics

                            Received    Sent
  Messages                  43          12
  Errors                    0           0
  Destination Unreachable   41          10
  Time Exceeded             0           0
  Parameter Problems        0           0
  Source Quenches           0           0
  Redirects                 0           0
  Echo Replies              2           0
  Echos                     0           2
  Timestamps                0           0
  Timestamp Replies         0           0
  Address Masks             0           0
  Address Mask Replies      0           0
  Router Solicitations      0           0
```

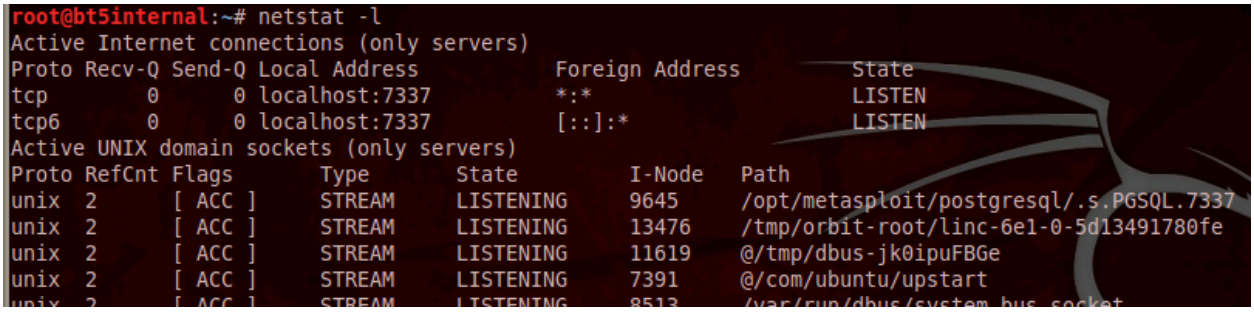
Scrolling up through the output reveals that this command gives statistics for several protocols (IP, ICMP, TCP and UDP). Protocol statistics can be viewed individually by specifying the protocol in the command. For example, TCP statistics can be viewed by typing the command **netstat –s –p tcp**.

11. The **netstat** command in Linux can be used to display much of the same information. Some of the command switches are even the same from Windows to Linux. Compare the available switches on the Backtrack 5 Internal machine by typing the command **netstat --help** into the terminal window and press **Enter**. Pay close attention to the available switches. Much of the same information can be obtained, but a different switch may need to be used.



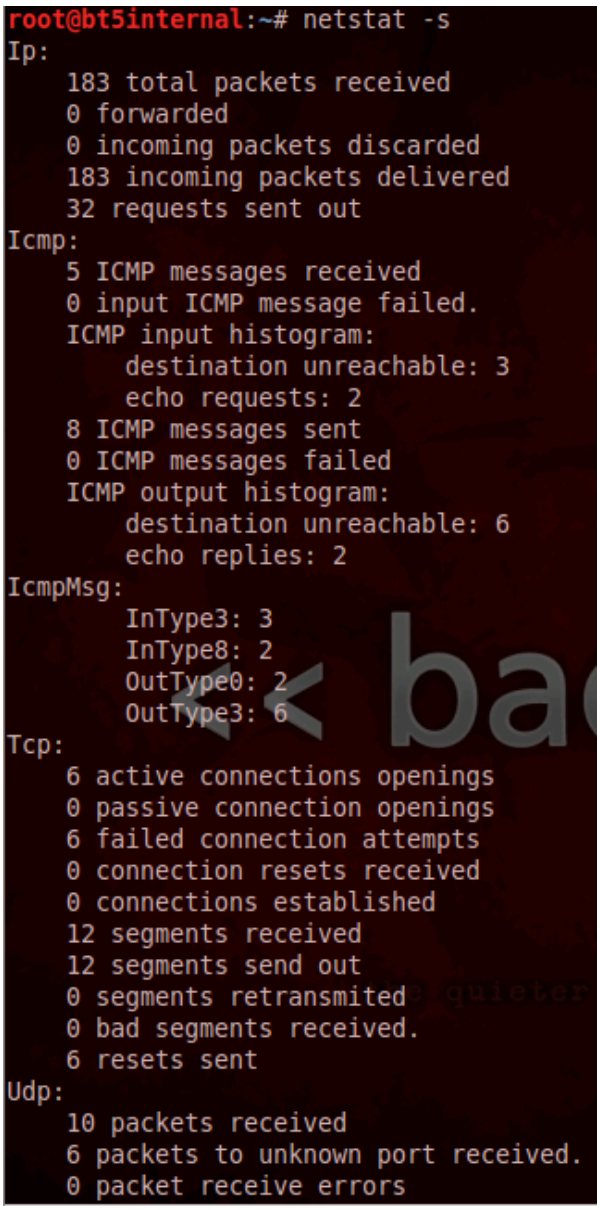
1. *What are some of the command switches that are the same between Windows and Linux?*

12. To display a list of listening sockets in Linux, type **netstat –l** into the terminal window and press **Enter**.



The output of this command lists two connection types – Active Internet Connections and Active UNIX domain sockets. Active Internet Connections are the ports listening for external connections to the machine. Active UNIX domain sockets are connections between applications on the same machine. They also support features not found in TCP/IP (hence the need for their own connection type).

13. To display protocol statistics, type the command **netstat –s** into the terminal window and press **Enter**.



Scrolling up through the output reveals that this command gives statistics for several protocols (such as IP, ICMP, TCP and UDP). Individual protocols cannot be viewed like the Windows output.

14. Keep all windows open to continue with the next task section.

4.2 Conclusion

Netstat, or network statistics, is a command-line tool that displays active TCP connections (both incoming and outgoing) as well as other network statistics. When used with the appropriate command switches, it also displays ports on which the computer is listening, the IP routing table, Ethernet and IP statistics. The utility is available in both Windows and Linux.

4.3 Review Questions

1. *What command is used to display connection information in Windows and Linux?*

2. *What is at least one of the command switches for the above command that is the same between Windows and Linux and what function does it perform?*

3. *If a connection has been made to an available port, what state is that connection in?*

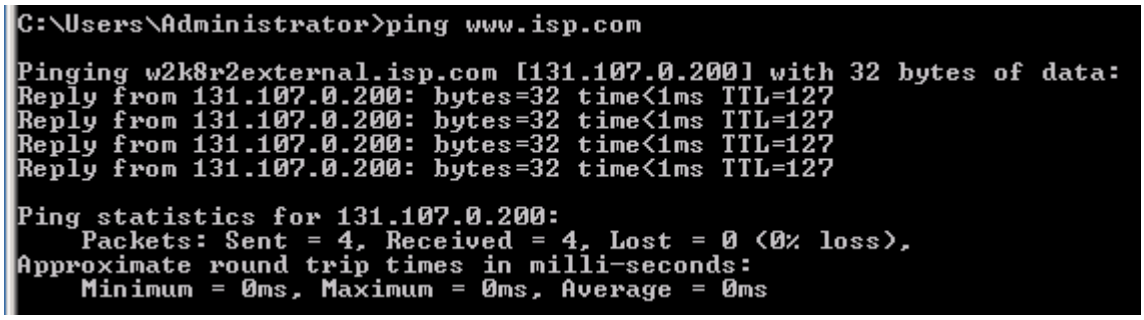
5 Using Commands to Test Network Connectivity

Three commands are especially useful for testing basic network connectivity – Packet Internet Groper (better known as ping), tracert (Windows) and traceroute (Linux). All of these commands use the ICMP protocol. Ping tests end-to-end connectivity, displays latency information and determines whether the IP protocol is correctly configured. Tracert and traceroute takes ping a step further by displaying the same information for every hop (router) between endpoints. These commands can help determine where a failure may have occurred in the network.

5.1 Testing Network Connectivity Using ping, tracecert and traceroute

The basic syntax for the ping command is the same in both Windows and Linux – **ping [host]**. [host] can be an IP address or DNS name. Pinging the DNS name can also help determine if DNS is resolving names correctly.

1. Type the command **ping www.isp.com** into the command prompt on the Windows 2k8 R2 Internal 1 machine and press **Enter**.



Notice the first line of the output. A DNS lookup was performed to resolve the domain name into its IP address. Four ICMP Request messages were then sent to the remote host one by one. Each ICMP Request message was replied to by an ICMP Reply message that was recorded in the output. Each reply also shows the packet size (32 bytes), the latency (<1ms) and the Time-To-Live (TTL) (127). The TTL determines how many hops (routers) the ICMP packet will go through before the packet is discarded. Each hop lowers the TTL by 1. If the TTL reaches 0 for any reason (perhaps the fault of a routing loop), the packet will be discarded. This prevents a packet from looping indefinitely through a network.

2. If the IP address of the remote host is known, the IP address can be used to ping. Type **ping 131.107.0.200** into the command prompt window and press **Enter**.

```
C:\Users\Administrator>ping 131.107.0.200

Pinging 131.107.0.200 with 32 bytes of data:
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127

Ping statistics for 131.107.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Notice the same information is provided, but no DNS lookup was performed.

3. A user can also have ping resolve an IP address into a DNS name in Windows by using the **–a** command switch. Type the command **ping –a 131.107.0.200** into the command prompt window and press **Enter**.

```
C:\Users\Administrator>ping -a 131.107.0.200

Pinging w2k8r2external.isp.com [131.107.0.200] with 32 bytes of data:
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127
Reply from 131.107.0.200: bytes=32 time<1ms TTL=127

Ping statistics for 131.107.0.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

4. If a DNS name cannot be resolved (perhaps from a mistyped name or a server failure), the command will error stating the host could not be found.

```
C:\Users\Administrator>ping www.noname.com
Ping request could not find host www.noname.com. Please check the name and try a
gain.
```

5. If an IP address cannot be reached (or if pings are not allowed to that host), the command will error stating *"Request timed out."*

```
C:\Users\Administrator>ping 131.107.0.201

Pinging 131.107.0.201 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 131.107.0.201:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

6. If an IP address cannot be reached because a route does not exist in the routing table, the router will reply with an error stating *"Destination host unreachable."*

```
C:\Users\Administrator>ping 10.20.30.40

Pinging 10.20.30.40 with 32 bytes of data:
Reply from 192.168.12.1: Destination host unreachable.
Reply from 192.168.12.1: Destination host unreachable.
Reply from 192.168.12.1: Destination host unreachable.
Reply from 192.168.12.1: Destination host unreachable.

Ping statistics for 10.20.30.40:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

7. The tracert command in Windows can also be used with the IP address or DNS name. The main difference is that the output displays statistics about each hop between the endpoints. Type **tracert www.isp.com** into the command prompt window and press **Enter**.

```
C:\Users\Administrator>tracert www.isp.com

Tracing route to w2k8r2external.isp.com [131.107.0.200]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.12.1
  2    <1 ms    <1 ms    <1 ms  w2k8r2external.isp.com [131.107.0.200]

Trace complete.
```

The first hop (192.168.12.1) is the default gateway (the Linux Firewall in the topology diagram); the second hop is the remote machine. Using the command with the IP address of the remote machine produces the exact same output – the DNS lookup is automatically performed.

```
C:\Users\Administrator>tracert 131.107.0.200

Tracing route to w2k8r2external.isp.com [131.107.0.200]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.12.1
  2    <1 ms    <1 ms    <1 ms  w2k8r2external.isp.com [131.107.0.200]

Trace complete.
```

8. The traceroute command in Linux produces the same information just in a different format. Type **traceroute www.isp.com** into the terminal window on the Backtrack 5 Internal machine and press **Enter**.

```
root@bt5internal:~# traceroute www.isp.com
traceroute to www.isp.com (131.107.0.200), 30 hops max, 60 byte packets
 1  * 192.168.12.1 (192.168.12.1)  0.232 ms  0.242 ms
 2  w2k8r2external.isp.com (131.107.0.200)  0.528 ms  0.511 ms  0.515 ms
```

If DNS can resolve the IP address of each hop into a DNS name, it will be displayed in the first column. Using the command with the IP address of the remote machine produces the exact same output.

```
root@bt5internal:~# traceroute 131.107.0.200
traceroute to 131.107.0.200 (131.107.0.200), 30 hops max, 60 byte packets
 1  192.168.12.1 (192.168.12.1)  0.357 ms  0.368 ms  0.392 ms
 2  w2k8r2external.isp.com (131.107.0.200)  0.648 ms  0.659 ms  0.660 ms
```

9. Close all open windows on the Windows machine.

5.2 Conclusion

Three commands are especially useful for testing basic network connectivity – Packet Internet Groper (better known as ping), tracert (Windows) and traceroute (Linux). All of these commands use the ICMP protocol. Ping tests end-to-end connectivity, displays latency information and determines whether the IP protocol is correctly configured. Tracert and traceroute takes ping a step further by displaying the same information for every hop (router) between endpoints. These commands can help determine where a failure may have occurred in the network.

5.3 Review Questions

1. *What command tests only end-to-end connectivity in Windows and Linux?*

2. *What command is used to test every hop between endpoints from a Windows machine?*

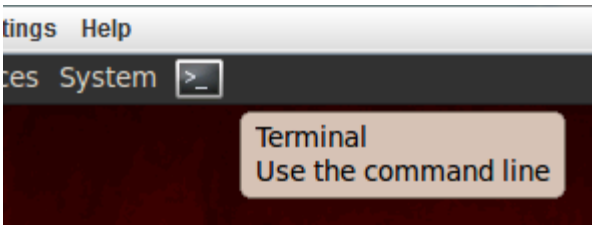
3. *What command is used to test every hop between endpoints from a Linux machine?*

6 Observing the ARP Process

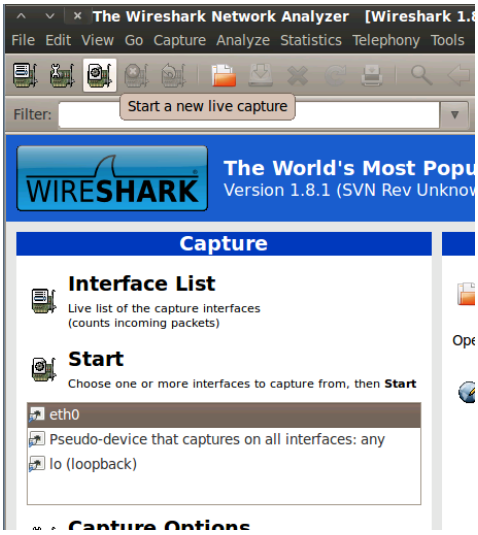
Address Resolution Protocol (ARP) belongs to the TCP/IP suite and is used to determine the OSI layer 2 MAC address associated with the OSI layer 3 IP address. This is a necessary process so that layer 3 packets can be correctly addressed when they are encapsulated as layer 2 frames. The process takes two steps to complete. The first is a request that is broadcast to all nodes on the network asking the "owner" of a particular IP address to respond with its MAC address. The second is a unicast reply with the required information. Wireshark is a protocol analyzer that can be used to observe this process on the network by capturing the packets and displaying them in a user-friendly format. ARP resides at layer 3 of the OSI model.

6.1 Observing the ARP Process Using Wireshark

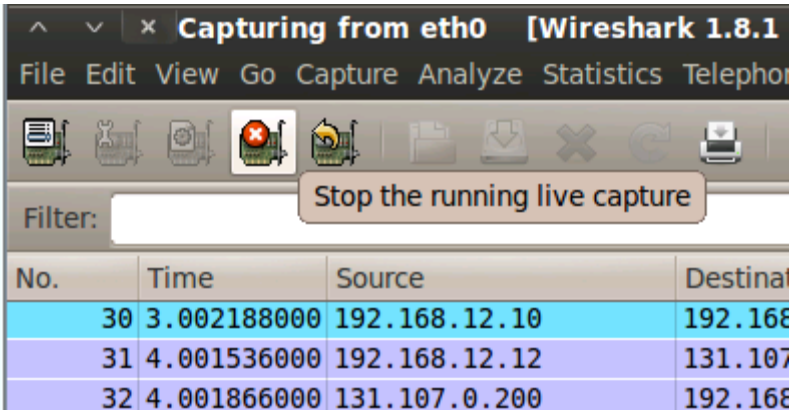
1. On the BackTrack 5 Internal machine, open a second terminal window by clicking the icon to the right of the **System** menu.



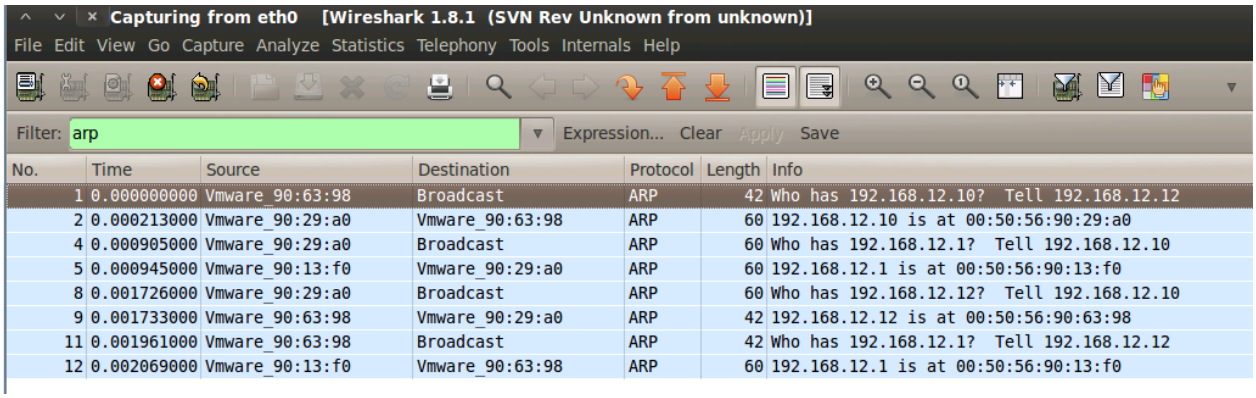
2. Start the Wireshark program by typing **wireshark** into the terminal window and pressing **Enter**. If a warning appears stating that running Wireshark as the root user can be dangerous, click **OK** to discard the window.
3. On the Wireshark homepage, choose the **eth0** interface by clicking on it under the **Start** heading. Next, click the icon to start a new capture.



4. Once the capture is started, network traffic needs to be generated. Click on the second terminal window at the bottom of the screen, type the command **ping www.isp.com –c 5** and press **Enter**. This will issue five pings will be issued to the host www.isp.com and will generate the traffic needed for the capture.
5. Click back on the Wireshark capture window and click the button to stop the capture.



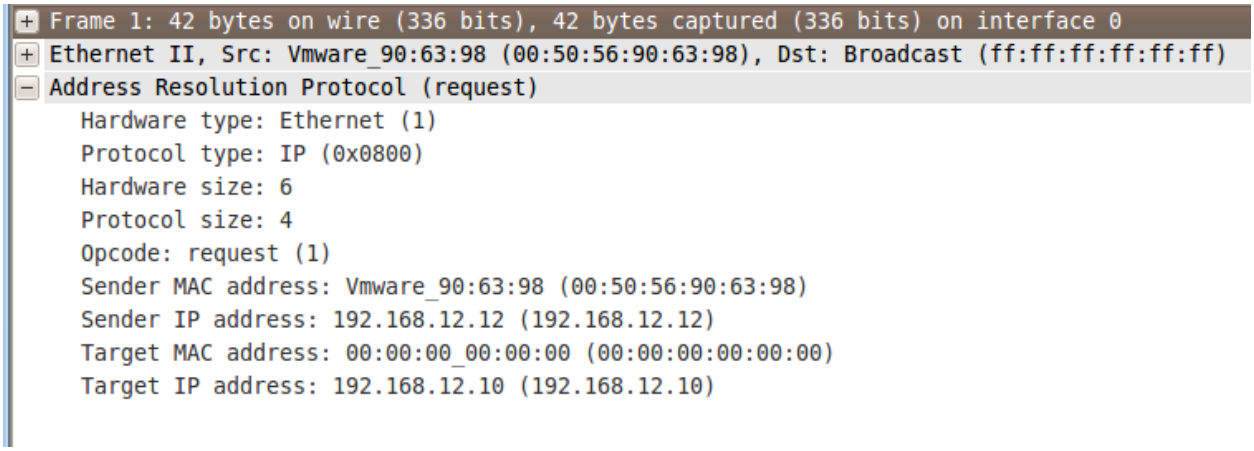
6. The **Filter** dialog box can be used to display only the protocols needed from the capture. In the **Filter** dialog box, type **arp** and press **Enter**. Only packets captured that use the ARP protocol will be displayed in the capture window.



Note that your capture may vary from these figures.

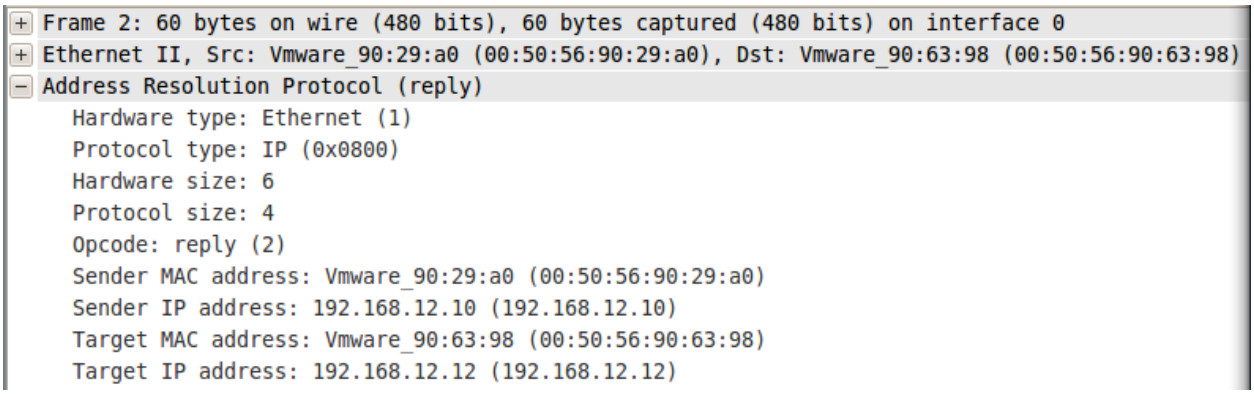
7. This example shows four complete ARP requests – packets 1 & 2, packets 4 & 5, packets 8 & 9 and packets 11 & 12. This can be determined easily by looking at the **Info** column. The first part of the ARP request (packets 1, 4, 8 and 11 in this example) is a broadcast asking whoever has a particular IP address to respond to them with their MAC address. The second part of the ARP request (packets 2, 5, 9 and 12) is the response to this query with their MAC address. Locate one of the ARP requests in the capture and click on it.

8. In the lower capture window (the detail pane), click the **+** next to **Address Resolution Protocol (request)** to expand the details of this section.



Notice the various fields included in the ARP request. The **Opcode** field is set to **1** indicating this is an ARP request. Also, the **Target MAC address** field is currently all zeros since as this is the information being requested.

9. Now locate one of the ARP replies in the capture and click on it. In the detail pane, notice the two differences in the packet. The **Opcode** field is now set to **2** indicating this is an ARP reply. Also, the **Target MAC address** field now contains the MAC address of the destination node.



10. Close the Wireshark program and all open terminal windows by clicking the **x** in the top left corner of each window. Do not save the capture.

6.2 Conclusion

Address Resolution Protocol (ARP) belongs to the TCP/IP suite and is used to determine the OSI layer 2 MAC address associated with the OSI layer 3 IP address. This is a necessary process so that layer 3 packets can be correctly addressed when they are encapsulated as layer 2 frames. The process takes two steps to complete. The first is a request that is broadcast to all nodes on the network asking the "owner" of a particular IP address to respond with its MAC address. The second is a unicast reply with the required information. Wireshark is a protocol analyzer that can be used to observe this process on the network by capturing the packets and displaying them in a user-friendly format. ARP resides at layer 3 of the OSI model.

6.3 Review Questions

1. *The ARP protocol is used to determine the _____ address of a node when the _____ address is known.*

2. *The first half of the ARP process is known as the _____ phase.*

3. *The second half of the ARP process is known as the _____ phase.*