

# Hands-On Ethical Hacking and Network Defense, 3rd Edition

## *Chapter 5* *Port Scanning*

# Objectives

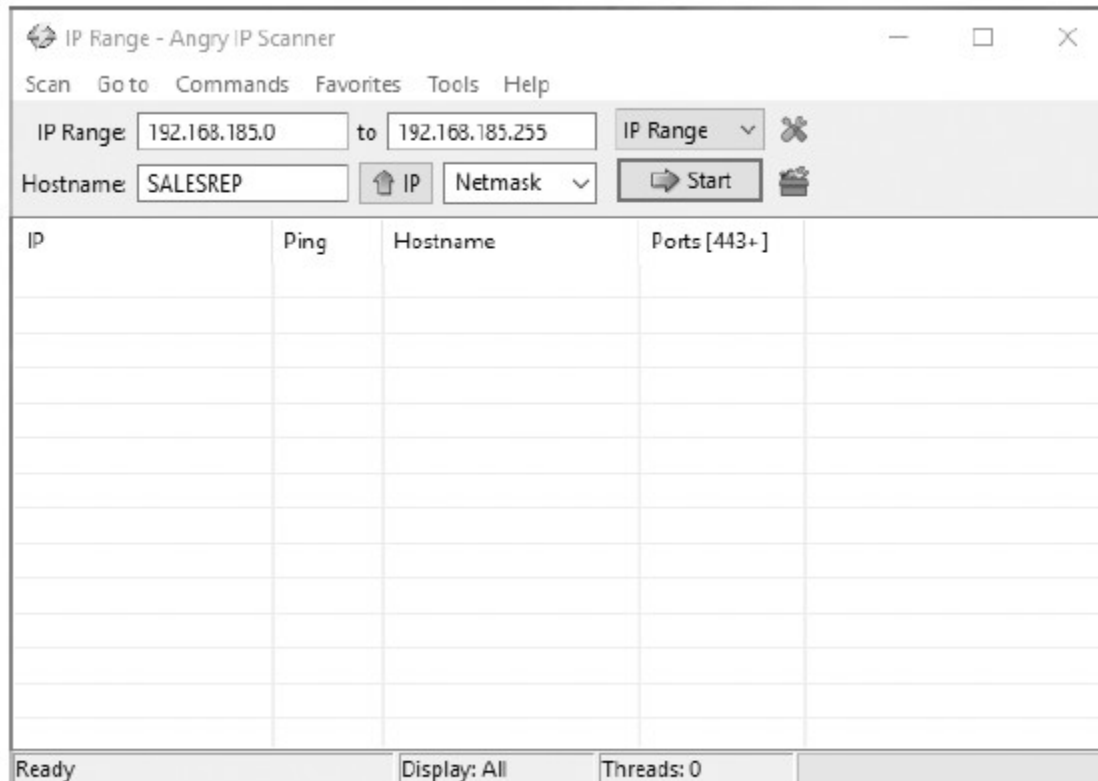
After completing this chapter, you will be able to:

- Describe port scanning and types of port scans
- Describe port-scanning tools
- Explain what ping sweeps are used for
- Explain how shell scripting is used to automate security tasks

# Introduction to Port Scanning

- Port Scanning
  - Method of finding which services are offered by a host
  - Identifies vulnerabilities
- Open services can be used on attacks
  - Identify vulnerable port and launch exploit
- Scans all ports when testing
  - Not just well-known ports

# Introduction to Port Scanning



**Figure 5-1** The Angry IP port scanner interface

Source: GNU General Public License

# Introduction to Port Scanning

- Port scanning programs report:
  - Open ports
    - Allows access to applications and can be vulnerable to attack
  - Closed ports
    - Doesn't allow entry or access to a service
  - Filtered ports
    - Might indicate that a firewall is being used to allow specified traffic into or out of the network

# Types of Port Scans

- SYN scan
  - Stealthy scan
- Connect scan
  - Completes three-way handshake
- NULL scan
  - Packet flags are turned off
- XMAS scan
  - FIN, PSH and URG flags are set

# Types of Port Scans

- ACK scan
  - Used to get past firewall
- FIN scan
  - Closed port responds with an RST packet
- UDP scan
  - Closed port responds with ICMP “Port Unreachable” message

# Using Port-Scanning Tools

- Port-scanning tools
  - Hundreds available
  - Not all are accurate
    - Be familiar with a variety of tools
    - Practice often to gain proficiency
  - Do not use one tool exclusively
- Some tools include:
  - Nmap
  - Nessus and OpenVAS



# Nmap

- Originally written for *Phrack* magazine
  - One of the most popular port scanning tools
  - New features frequently added
- GUI front end
  - Zenmap
  - Makes working with complex options easier
- Standard tool for security professionals
  - Command: `nmap 193.145.85.201`
    - Scans every port on computer with this IP address

# Nmap

```
root@kali: ~  
File Edit View Search Terminal Help  
Nmap 7.01 ( https://nmap.org )  
Usage: nmap [Scan Type(s)] [Options] {target specification}  
TARGET SPECIFICATION:  
  Can pass hostnames, IP addresses, networks, etc.  
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254  
  -iL <inputfilename>: Input from list of hosts/networks  
  -iR <num hosts>: Choose random targets  
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks  
  --excludefile <exclude_file>: Exclude list from file  
HOST DISCOVERY:  
  -sL: List Scan - simply list targets to scan  
  -sn: Ping Scan - disable port scan  
  -Pn: Treat all hosts as online -- skip host discovery  
  -PS/PA/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports  
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes  
  -PO[protocol list]: IP Protocol Ping  
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]  
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers  
  --system-dns: Use OS's DNS resolver  
  --traceroute: Trace hop path to each host  
SCAN TECHNIQUES:  
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans  
  -sU: UDP Scan  
  -sN/sF/sX: TCP Null, FIN, and Xmas scans  
  --scanflags <flags>: Customize TCP scan flags  
  -sI <zombie host[:probeport]>: Idle scan  
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans  
  -sO: IP protocol scan  
  -b <FTP relay host>: FTP bounce scan  
PORT SPECIFICATION AND SCAN ORDER:  
  -p <port ranges>: Only scan specified ports  
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9  
  --exclude-ports <port ranges>: Exclude the specified ports from scanning  
  -F: Fast mode - Scan fewer ports than the default scan  
  -r: Scan ports consecutively - don't randomize  
  --top-ports <number>: Scan <number> most common ports  
  --port-ratio <ratio>: Scan ports more common than <ratio>  
SERVICE/VERSION DETECTION:  
  -sV: Probe open ports to determine service/version info  
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)  
  --version-light: Limit to most likely probes (intensity 2)  
  --version-all: Try every single probe (intensity 9)  
  --version-trace: Show detailed version scan activity (for debugging)  
SCRIPT SCAN:  
:
```

Figure 5-2 The Nmap help screen

Source: © 1996-2015 Insecure.Com LLC

# Nessus and OpenVAS (or Greenbone Security Assistant)

- Nessus
  - First released in 1998
  - No longer under GPL license
    - Still available for download from Tenable Network Security Corporation for noncommercial personal use

# Nessus and OpenVAS (or Greenbone Security Assistant)

- OpenVAS
  - Open-source fork of Nessus in 2005
  - Now branded as Greenbone Security Assistant
  - Capable of updating security check plug-ins
    - Security test programs (scripts)
  - Performs complex queries while client interfaces with server
  - Can also determine what vulnerabilities are associated with services

# Nessus and OpenVAS (or Greenbone Security Assistant)



Figure 5-3 OpenVAS (Greenbone Security Assistant) home screen

Source: GNU General Public License

# Nessus and OpenVAS (or Greenbone Security Assistant)

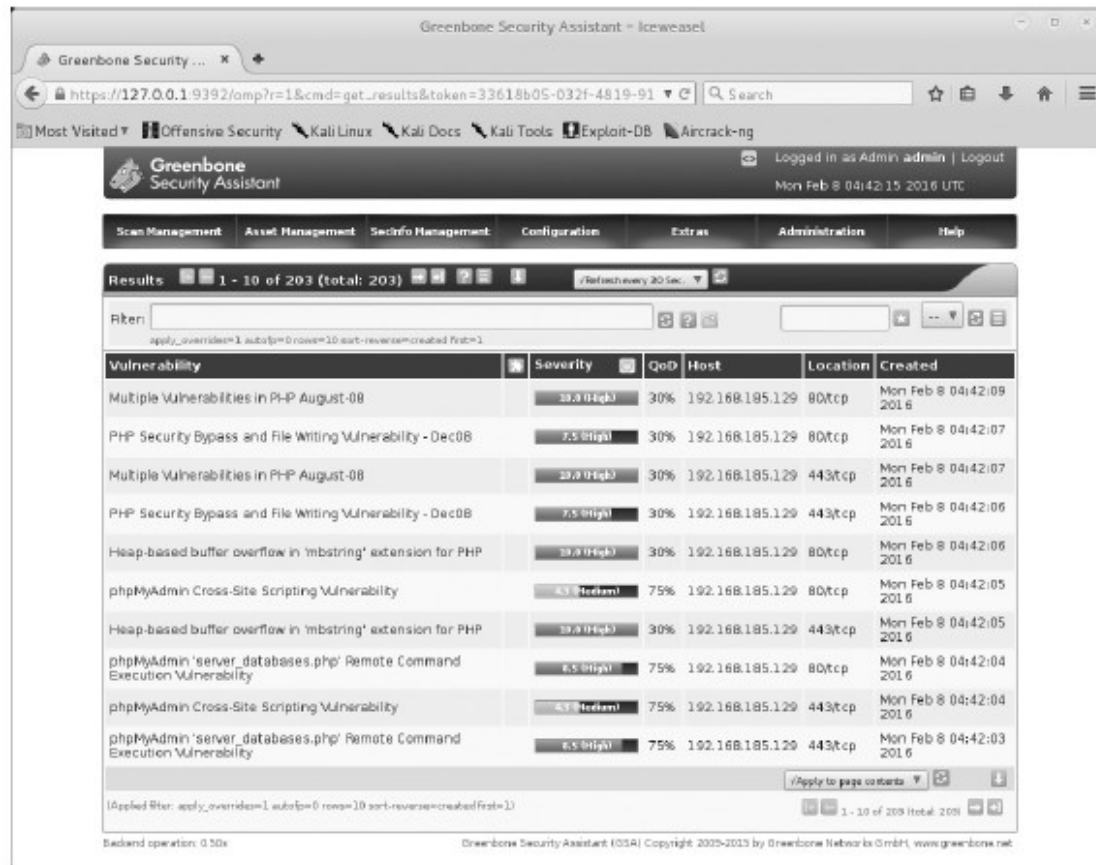


Figure 5-4 OpenVAS discovers a vulnerability

Source: GNU General Public License

# Conducting Ping Sweeps

- Ping sweeps
  - Identify which IP addresses belong to active hosts
    - Ping a range of IP addresses to see what type of response is returned
- Problems
  - Shut down computers cannot respond
  - Networks may be configured to block ICMP Echo Requests
  - Firewalls may filter out ICMP traffic

# Fping

- With the Fping tool you can ping multiple IP addresses simultaneously
  - Included on the Kali Linux DVD
- Accepts a range of IP addresses
  - Entered at a command prompt
  - File containing multiple IP addresses
- Input file
  - Usually created with a shell-scripting language so you don't need to type thousands of IP addresses needed for a ping sweep



# Fping

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# fping -h  
Usage: fping [options] [targets...]  
-a      show targets that are alive  
-A      show targets by address  
-b n    amount of ping data to send, in bytes (default 56)  
-B f    set exponential backoff factor to f  
-c n    count of pings to send to each target (default 1)  
-C n    same as -c, report results in verbose format  
-D      print timestamp before each output line  
-e      show elapsed time on return packets  
-f file  read list of targets from a file [ - means stdin] (only if no -g specified)  
-g      generate target list (only if no -f specified)  
        (specify the start and end IP in the target list, or supply a IP netmask)  
        (ex. fping -g 192.168.1.0 192.168.1.255 or fping -g 192.168.1.0/24)  
-H n    Set the IP TTL value (Time To Live hops)  
-i n    interval between sending ping packets (in millsec) (default 25)  
-I if   bind to a particular interface  
-l      loop sending pings forever  
-m      ping multiple interfaces on target host  
-n      show targets by name (-d is equivalent)  
-O n    set the type of service (tos) flag on the ICMP packets  
-p n    interval between ping packets to one target (in millsec)  
        (in looping and counting modes, default 1000)  
-q      quiet (don't show per-target/per-ping results)  
-Q n    same as -q, but show summary every n seconds  
-r n    number of retries (default 3)  
-s      print final stats  
-S addr set source address  
-t n    individual target initial timeout (in millsec) (default 500)  
-T n    ignored (for compatibility with fping 2.4)  
-u      show targets that are unreachable  
-v      show version  
targets list of targets to check (if no -f specified)  
root@kali:~#
```

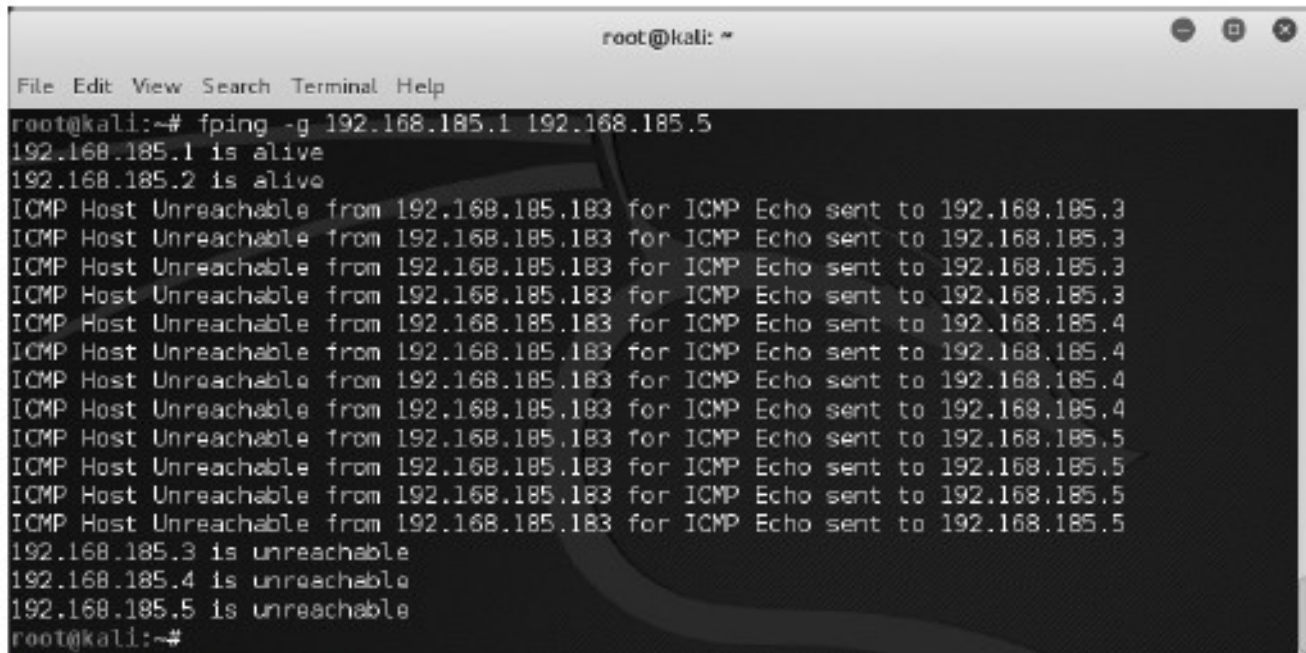
Figure 5-5 Fping parameters

Source: GNU General Public License

# Fping

- To ping sweep a range of IP addresses without using an input file, use the command:
  - `fping -g BeginningIPAddress EndingIPAddress`
  - The `-g` parameter is used when no input file is available
  - Example:
    - `fping -g 192.168.185.1 192.168.185.5` command returns the results shown on Figure 5-6

# Fping

A terminal window titled 'root@kali: ~' with a menu bar containing 'File Edit View Search Terminal Help'. The terminal shows the command 'fping -g 192.168.185.1 192.168.185.5' and its output. The output indicates that 192.168.185.1 and 192.168.185.2 are alive, while 192.168.185.3, 192.168.185.4, and 192.168.185.5 are unreachable. The unreachable status is confirmed by multiple 'ICMP Host Unreachable' messages from 192.168.185.183 to each of the unreachable IP addresses.

```
root@kali:~# fping -g 192.168.185.1 192.168.185.5
192.168.185.1 is alive
192.168.185.2 is alive
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.3
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.3
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.3
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.3
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.4
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.4
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.4
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.4
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.5
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.5
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.5
ICMP Host Unreachable from 192.168.185.183 for ICMP Echo sent to 192.168.185.5
192.168.185.3 is unreachable
192.168.185.4 is unreachable
192.168.185.5 is unreachable
root@kali:~#
```

**Figure 5-6** Results of an Fping command

Source: GNU General Public License

# Hping

- Used to:
  - Perform ping sweeps
  - Bypass filtering devices
    - Allows users to inject modified IP packets
- Powerful tool
  - All security testers must be familiar with tool
  - Supports many parameters

# Hping

```
File Edit View Terminal Go Help
usage: hping host [options]
-h --help      show this help
-v --version   show version
-c --count     packet count
-i --interval  wait (uX for X microseconds, for example -i u1000)
               --fast      alias for -i u10000 (10 packets for second)
-n --numeric   numeric output
-q --quiet     quiet
-I --interface interface name (otherwise default routing interface)
-V --verbose   verbose mode
-D --debug     debugging info
-z --bind      bind ctrl+z to ttl          (default to dst port)
-Z --unbind   unbind ctrl+z

Mode
default mode   TCP
-O --rawip     RAW IP mode
-I --icrp      ICMP mode
-U --udp       UDP mode
-S --scan      SCAN mode.
               Example: hping --scan 1-30,70-90 -S www.target.host
-L --listen    listen mode

IP
-a --spooft    spoof source address
--rand-dest    random destination address mode. see the man.
--rand-source  random source address mode. see the man.
-t --ttl       ttl (default 64)
-N --id        id (default random)
-W --winid     use win* id byte ordering
-r --rel       relativize id field          (to estimate host traffic)
-f --frag      split packets in more frag. (may pass weak acl)
-x --norefrag  set more fragments flag
-y --dontfrag  set dont fragment flag
-g --fragoff   set the fragment offset
-n --mtu       set virtual mtu, implies --frag if packet size > mtu
-o --tos       type of service (default 0x00), try --tos help
-G --rroute    includes RECORD_ROUTE option and display the route buffer
--lsrr        loose source routing and record route
--ssrr        strict source routing and record route
-H --ipproto   set the IP protocol field, only in RAW IP mode
```

Figure 5-7 Hping help, page 1

Source: GNU General Public License

# Hping

```
File Edit View Terminal Go Help
ICMP
-C --icmp-type icmp type (default echo request)
-K --icmp-code icmp code (default 0)
--force-icmp send all icmp types (default send only supported types)
--icmp-gw set gateway address for ICMP redirect (default 0.0.0.0)
--icmp-ts Alias for --icmp --icmp-type 13 (ICMP timestamp)
--icmp-addr Alias for --icmp --icmp-type 17 (ICMP address subnet mask)
--icmp-help display help for others icmp options
UDP/TCP
-s --baseport base source port (default random)
-p --destport [+] [+] <port> destination port (default 0) ctrl+z inc/dec
-k --keep keep still source port
-w --win winsize (default 64)
-o --tcpoff set fake tcp data offset (instead of tcphdr.len / 4)
-Q --seqnum shows only tcp sequence number
-b --badcksum (try to) send packets with a bad IP checksum
many systems will fix the IP checksum sending the packet
so you'll get bad UDP/TCP checksum instead.
-M --setseq set TCP sequence number
-L --setack set TCP ack
-F --fin set FIN flag
-S --syn set SYN flag
-R --rst set RST flag
-P --push set PUSH flag
-A --ack set ACK flag
-U --urg set URG flag
-X --xmas set X unused flag (0x40)
-Y --ynmas set Y unused flag (0x80)
--tcpexitcode use last tcp->th_flags as exit code
--tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime
Common
-d --data data size (default is 0)
-E --file data from file
-e --sign add 'signature'
-j --dump dump packets in hex
-J --print dump printable characters
-B --safe enable 'safe' protocol
-u --end tell you when --file reached EOF and prevent rewind
-T --traceroute traceroute mode (implies --bind and --ttl 1)
:
```

Figure 5-8 Hping help, page 2

Source: GNU General Public License

# Hping

```
File Edit View Terminal Go Help
--icrp-ts Alias for --icrp --icmp-type 13 (ICMP timestamp)
--icrp-addr Alias for --icrp --icmp-type 17 (ICMP address subnet mask)
--icrp-help display help for others icmp options
UDP/TCP
-s --baseport base source port (default random)
-p --destport [+][+]<port> destination port(default 0) ctrl+z inc/dec
-k --keep keep still source port
-w --win winsize (default 64)
-o --tcpoff set fake tcp data offset (instead of tephdrLen / 4)
-Q --seqnum shows only tcp sequence number
-b --badchecksum (try to) send packets with a bad IP checksum
many systems will fix the IP checksum sending the packet
so you'll get bad UDP/TCP checksum instead.
-M --setseq set TCP sequence number
-L --setack set TCP ack
-F --fin set FIN flag
-S --syn set SYN flag
-R --rst set RST flag
-P --push set PUSH flag
-A --ack set ACK flag
-U --urg set URG flag
-X --xmas set X unused flag (0x40)
-Y --ymas set Y unused flag (0x80)
--tcpexitcode use last tcp->th_flags as exit code
--tcp-timestamp enable the TCP timestamp option to guess the HZ/uptime
Common
-d --data data size (default is 0)
-E --file data from file
-e --sign add 'signature'
-j --dump dump packets in hex
-J --print dump printable characters
-B --safe enable 'safe' protocol
-u --end tell you when --file reached EOF and prevent rewind
-T --traceroute traceroute mode (implies --bind and --ttl 1)
--tr-stop Exit when receive the first not ICMP in traceroute mode
--tr-keep-ttl Keep the source TTL fixed, useful to monitor just one hop
--tr-no-rtt Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
--arp-send Send the packet described with APD (see docs/APD.txt)
(END)
```

Figure 5-9 Hping help, page 3

Source: GNU General Public License

# Crafting IP Packets

- Packet components
  - Source IP address
  - Destination IP address
  - Flags
- Helps obtain information about a service
- Tools:
  - Hping
  - Fping



# Understanding Scripting

- Some tools might need to be modified to better suit your needs as a security tester
- Customized scripts
  - Automates tasks
  - Time saving
  - Requires basic programming skills

# Scripting Basics

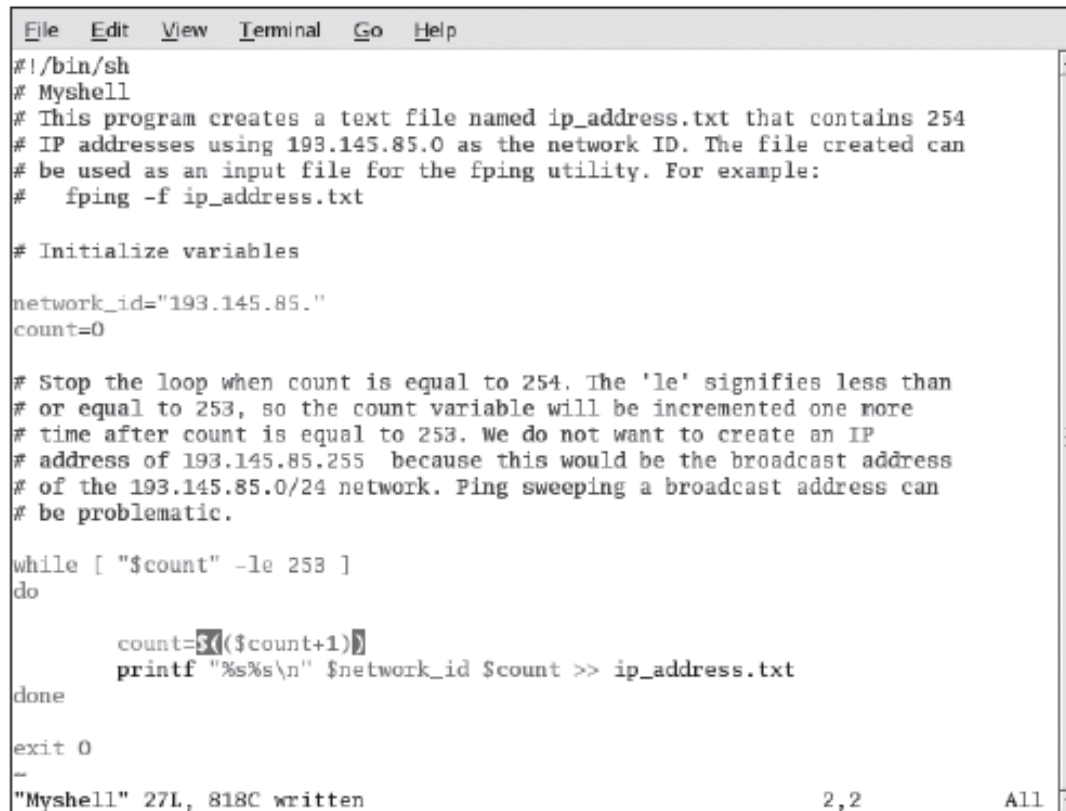
- Similar to DOS batch programming
- A script or batch file
  - Text file
  - Contains multiple commands that would be entered manually at the command prompt
- If you find that you are using repetitive commands to perform the same task
  - That task is a good candidate for scripting
- Practice is the key

# Scripting Basics

<b>vim commands</b>	<b>Description</b>
A	Appends text after the insertion point
I	Inserts text before the insertion point
Delete key	Overwrites the last character when in Insert mode
X	Deletes the current character
Dd	Deletes the current line
Dw	Deletes the current word
P	Replaces the previously deleted text
ZZ	Exits vi and saves all changes
Wq	Writes changes and quits the edit session

**Table 5-1** Summary of vim commands

# Scripting Basics



```
File Edit View Terminal Go Help
#!/bin/sh
# Myshell
# This program creates a text file named ip_address.txt that contains 254
# IP addresses using 193.145.85.0 as the network ID. The file created can
# be used as an input file for the fping utility. For example:
# fping -f ip_address.txt

# Initialize variables
network_id="193.145.85."
count=0

# Stop the loop when count is equal to 254. The 'le' signifies less than
# or equal to 253, so the count variable will be incremented one more
# time after count is equal to 253. We do not want to create an IP
# address of 193.145.85.255 because this would be the broadcast address
# of the 193.145.85.0/24 network. Ping sweeping a broadcast address can
# be problematic.

while [ "$count" -le 253 ]
do
    count=$((count+1))
    printf "%s%s\n" $network_id $count >> ip_address.txt
done

exit 0
~
"Myshell" 27L, 818C written                2,2                All
```

**Figure 5-10** A shell script with comments

Source: GNU General Public License