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

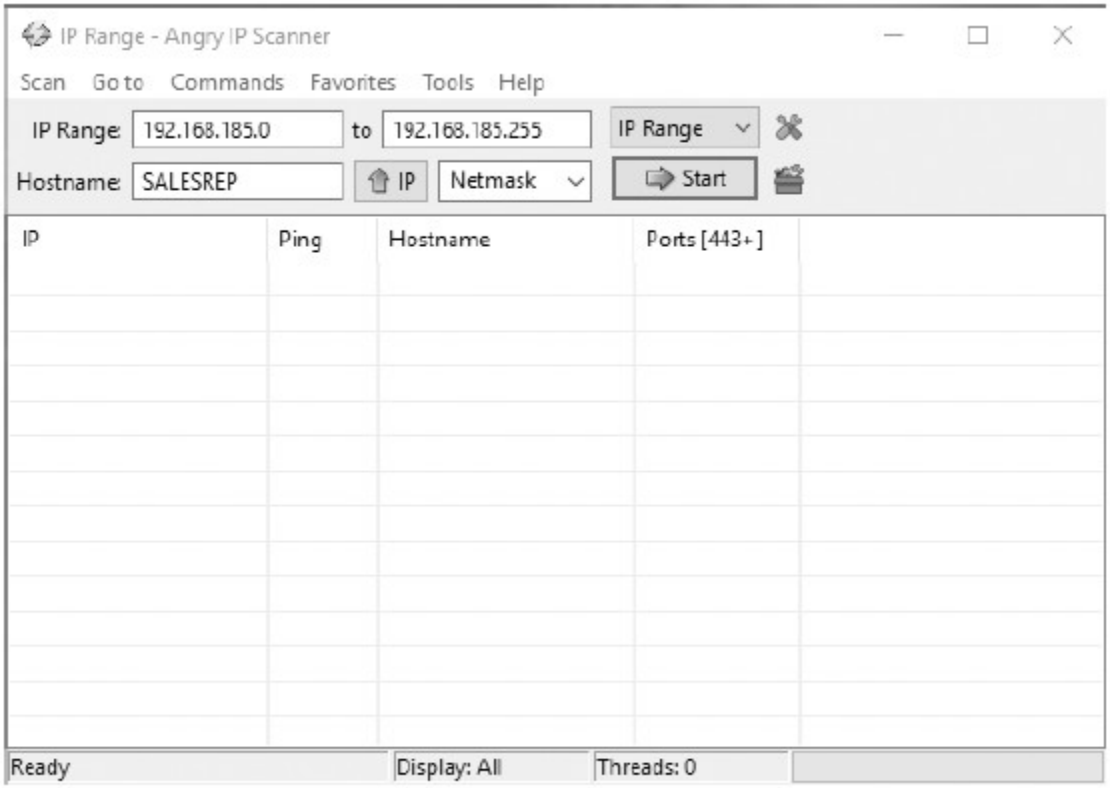

Figure 5-1 The Angry IP port scanner interface

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 allows 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

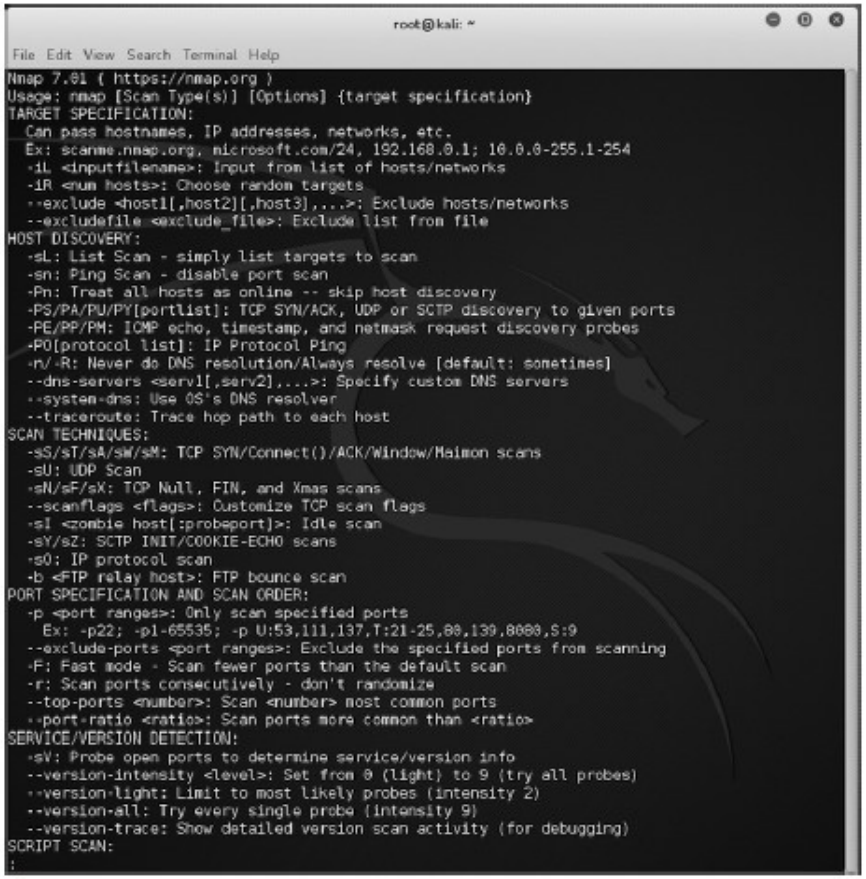

Figure 5-2 The Nmap help screen

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)

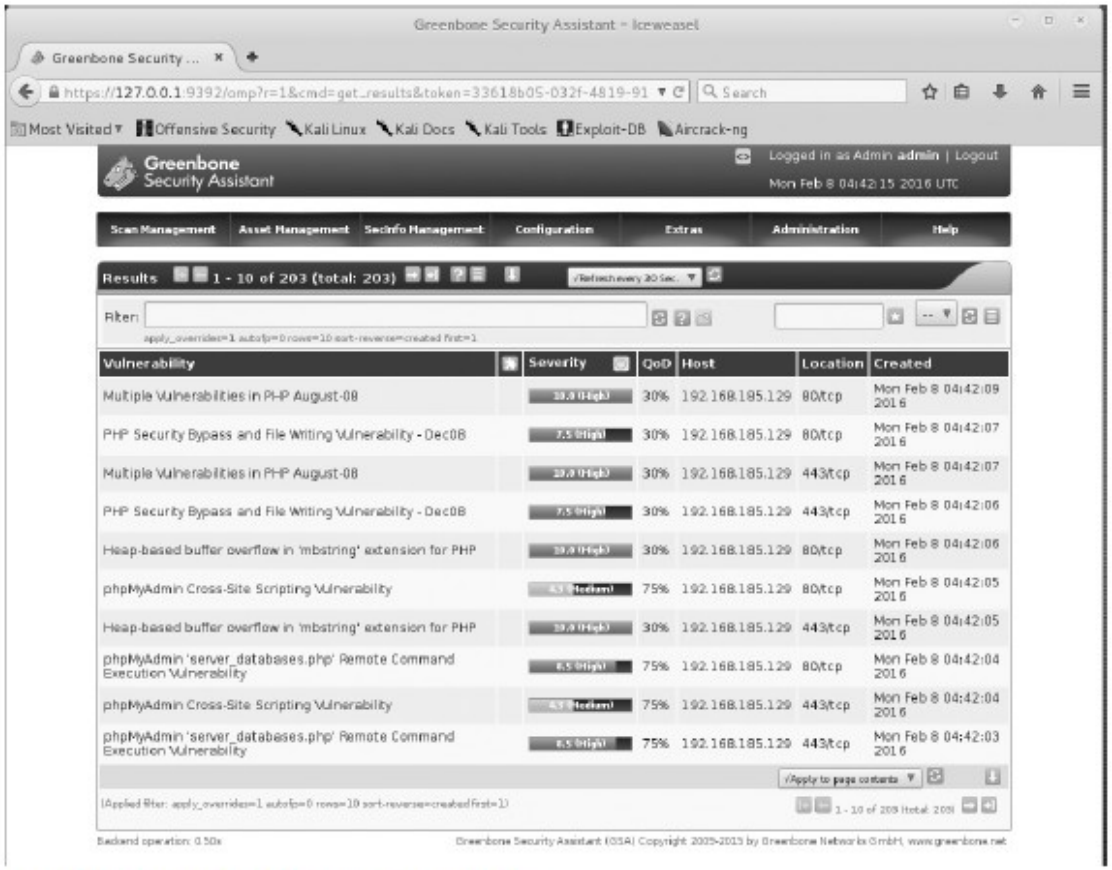

Figure 5-4 OpenVAS discovers a vulnerability

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

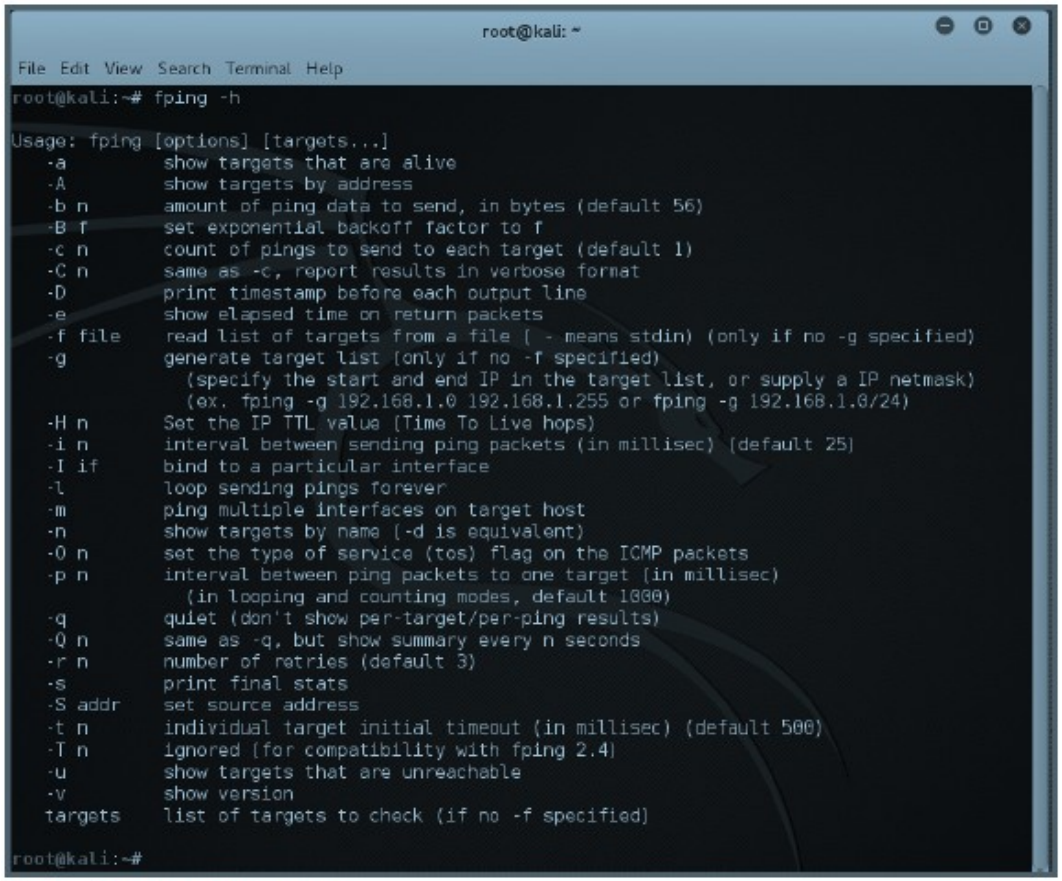

Figure 5-5 Fping parameters

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

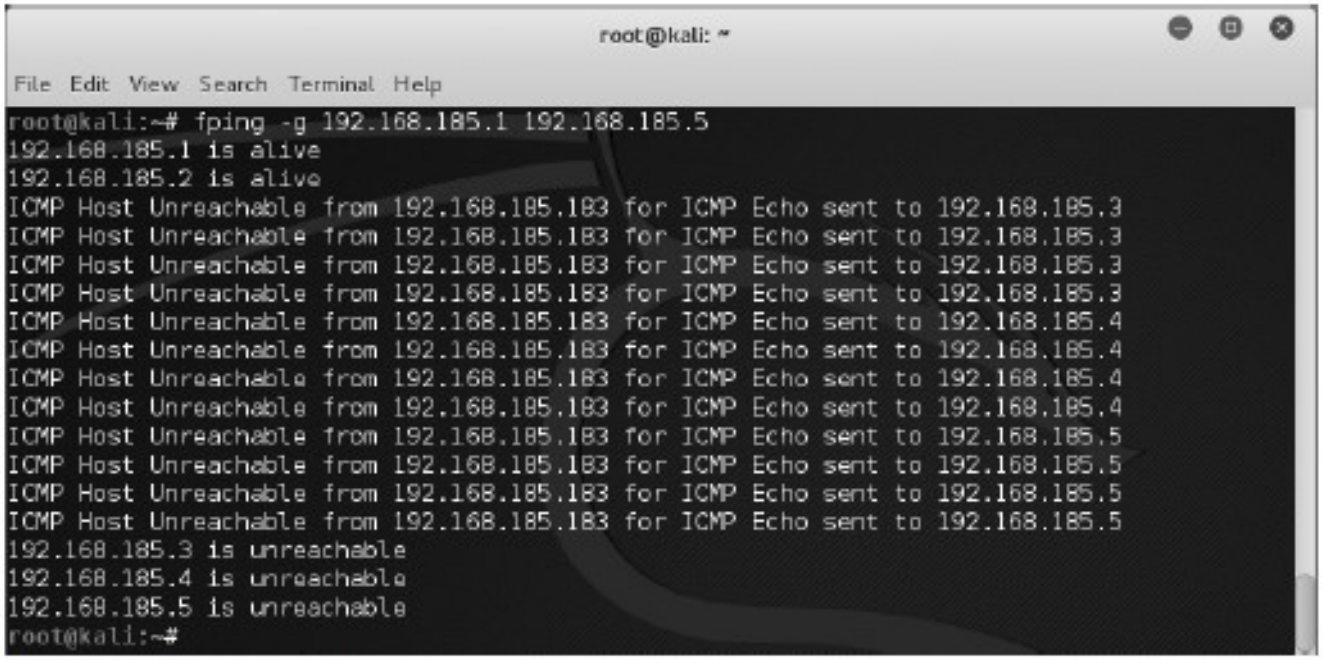

Figure 5-6 Results of an `Fping` command

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
  -0  --rawip       RAW IP mode
  -1  --icmp        ICMP mode
  -2  --udp         UDP mode
  -8  --scan        SCAN mode.
                    Example: hping --scan 1-30,70-90 -S www.target.host
  -9  --listen      listen mode
IP
  -a  --spoof       spoof source address
  --rand-dest       random destionation 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  --morefrag    set more fragments flag
  -y  --dontfrag    set dont fragment flag
  -g  --fragoff     set the fragment offset
  -m  --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  --icmptype   icmp type (default echo request)
  -K  --icmpcode   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 --icmptype 13 (ICMP timestamp)
      --icmp-addr  Alias for --icmp --icmptype 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 tcphdrlen / 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 checksun 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)
:
```

Figure 5-8 Hping help, page 2

Source: GNU General Public License

Hping

```
File  Edit  View  Terminal  Go  Help
      --icmp-ts    Alias for --icmp --icmptype 13 (ICMP timestamp)
      --icmp-addr  Alias for --icmp --icmptype 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 tcphdrlen / 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  --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)
  --apd-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

vim commands	Description
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