

Fundamental Networking and Security Tools

WHAT YOU WILL LEARN IN THIS CHAPTER:

- Ping
- IPConfig
- Tracert
- NSLookup
- NetStat
- PuTTY

Before heading off to the cybersecurity conference Black Hat in Las Vegas, a friend of mine, Douglas Brush, posted on his LinkedIn page a warning for other InfoSec professionals. He said, “Don’t go to these events to buy curtains for the house when you don’t have the concrete for the foundation poured yet.”

Too many times in the many years I’ve been in information technology (IT), I have seen people forget they need the basics in place before they try to use their shiny new tools. Before you can use any new tools, you must have a foundation to build upon. In IT, these tools are fundamental. They are a must for any computer/InfoSec/analyst to know how to use and when to use them. It’s also rather impressive when a manager who you assumed was nontechnical asks you to ping that asset, run a tracert, and discover the physical and logical addresses of the web server that is down. Sometimes they *do* speak your language!

Ping

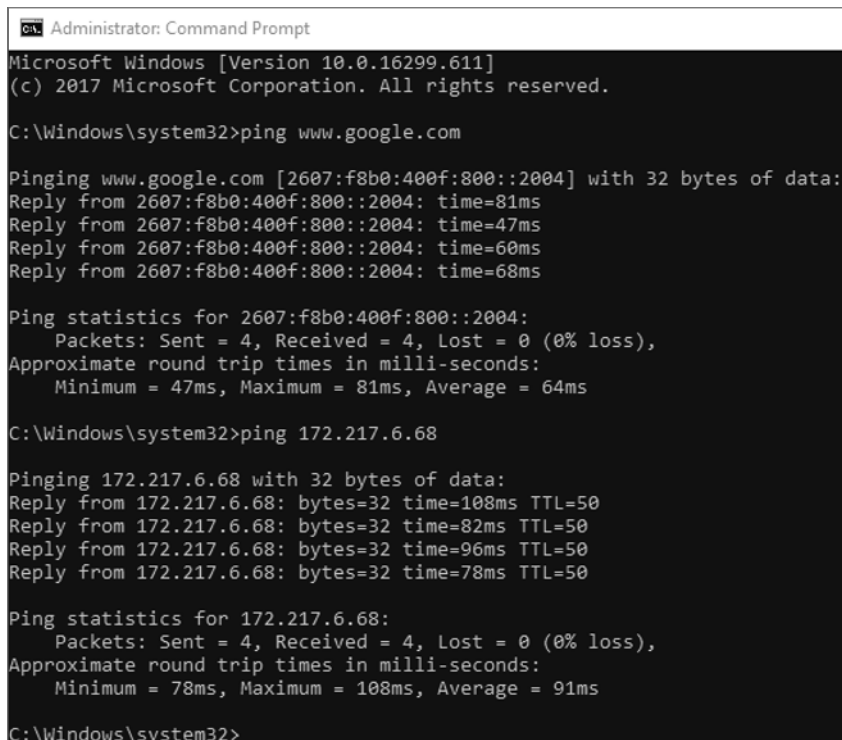
Ping will make you think one of two things. If it makes you think of irons and drivers and 18 holes of beautiful green fairway, then you are definitely CIO/

CEO/CISO material. If it makes you think of submarines or bats, then you're probably geekier like me.

Packet InterNet Groper, or what we affectionately call *ping*, is a networking utility. It is used to test whether a host is “alive” on an Internet Protocol (IP) network. A host is a computer or other device that is connected to a network. It will measure the time it takes for a message sent from one host to reach another and echo back to the original host. Bats are able to use *echo-location*, or bio sonar, to locate and identify objects. We do the same in our networked environments.

Ping will send an Internet Control Message Protocol (ICMP) echo request to the target and wait for a reply. This will report problems, trip time, and packet loss if the asset has a heartbeat. If the asset is not alive, you will get back an ICMP error. The command-line option for ping is easy to use no matter what operating system you are using and comes with multiple options such as the size of the packet, how many requests, and time to live (TTL) in seconds. This field is decremented at each machine where data is processed. The value in this field will be at least as great as the number of gateways it has to hop. Once a connection is made between the two systems, this tool can test the latency or the delay between them.

Figure 1.1 shows a running ping on a Windows operating system sending four echo requests to www.google.com using both IPv4 and IPv6.

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the output of two ping commands. The first command is "ping www.google.com", which shows four successful replies from the IPv6 address 2607:f8b0:400f:800::2004 with response times of 81ms, 47ms, 60ms, and 68ms. The second command is "ping 172.217.6.68", which shows four successful replies from the IPv4 address 172.217.6.68 with response times of 108ms, 82ms, 96ms, and 78ms. Both commands also display ping statistics showing 0% packet loss and average round trip times of 64ms and 91ms respectively.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.16299.611]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>ping www.google.com

Pinging www.google.com [2607:f8b0:400f:800::2004] with 32 bytes of data:
Reply from 2607:f8b0:400f:800::2004: time=81ms
Reply from 2607:f8b0:400f:800::2004: time=47ms
Reply from 2607:f8b0:400f:800::2004: time=60ms
Reply from 2607:f8b0:400f:800::2004: time=68ms

Ping statistics for 2607:f8b0:400f:800::2004:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 47ms, Maximum = 81ms, Average = 64ms

C:\Windows\system32>ping 172.217.6.68

Pinging 172.217.6.68 with 32 bytes of data:
Reply from 172.217.6.68: bytes=32 time=108ms TTL=50
Reply from 172.217.6.68: bytes=32 time=82ms TTL=50
Reply from 172.217.6.68: bytes=32 time=96ms TTL=50
Reply from 172.217.6.68: bytes=32 time=78ms TTL=50

Ping statistics for 172.217.6.68:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 78ms, Maximum = 108ms, Average = 91ms

C:\Windows\system32>
```

Figure 1.1: Running a ping against a URL and IP address

What this figure translates to is that my computer can reach through the network and touch a Google server. The `www.google.com` part of this request is called a *uniform resource locator* (URL). A URL is the address of a page on the World Wide Web (WWW). The numbers you see next to the URL is called an *IP address*. Every device on a network must have a unique IP network address. If you are attempting to echo-locate another host, you could substitute the URL `www.google.com` for an IP address. We will do a deeper dive on IPv4 and IPv6 in Chapter 9, Log Management.

There are more granular `ping` commands. If you type `ping` along with an option or switch, you can troubleshoot issues that might be occurring in your network. Sometimes these issues are naturally occurring problems. Sometimes they could signal some type of attack.

Table 1.1 shows different options you can add to the base command `ping`.

Table 1.1: `ping` command syntax

OPTION	MEANING
<code>/?</code>	Lists command syntax options.
<code>-t</code>	Pings the specified host until stopped with Ctrl+C. <code>ping -t</code> is also known as the <i>ping of death</i> . It can be used as a denial-of-service (DoS) attack to cause a target machine to crash.
<code>-a</code>	Resolves address to hostname if possible.
<code>-n count</code>	How many echo requests to send from 1 to 4.2 billion. (In Windows operating systems, 4 is the default.)
<code>-r count</code>	Records route for count hops (IPv4 only). The maximum is 9, so if you need more than 9, <code>tracert</code> might work better (covered later in the chapter).
<code>-s count</code>	Timestamp for count hops (IPv4 only).
<code>-i TTL</code>	Time to live; maximum is 255.

Did you know that you could ping yourself? Figure 1.2 shows that `127.0.0.1` is a special reserved IP address. It is traditionally called a *loopback address*. When you ping this IP address, you are testing your own system to make sure it is working properly. If this IP doesn't return an appropriate response, you know the problem is with your system, not the network, the Internet service provider (ISP), or your target URL.

```
C:\Windows\system32>ping -a 127.0.0.1

Pinging DESKTOP-0U8N7VK [127.0.0.1] with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figure 1.2: Pinging a lookback address

If you are experiencing network difficulties, this is the first tool to pull out of your toolkit. Go ping yourself and make sure everything is working as it should (see Lab 1.1).

LAB 1.1: PING

1. Open a command prompt or a terminal window.
2. Type `ping -t www.example.com` and then press Enter. (You can use another URL or hostname of your choice.)
3. After a few seconds, hold the Ctrl button and press C (abbreviated as Ctrl+C in subsequent instructions in this book).
4. When the command prompt returns, type `ping -a 127.0.0.1` and press Enter.

What is the name of your host? As you can see in Figure 1.2, mine is DESKTOP-OU8N7VK. A hostname is comprised of alphanumeric characters and possibly a hyphen. There may be times in the future you know an IP address but not the hostname or you know a hostname but not the IP address. For certain troubleshooting steps, you will need to be able to resolve the two on a single machine.

IPConfig

The command `ipconfig` is usually the next tool you will pull out of your toolbox when you're networking a system. A lot of valuable knowledge can be gleaned from this tool.

Internet Protocol is a set of rules that govern how data is sent over the Internet or another network. This routing function essentially creates the Internet we know and love.

IP has the function of taking packets from the source host and delivering them to the proper destination host based solely on the IP addresses in a packet. The datagram that is being sent has two parts: a header and a payload. The header has the information needed to get the information where it should go. The payload is the stuff you want the other host to have.

In Lab 1.2, you'll use the `ipconfig` command.

LAB 1.2: IPCONFIG

1. Open a command prompt or a terminal window.
2. Type `ipconfig` and press Enter if you are on a Windows system. If you are on Linux, try `ifconfig`.
3. Scroll through your adapters and note the ones that are for Ethernet or Wi-Fi or Bluetooth.

With the preceding steps, you can answer the following questions: Which adapters are connected with an IP address? Which ones are disconnected?

4. At the command prompt, type `ipconfig /all` and press Enter.

Now you have a wealth of information to begin your troubleshooting hypothesis. In Figure 1.3, you see the IP addresses and default gateways for each network adapter on the machine.



```
Administrator: Command Prompt
C:\Windows\system32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::cd8d:3b96:32a6:9afa%9
    IPv4 Address. . . . . : 192.168.229.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::d4e9:8916:372a:e132%20
    IPv4 Address. . . . . : 192.168.124.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : lan
    IPv6 Address. . . . . : 2600:1:9507:2759:bc87:7fd4:1989:cb35
    Temporary IPv6 Address. . . . . : 2600:1:9507:2759:88f8:883a:1236:a114
    Link-local IPv6 Address . . . . . : fe80::bc87:7fd4:1989:cb35%5
    IPv4 Address. . . . . : 192.168.128.21
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::895:2734:a5ab:7ea7%5
                                192.168.128.1

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

Figure 1.3: Using `ipconfig /all`

To find your router's private IP address, look for the default gateway. Think of this machine as a literal gateway that you will use to access the Internet or another network. What tool would you use to make sure that the router is alive? Why, ping of course!

THE INTERNET IS DOWN—NOW WHAT?

The Internet is down.

You ping yourself at 127.0.0.1, and everything is fine on your machine. You ping `www.google.com`, and it times out. You do an `ipconfig /all` on your host machine. What can you assume if your `ipconfig /all` command listed the default gateway as being 0.0.0.0? The router!

As an experienced IT person will tell you, the best thing to do is turn any device off and on again—first your host and then the router. Still not working? Expand your hypothesis to another host on your network—can it reach the Internet or the router? Does it pull an IP address from the router? When you are troubleshooting, it is all about the scientific method. Form a hypothesis, test, modify, and form a new hypothesis.

Here are two more acronyms to add to your IT vernacular: DHCP and DNS. DHCP stands for Dynamic Host Configuration Protocol. Let's isolate each word.

Dynamic: Ever-changing, fluid

Host: Asset on a network

Configuration: How the asset is supposed to work

Protocol: Rules that allow two more assets to talk

DHCP is a network management tool. This is the tool that dynamically assigns an IP address to a host on a network that lets it talk to other hosts. Most simply, a router or a gateway can be used to act as a DHCP server. Most residential routers will get their unique public IP address from their ISP. This is who you write the check to each month.

In a large enterprise, DHCP is configured on servers to handle large networks' IP addressing. DHCP decides which machine gets what IP address and for how long. If your machine is using DHCP, did you notice in your `ipconfig /all` command how long your lease was? If you are not leasing, then you are using a static IP address.

Here are two more commands for you to use if you want a new IP address:

`ipconfig /release`: This releases all IPv4 addresses.

`ipconfig /renew`: This retrieves a new IP address, which may take a few moments.

DNS is an acronym for Domain Name System. This is a naming system for all hosts that are connected to the Internet or your private network. As you do what you do on the Internet or in a private network, DNS will remember domain names. It will store this data in something we call a *cache* (pronounced “cash”). This is done to speed up subsequent requests to the same host. Sometimes your DNS cache can get all wonky—sometimes by accident, sometimes by a hacker.

NOTE Cache poisoning—sometimes called *DNS spoofing*—is an attack where a malicious party corrupts the DNS cache or table, causing the nameserver to return an incorrect IP address and network traffic to be diverted.

Here are two more commands to try:

`ipconfig /displaydns`: This may scroll for a while because this is a record of all the domain names and their IP addresses you have visited on a host.

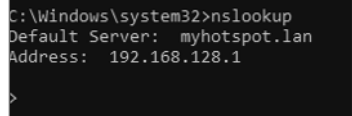
`ipconfig /flushdns`: If you start encountering HTML 404 error codes, you may need to flush your cache clean. This will force your host to query nameservers for the latest and greatest information.

NSLookup

The main use of `nslookup` is to help with any DNS issues you may have. You can use it to find the IP address of a host, find the domain name of an IP address, or find mail servers on a domain. This tool can be used in an interactive and a noninteractive mode. In Lab 1.3, you'll use `nslookup`.

LAB 1.3: NSLOOKUP

1. Open a command prompt or a terminal window.
2. To work in interactive mode, type `nslookup` at the prompt and press Enter. You will get an `nslookup` prompt, as you see in Figure 1.4. To escape the prompt, press Ctrl+C.



```
C:\Windows\system32>nslookup
Default Server: myhotspot.lan
Address: 192.168.128.1
>
```

Figure 1.4: Using `nslookup`

3. To work in noninteractive mode, type `nslookup www.example.com` at the prompt to acquire DNS information for the specific site such as Figure 1.5.

Continues

LAB 1.3 (CONTINUED)

```
C:\Windows\system32>nslookup www.example.com
Server: router.asus.com
Address: 192.168.1.1

Non-authoritative answer:
Name: www.example.com
Addresses: 2606:2800:220:1:248:1893:25c8:1946
          93.184.216.34

C:\Windows\system32>
```

Figure 1.5: Using nslookup on a URL

4. Now try nslookup with one of the IP addresses displayed in your terminal window attributed to www.wiley.com. This will do a reverse lookup for the IP address and resolve to a domain name.
5. To find specific type assets, you can use nslookup -querytype=mx www.example.com. In Figure 1.6, you see the result of using qureytype=mx.

```
C:\Windows\system32>nslookup -querytype=mx www.example.com
Server: router.asus.com
Address: 192.168.1.1

example.com
primary name server = sns.dns.icann.org
responsible mail addr = noc.dns.icann.org
serial = 2018080109
refresh = 7200 (2 hours)
retry = 3600 (1 hour)
expire = 1209600 (14 days)
default TTL = 3600 (1 hour)

C:\Windows\system32>nslookup www.example.com
Server: router.asus.com
Address: 192.168.1.1

Non-authoritative answer:
Name: www.example.com
Addresses: 2606:2800:220:1:248:1893:25c8:1946
          93.184.216.34

C:\Windows\system32>
```

Figure 1.6: Using nslookup with -querytype=mx

Instead of -querytype=mx, you can use any of the following:

HINFO	Specifies a computer's CPU and type of operating system
UNIFO	Specifies the user information
MB	Specifies a mailbox domain name
MG	Specifies an email group member
MX	Specifies the email server

Tracert

So, now you know that all machines that are on a network need to have an IP address. I live in Denver, Colorado, and one of my best friends, Ryan, lives in Albuquerque, New Mexico. When I send him a message, it does not travel from my house through the wires directly to his house. It goes through “hops” (and not the beer kind, unfortunately for him). These hops are the routers between us.

Tracert is a cool diagnostic utility. It will determine the route the message takes from Denver to Albuquerque by using ICMP echo packets sent to the destination. You’ve seen ICMP in action before—with the `ping` command.

ICMP is one of the Internet’s original protocols used by network devices to send operational information or error messages. ICMP is not usually used to send data between computers, with the exception of `ping` and `tracert`. It is used to report errors in the processing of datagrams.

Each router along the path subtracts the packets TTL value by 1 and forwards the packet, giving you the time and the intermediate routers between you and the destination. Tracert will print the trace of the packet’s travels.

Why is this an important part of your toolkit? This is how you find out where a packet gets stopped or blocked on the enterprise network. There may be a router with a configuration issue. Firewalls can be configured to filter packets. Perhaps your website is responding slowly. If packets are being dropped, this will be displayed in the tracert as an asterisk.

This is a good tool when you have many paths that lead to the same destination but several intermediary routers are involved.

One caveat before Lab 1.4: As I mentioned previously, most of my strengths lie in Windows machines. If you are on a Linux or Mac/Unix-type operating system (OS), then you will want to use the tool `traceroute`. The commands `tracert` and `traceroute` are basically the same thing. The difference lies in which OS you are troubleshooting. If you want to get supremely technical, in Linux the command sends a UDP packet. In Windows, it sends an ICMP echo request.

LAB 1.4: TRACERT

1. Open a command prompt or a terminal window.
2. At the command prompt, type `tracert 8.8.8.8` and press Enter.

In Figure 1.7, you can see the hops my machine takes to reach that public Google DNS server. How many hops does yours take?

Continues

LAB 1.4 (CONTINUED)

```

C:\Windows\system32>tracert 8.8.8.8

Tracing route to google-public-dns-a.google.com [8.8.8.8]
over a maximum of 30 hops:

  0  4 ms   2 ms   1 ms  my hotspot.lan [192.168.128.1]
  1  74 ms  76 ms  61 ms  ip-68-29-121-1.pools.spcsdns.net [68.29.121.1]
  2  *      *      *      Request timed out.
  3  90 ms  56 ms  72 ms  66.1.24.242
  4  85 ms  58 ms  61 ms  sl-crs1-che--sprintlink.net [144.223.173.129]
  5  87 ms  50 ms  50 ms  144.232.12.40
  6  77 ms  55 ms  42 ms  209.85.172.62
  7  80 ms  46 ms  60 ms  108.170.254.81
  8  70 ms  50 ms  59 ms  64.233.175.111
  9  80 ms  54 ms  47 ms  google-public-dns-a.google.com [8.8.8.8]

Trace complete.

C:\Windows\system32>

```

Figure 1.7: Using `tracert`, counting hops3. Now try `tracert -d 8.8.4.4`.

This is another public Google DNS server, but now `tracert` will not try to resolve DNS while counting the hops.

4. For fun, try `tracert 127.0.0.1`. Why is it only one hop?

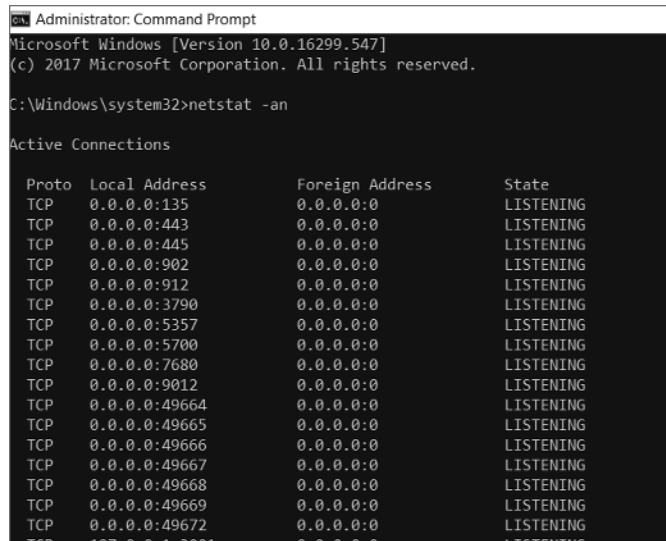
NetStat

Mathematical statistics is the collection, organization, and presentation of data to be used in solving problems. When you analyze statistics, you are going to use probability to fix issues. For example, in a room of 23 people, there is a 50 percent probability that two of those people share the same birthday. In cybersecurity, a birthday attack is a type of cryptographic attack that exploits the math behind the birthday statistic. This attack can be used to find collisions in a hash function. In our world of networking, learning your network statistics can be quite valuable.

NetStat is a network utility tool that displays networking connections (incoming and outgoing), routing tables, and some other details such as protocol statistics. It will help you gauge the amount of network traffic and diagnose slow network speeds. Sounds simple, yes? From a cybersecurity standpoint, how quickly can you tell which ports are open for incoming connections? What ports are currently in use? What is the current state of connections that already exist?

The output from the `netstat` command is used to display the current state of all the connections on the device. This is an important part of configuration and troubleshooting. NetStat also has many parameters to choose from to answer the questions presented in the previous paragraph. One thing to remember about the parameters discussed next is that when you type them into your `cmd` shell, you can literally squish them together. For example, when I am teaching my Metasploit Pro class, we launch a proxy pivot via a Meterpreter shell and scan another network segment. (That might sound like gibberish now, but just

finish the book.) How do you know what is actually transpiring on the compromised system? Using the `netstat` command and the options `-a` for all and `-n` for addresses and ports, you will have a list of all active network conversations this machine is having, as shown in Figure 1.8.



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>netstat -an

Active Connections

Proto Local Address          Foreign Address         State
TCP   0.0.0.0:135             0.0.0.0:0               LISTENING
TCP   0.0.0.0:443             0.0.0.0:0               LISTENING
TCP   0.0.0.0:445             0.0.0.0:0               LISTENING
TCP   0.0.0.0:902             0.0.0.0:0               LISTENING
TCP   0.0.0.0:912             0.0.0.0:0               LISTENING
TCP   0.0.0.0:3790            0.0.0.0:0               LISTENING
TCP   0.0.0.0:5357            0.0.0.0:0               LISTENING
TCP   0.0.0.0:5700            0.0.0.0:0               LISTENING
TCP   0.0.0.0:7680            0.0.0.0:0               LISTENING
TCP   0.0.0.0:9012            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49664           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49665           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49666           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49667           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49668           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49669           0.0.0.0:0               LISTENING
TCP   0.0.0.0:49672           0.0.0.0:0               LISTENING

```

Figure 1.8: NetStat finding active connections

To translate the figure, when running `netstat` on your host, you may see both `0.0.0.0` and `127.0.0.1` in this list. You already know what a loopback address is. A loopback address is accessible only from the machine you’re running `netstat` on. The `0.0.0.0` is basically a “no particular address” placeholder. What you see after the `0.0.0.0` is called a *port*.

One of my favorite explanations of ports is that you have 65,536 windows and doors in your network ranging from 0 to 65,535. Computers start counting at 0. Network admins are constantly yelling, “Shut the windows and close the doors—you’re letting the data out!” Ports can be TCP or UDP. Simply put, TCP means there is a connection made between the host and the destination. UDP doesn’t worry about whether there is a connection made. Both TCP and UDP have 65,535 ports available to them. This was the highest number that could be represented by a 16-bit, or 2-byte, number. You may see this represented mathematically as $2^{16} - 1$.

The Internet Assigned Numbers Authority (IANA) maintains an official assignment of port numbers for specific uses. Sometimes this list becomes antiquated at the same time new technologies are becoming available. Some of the most common ones you might see are the “well-known” ports, which are 0–1023. Looking at the list in the previous figure, you see this machine is listening on port 135. Port 135 is traditionally used for a service called `epmap/loc-srv`. That should tell you, among other things in Figure 1.8, that this is a

Windows host. When a Windows host wants to connect to an RPC service on a remote machine, it checks for port 135.

The next port that is listening is 443. Most IT professionals memorize this port early in their career. Port 443 is Hypertext Transfer Protocol over TLS/SSL—better known as HTTPS. HTTPS is the authentication of a website that is being accessed and protecting the confidentiality of the data being exchanged. Ports from 1023 all the way up to 49151 are “registered” ports. Above that, you have dynamic or private ports.

NetStat is an abbreviation for “network statistics.” If a host is not listening on the correct port for a specific service, then no communication can occur. Take another step in your network path, and these ports may be listening, but this does not mean that a firewall is allowing the traffic to get to the device. To test that hypothesis, you can temporarily disable your host-based firewall causing the networking issue.

Among my favorite `netstat` commands are the statistics options shown in Figure 1.9. In Lab 1.5, you’ll use the `netstat` command.

```

C:\Users\Wadean>netstat -sp IP

IPv4 Statistics

Packets Received           = 128475
Received Header Errors     = 0
Received Address Errors    = 0
Datagrams Forwarded        = 0
Unknown Protocols Received = 0
Received Packets Discarded = 1040
Received Packets Delivered  = 133996
Output Requests            = 50228
Routing Discards           = 0
Discarded Output Packets   = 1195
Output Packet No Route     = 50
Reassembly Required        = 0
Reassembly Successful      = 0
Reassembly Failures        = 0
Datagrams Successfully Fragmented = 0
Datagrams Failing Fragmentation = 0
Fragments Created         = 0

C:\Users\Wadean>netstat -sp UDP

UDP Statistics for IPv4

Datagrams Received = 28100
No Ports          = 1045
Receive Errors    = 10875
Datagrams Sent    = 8789

Active Connections

Proto Local Address      Foreign Address    State
C:\Users\Wadean>
  
```

Figure 1.9: NetStat statistics

LAB 1.5: NETSTAT

1. Open a command prompt or a terminal window.
2. At the command prompt, type `netstat -help`.
3. When the prompt is available, use `netstat -an -p TCP`.
4. Next try `netstat -sp TCP`.

INVESTIGATING THE UNEXPECTED

You're sitting in your office, putting the final touches on a presentation that you're giving in an hour on cybersecurity trends that your specific industry is experiencing to the C-level employees at your company. You're feeling confident with your data. You are hitting the Save button after every major change. You're concentrating on the agenda in your presentation when a balloon in your task pane from your anti-virus software pops up and notifies you that an IP address will be blocked for 600 seconds.

As most end users do, you click the X with no hesitation and continue building your presentation. Then you notice you have mail in your inbox from your firewall. It is an alert notification. You start to worry less about your presentation and start thinking a possible breach is being attempted against your host.

You open a command shell and drop a `netstat -nao`. Not only will this give you the protocol, local/foreign address, and state but also the process identifier (PID) associated with that communication. You can easily get overwhelmed by the data displayed, but check your taskbar. Are there any network-centric applications running? Close your browsers and try `netstat -nao` again.

Did anything change? Are there any foreign addresses or odd port numbers that you've never seen before?

Two ports to be wary of are 4444 and 31337. Port 4444 is the default port that Metasploit will use as a default listening port. Port 31337 spells *leet*.

Leet speak originated in the 1980s when message boards discouraged the discussion of hacking. The purposeful misspelling of words and substitution of letters for numbers was a way to indicate you were knowledgeable about hackers and circumvent the message board police. When we substitute letters with numbers to enhance our passwords, we are using leet speak for good.

If either of these two ports shows up in your NetStat statistics, it's time for a procedure that has been previously agreed upon to kick in. Either pull the network cable on this machine or alert your incident response (IR) team so they can triage the situation and make the best decision on how to stop the attack. My own personal recommendation is that if you have an IR team, use it. If you pull the plug on an attacker, you lose valuable forensic information.

PuTTY

Up until now, all the tools discussed are embedded in your operating systems. This tool will require a little more effort on your part. PuTTY is a free, open-source terminal emulation, serial console, and network file transfer program. Originally written for Windows, it has evolved to be used with other operating systems. PuTTY is an amazingly versatile tool that allows you to gain secure remote access to another computer and is most likely the most highly used SSH client for the Microsoft Windows platform.

I believe that many IT professionals who have been in the industry for a while lose track of where we have been. We keep adding knowledge and experience and expertise to our repertoire and think, “Everyone should know that.” As an educator, I am not allowed to do that. It’s my job to show you how to use all these new shiny things in your toolbox. I can hear some people saying, “You had me until SSH!”

Secure Shell (SSH) is a network protocol for creating an encrypted channel over an unencrypted network. The Internet is *way* unsecured. You don’t want your data out there in the World Wide Web dangling freely for all to see! SSH provides a computer administrator with a safe way to reach a system that is remote using strong authentication and secure, encrypted data transmission. There have been times as an administrator when part of my responsibilities were to manage computers I could not reach out and physically touch—execute commands or move files from one computer to another. SSH is the protocol most hosts support. An SSH server, by default, will listen on TCP port 22.

As I mentioned earlier in this chapter, SSH creates an encrypted channel to communicate over. The first version of SSH debuted in 1995. Brad Pitt was the Sexiest Man Alive, Mel Gibson’s *Braveheart* won Best Picture, and Match.com was new and the only online dating site. A lot. . .and I mean a lot has changed since then. Over the years, several flaws were found in SSH1, and it is no longer used. The current SSH2 was adopted in 2006 and uses a stronger encryption algorithm to improve security. As of yet, there are no known exploitable vulnerabilities in SSH2, although there have been rumors that the National Security Agency (NSA) may be able to decrypt some SSH2 traffic.

In Lab 1.6, you’ll use PuTTY.

LAB 1.6: PuTTY

1. You can download a copy of PuTTY from www.putty.org. There will be a link on the page that takes you to the package file. Make sure you are getting the correct version for the hardware you are running. One size does not fit all.
2. Double-click the file you just downloaded. Follow the instructions until you finish the installation and then open PuTTY by double-clicking the icon that looks like two old computers linked together with a lightning bolt.

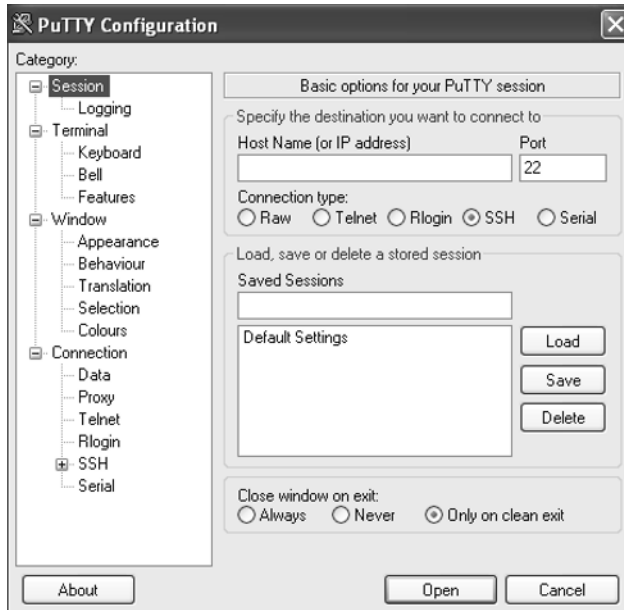


Figure 1.10: PuTTY Configuration window

When the software starts, a PuTTY Configuration window should open, such as what you see in Figure 1.10. The window pane on the left side lists the categories: Session, Terminal, Window, and Connection. The right side of the window will change depending on what category you have selected on the left.

3. In the Session view, enter the domain name or IP address you want to connect to. Port 22 specifies that you will be using SSH. The Connection Type setting lets you choose one of the following options:
 - **Raw:** This is usually used by developers for testing.
 - **Telnet:** Telnet is no longer secure. Passwords are sent in clear text. This is a bad idea.
 - **Rlogin:** This is legacy, which means old (like 1982 old). It uses port 513 and only connects Unix to Unix. Ignore it.
 - **SSH:** This is the protocol most hosts support. An SSH server, by default, will listen on TCP port 22.
 - **Serial:** This is used for controlling some physical machinery or communication devices.
4. After you have supplied the IP or domain address, you should get a terminal window, which will ask for your credentials. If you are able to supply them, you will have a command-line terminal on the machine you just accessed. Some useful commands include the following:

Continues

LAB 1.6 (CONTINUED)

<code>pwd</code>	Present working directory
<code>cd</code>	Change directory
<code>cd ~</code>	Go to the home folder
<code>ls</code>	List files
<code>ls -h</code>	List files with the size
<code>cp</code>	Copy a file
<code>cp -r</code>	Copy a folder with all the files inside
<code>mv</code>	Move a file
<code>mkdir</code>	Make a directory
<code>rm</code>	Delete a file

The session will terminate when you press Ctrl+D.

NOTE The first time you connect another system, you may be prompted to accept the server's SSH key or certificate. It might have some wording like "The server's host key is not cached in the registry." You see an example in Figure 1.11. This is normal. When you click Yes, you are establishing trust between the two hosts.



Figure 1.11: PuTTY security alert

I truly hope that I have given you a foundation to start to build on and that you have added these tools to your cybersecurity toolkit. Some of these tools may have just been a review for you, and some of them might have been new. These tools will help you not only with troubleshooting networks but with securing them as well.