

Troubleshooting and Security of Network

Chapter 7: Network Troubleshooting and Security

Study Notes for Engineering Students

Topics Covered
7.1 Understanding the Problem: Troubleshooting, Segmenting, Isolating, Setting Priorities
7.2 Troubleshooting Tools: Hardware, Software, Monitoring and Troubleshooting Tools
7.3 Internal Security: Account Security, File and Directory Permissions, Practices and User Education
7.4 External Threats: Front Door, Back Door, Denial of Service, Viruses, Worms and Malicious Code
Also Included: diagrams, tables, quick-revision sheets, 15 MCQs with answer key and 10 broad questions

Contents

Chapter Overview	3
Learning Objectives	3
7.1 Understanding the Problem	4
7.1.1 What Is Troubleshooting?	4
7.1.2 A Troubleshooting Methodology	4
7.1.3 Segmenting the Problem	5
7.1.4 Isolating the Problem	6
7.1.5 Setting Priorities	6
7.2 Troubleshooting Tools	8
7.2.1 Hardware Tools	8
7.2.2 Software Tools	8
7.2.3 Monitoring and Troubleshooting Tools	9
7.3 Internal Security	12
7.3.1 Why Internal Security Matters	12
7.3.2 Account Security	12
7.3.3 File and Directory Permissions	13
7.3.4 Practices and User Education	14
7.4 External Threats	16
7.4.1 Who Attacks and How	16
7.4.2 Front Door Threats	16
7.4.3 Back Door Threats	16
7.4.4 Denial of Service Threats	17
7.4.5 Viruses, Worms and Other Malicious Code	18
7.4.6 Defensive Technologies	19
7.4.7 Incident Response	20
Quick Revision Tables	21
Practice Questions	22
Part A: Multiple Choice Questions (15)	22
Answer Key for MCQs	24
Part B: Broad Questions (10)	24

Chapter Overview

A computer network is only useful while it works and while it can be trusted. A **network administrator** therefore has two connected duties. The first is **troubleshooting**: finding and correcting faults quickly and methodically so that users can keep working. The second is **security**: protecting the network, its devices and its data from misuse, whether the danger comes from careless users inside the organisation or from attackers outside it.

This chapter first teaches a structured way of thinking about faults (understand, segment, isolate and prioritise) and the hardware and software tools that support it. It then studies network security in two parts: **internal security**, which concerns accounts, permissions and the behaviour of users, and **external threats**, which include intruders, denial-of-service attacks and malicious programs.

Table 7.1: Summary of the chapter

Section	Core question	Key ideas
7.1	How do I approach a network problem logically?	Methodology, OSI layers, isolation, priorities
7.2	Which tools help me find the fault?	Cable testers, TDR, ping, traceroute, Wireshark, SNMP
7.3	How do I protect the network from inside?	Passwords, least privilege, permissions, training
7.4	How do I defend against outside attacks?	Intrusions, DoS, malware, firewalls, IDS

Learning Objectives

After studying this chapter the student should be able to:

- Apply a step-by-step troubleshooting method, divide a problem into layers or segments and isolate the failing component.
- Rank problems by impact and urgency and decide what should be fixed first.
- Select and use suitable hardware and software tools to test, monitor and analyse a network.
- Explain account security, permission settings and the importance of user education.
- Describe front door, back door, denial-of-service and malware threats and the matching countermeasures.

7.1 Understanding the Problem

7.1.1 What Is Troubleshooting?

Troubleshooting is the systematic process of detecting that something is wrong, finding the cause, correcting it and confirming that normal operation has returned. It is not guessing or randomly replacing parts. A skilled troubleshooter works like a doctor: listen to the symptoms, examine, form a diagnosis, treat, and then check that the patient has recovered.

Network faults seldom announce their cause. A user simply reports that "the Internet is down", yet the cause may be a loose cable, a wrong IP address, a failed DNS server, a blocked port, a broken router or even a virus. Good troubleshooting therefore begins with understanding what is actually happening.

Table 7.2: Common symptoms and their possible causes

Symptom	Possible causes
No connectivity at all	Unplugged or faulty cable, disabled NIC, failed switch port, wrong IP configuration.
Can reach local hosts but not the Internet	Wrong default gateway, router or ISP failure, firewall rule, DNS failure.
Can ping IP addresses but not names	DNS server unreachable or wrong DNS entry.
Network is slow	Congestion, duplex mismatch, broadcast storm, faulty NIC, malware or heavy downloads.
Problem occurs intermittently	Loose connector, interference, overheating device, DHCP address conflict.
Cannot access a particular server or share	Permissions, server service stopped, port blocked, wrong credentials.

7.1.2 A Troubleshooting Methodology

Following a fixed series of steps prevents wasted effort and makes the results repeatable. The widely used method has seven steps.

1. **Identify the problem.** Collect information from users, error messages and logs. Ask what changed recently and whether the problem can be reproduced.
2. **Gather information** about the affected hosts, devices and services (IP settings, link status, logs).
3. **Establish a theory** of the probable cause, beginning with the simplest and most likely.
4. **Test the theory.** If it is confirmed, continue; if not, form a new theory or ask for help.
5. **Plan and implement** the solution, taking care not to disturb other users.
6. **Verify full functionality** and, if possible, apply preventive measures.
7. **Document** the symptoms, cause, solution and any changes made.

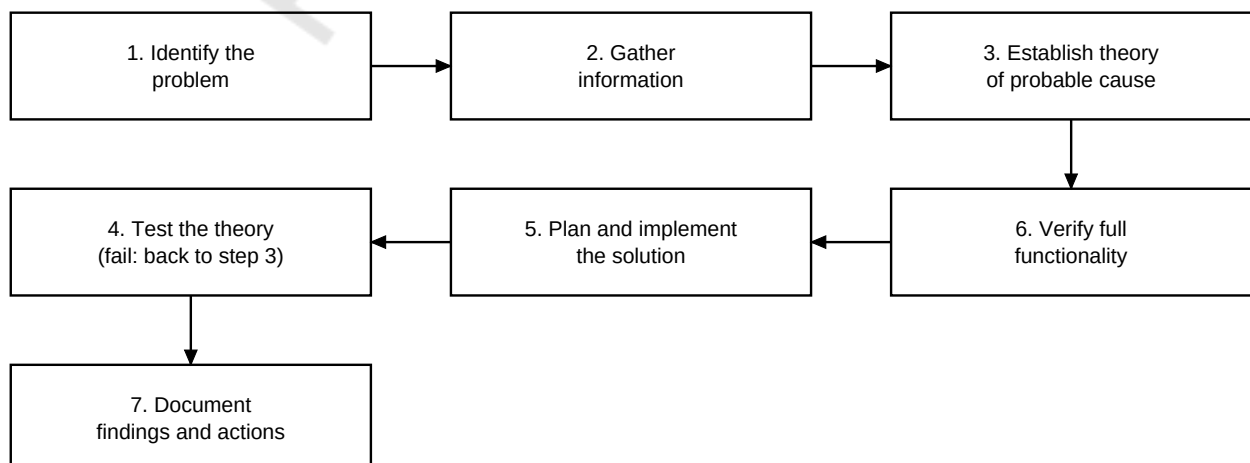


Fig. 7.1: Seven steps of network troubleshooting

Common troubleshooting approaches

Table 7.3: Troubleshooting approaches

Approach	How it works	Best used when
Top-down	Start at the Application layer and work down to Physical.	Software or application problems are suspected.
Bottom-up	Start at the Physical layer and work up.	Hardware or cabling problems are suspected.
Divide and conquer	Begin at a middle layer (for example, ping) and move up or down based on the result.	Experience suggests a likely layer.
Follow the path	Trace the route of the traffic from source to destination.	Faults along a path with several devices.
Substitution (swap)	Replace a suspected part with a known-good one.	Cable, NIC, port or module is suspected.
Comparison	Compare the faulty system with a similar working one.	Configuration differences are suspected.

Rule: Change **one thing at a time** and record it. If several settings are changed together, it is impossible to know which change solved (or worsened) the problem.

7.1.3 Segmenting the Problem

Large networks contain many components, so the problem has to be **divided into smaller parts** that can be checked independently. This is called segmenting. A network can be segmented in three useful ways.

- **By layer:** use the OSI or TCP/IP model and check each layer in turn (physical connections, addressing, transport ports, applications).
- **By location:** separate the workstation, the local segment (LAN), the router or gateway, the WAN link and the remote server.
- **By type:** decide whether the cause lies in hardware, software, configuration or the user.

Useful questions at this stage are: Who is affected, one user or many? Which service is affected? When did it start and did anything change? Does it affect only one segment, floor or building? Can the same task be done from another computer?

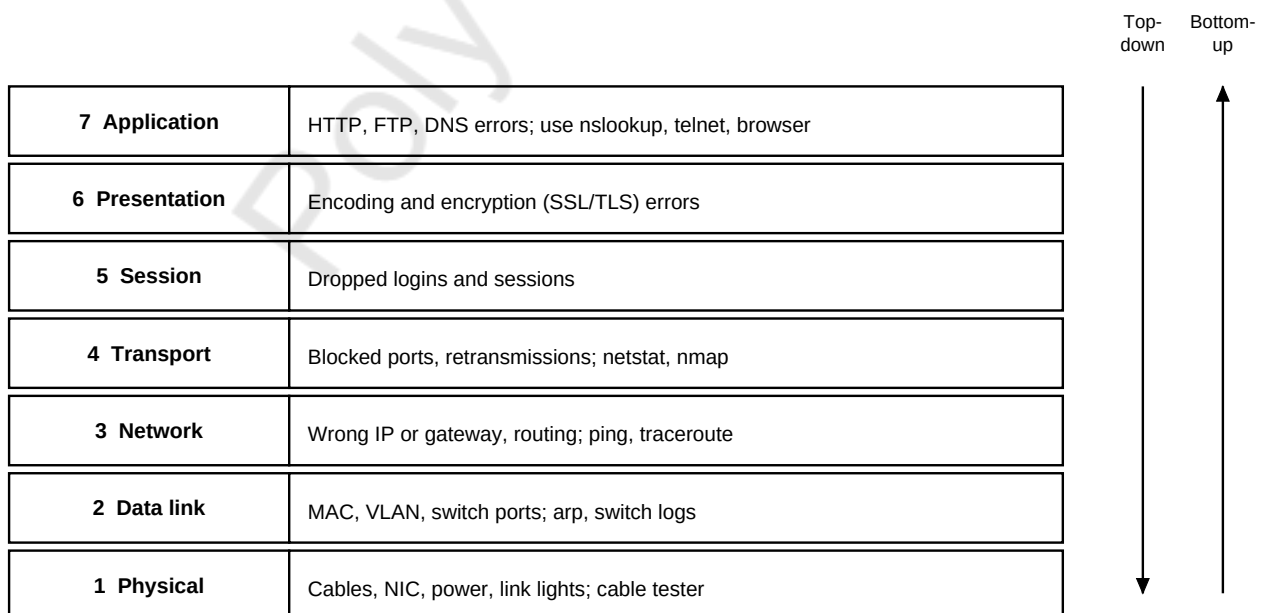
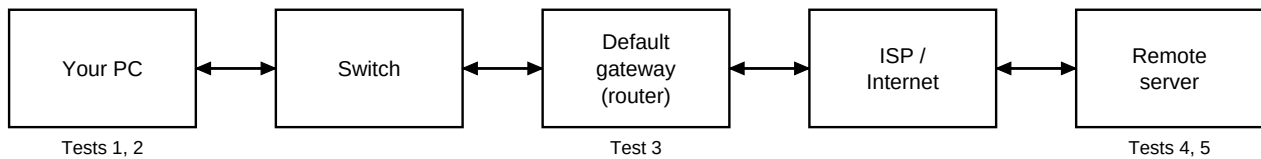


Fig. 7.2: Segmenting a problem by OSI layer, with typical faults and tools

7.1.4 Isolating the Problem

After segmenting, the troubleshooter must **isolate** the single component that is responsible. The idea is to test from one end to the other and find the point at which the result changes from "working" to "not working". The usual technique is a ladder of ping tests that begins at the local machine and moves outward step by step.



Test 1: ping 127.0.0.1 checks the local TCP/IP software
Test 2: ping own IP address checks the network card configuration
Test 3: ping the default gateway checks the local segment and cabling
Test 4: ping a remote IP address checks routing beyond the gateway
Test 5: ping a host name checks DNS name resolution

Fig. 7.3: Isolating a connectivity problem using a ladder of ping tests

If test 1 fails, the TCP/IP software on the machine is faulty. If test 2 fails, the NIC or its IP configuration is wrong. If test 3 fails, the cable, switch port or local segment is at fault. If test 4 fails, the problem lies in routing or the WAN link. If test 5 fails while test 4 succeeds, DNS is the culprit. The first test that fails identifies the location of the fault.

Table 7.4: Interpreting ping results

Message / result	Meaning	Next action
Reply received with times	Path and remote host are working.	Fault is probably higher up (service or application).
Request timed out	No reply; host down, packets dropped or ICMP blocked.	Check the next device closer to the source; check firewalls.
Destination host unreachable	Router or local system has no route to the host.	Check gateway, routing table and IP/mask.
Unknown host / could not find host	Name cannot be resolved.	Check DNS settings and the host name.
TTL expired in transit	Packet looped or travelled too many hops.	Look for routing loops with <code>tracert</code> .

7.1.5 Setting Priorities

An administrator is rarely given a single fault at a time. Several users may report different problems simultaneously, so the faults must be **prioritised** according to their effect on the organisation. Treating all problems as equal wastes time on minor issues while critical systems remain down. The factors to consider are:

- **Number of users affected:** a core switch failure affects hundreds; a single desktop affects one.
- **Business importance:** payment servers, e-mail, and production systems matter more than convenience services.
- **Security risk:** a suspected intrusion normally outranks a slow printer.
- **Availability of a workaround:** a problem with a temporary alternative is less urgent.
- **Service Level Agreements (SLAs):** contracts may fix the maximum response time.

Table 7.5: Example priority scheme (response times vary by organisation)

Priority	Description	Typical example	Typical response
1 – Critical	Entire network or a vital service is down; security breach in progress.	Core router failure, ransomware outbreak	Immediate; work continues until resolved
2 – High	Large group or important system seriously affected.	File server down for one department	Within an hour or two
3 – Medium	Few users affected or a workaround exists.	One workstation cannot print	Within the same working day
4 – Low	Minor inconvenience or a request.	Request for new software	Scheduled or planned

Good practice also includes keeping users informed, using a **ticketing system** to record each request, escalating the problem to a senior engineer when it cannot be solved in time, and performing a root-cause analysis afterwards so that the same fault does not return.

7.2 Troubleshooting Tools

Tools extend the senses of the administrator. They fall into three groups: **hardware tools** for the physical layer, **software tools** that interrogate the network from a computer, and **monitoring tools** that watch the network continuously and warn about problems before users complain.

7.2.1 Hardware Tools

Table 7.6: Hardware tools for network troubleshooting

Tool	Purpose	Typical use
Cable tester	Checks continuity, shorts, opens and wiring order (pin-out) of copper cables.	Verifying a newly made patch cable.
Cable certifier	Measures performance (attenuation, crosstalk, length) against a standard such as Cat 5e/6.	Certifying an installed cable run.
TDR (Time Domain Reflectometer)	Sends a pulse and measures reflections to find the distance to a break or fault in copper cable.	Locating a cut in a long cable.
OTDR / optical power meter	Finds faults and measures loss in fibre-optic links.	Fibre backbone testing.
Tone generator and probe	Generator sends a tone on a wire; the probe detects it to trace the cable.	Finding which wall jack leads to which patch-panel port.
Crimper, punch-down tool, wire stripper	Attach connectors (RJ-45) and terminate cables in patch panels.	Building and repairing cables.
Loopback plug	Returns transmitted signal to the sender so a port or NIC can be tested.	Testing a faulty NIC or serial port.
Multimeter	Measures voltage, current and resistance.	Checking power supplies and cable continuity.
Wireless / spectrum analyser	Shows signal strength and interference on Wi-Fi channels.	Locating dead zones and interference.
Hardware protocol analyser / network tap	Captures traffic passing through a link without disturbing it.	Analysing traffic on a switch uplink.

Tip: The simplest hardware checks solve many faults: look at the **link LEDs** on the NIC and switch, confirm the cable is firmly seated, try another port, and swap with a known-good cable before using advanced equipment.

7.2.2 Software Tools

Operating systems include a number of command-line utilities that test connectivity, display configuration and show active connections. The names differ slightly between Windows and Linux.

Table 7.7: Common software tools

Purpose	Windows	Linux / UNIX
Test reachability (ICMP echo)	ping	ping
Show IP configuration	ipconfig /all	ifconfig, ip addr
Trace route to a host	tracert	tracert, mtr
DNS lookup	nslookup	nslookup, dig, host
Active connections and ports	netstat -an	netstat -tuln, ss
ARP cache (IP to MAC)	arp -a	arp -n
Routing table	route print	route -n, ip route
Test one TCP port	telnet host port	telnet, nc host port
Scan ports on hosts	nmap (third party)	nmap
Capture and decode packets	Wireshark	tcpdump, Wireshark

```

$ ping -c 4 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.45 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.41 ms
...
4 packets transmitted, 4 received, 0% packet loss

$ traceroute www.example.com      list of routers (hops) to the target
$ nslookup www.example.com        ask the DNS server for the address
$ netstat -tuln                   which ports are listening
$ tcpdump -i eth0 port 80         capture web traffic on eth0

```

Packet analysers (sniffers) such as **Wireshark** and **tcpdump** capture frames from the network and decode them layer by layer. They reveal what is really being sent: failed TCP handshakes, retransmissions, wrong addresses, or suspicious traffic. They must be used responsibly because captured data may include passwords and private content.

7.2.3 Monitoring and Troubleshooting Tools

Instead of waiting for a failure, administrators use **network monitoring systems** that poll devices and record performance. A recorded history lets the administrator build a **baseline**, meaning a picture of what is normal for the network. A problem is then recognised as a departure from the baseline, such as traffic doubling or response time rising sharply.

SNMP: Simple Network Management Protocol

SNMP is the standard protocol for monitoring and managing network devices. A **Network Management Station (NMS)** communicates with an **agent** program running on every managed device (router, switch, server, printer). Each agent keeps its data in a **MIB** (Management Information Base), a tree of numbered variables such as interface traffic, errors and CPU load.

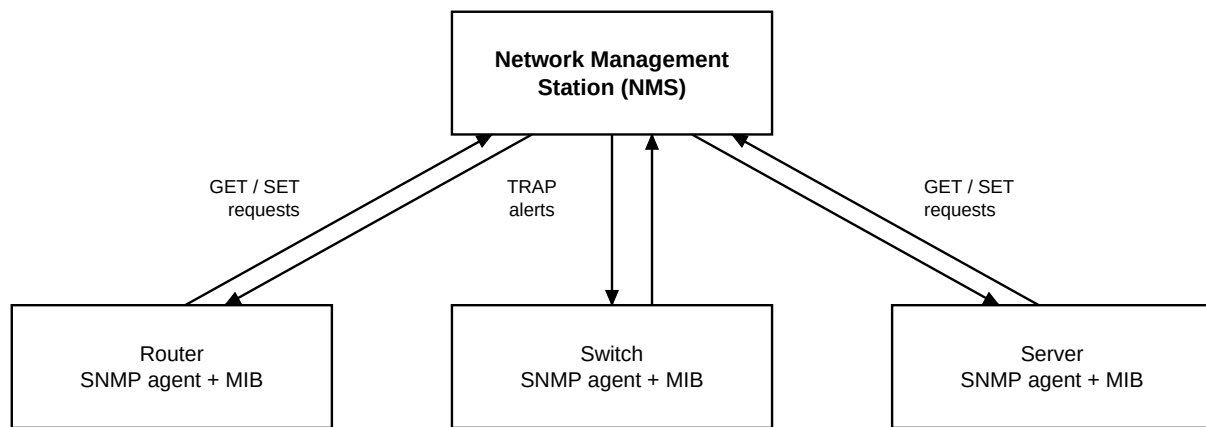


Fig. 7.4: SNMP management model

Table 7.8: SNMP terms

SNMP element	Role
Manager (NMS)	Collects data, shows graphs and raises alarms.
Agent	Software on the device that answers requests and reports events.
MIB	Database of objects (variables) identified by Object Identifiers (OIDs).
GET / SET	Manager reads a variable / changes a value on the device.
TRAP	Unrequested alert sent by an agent when something happens (for example link down).
Community string (v1/v2c)	Password-like text used for access; SNMPv3 adds authentication and encryption.

Table 7.9: Monitoring tools

Tool	Function
Nagios, Zabbix, PRTG	Monitor hosts and services; send e-mail or SMS alerts when thresholds are crossed.
MRTG, Cacti	Draw long-term graphs of bandwidth and other values collected through SNMP.
NetFlow / sFlow collectors	Analyse which hosts and applications use the bandwidth.
Syslog server	Collects log messages from many devices in one place.
Windows Event Viewer, Performance Monitor	Show system events and counters on Windows hosts.
top, vmstat, iostat (Linux)	Show CPU, memory and disk load on a server.

Table 7.10: Important performance metrics

Metric	Meaning	Sign of trouble
Bandwidth utilisation	Percentage of link capacity in use.	Constantly above about 70-80% indicates congestion.
Latency	Time for a packet to reach the destination.	Sudden rise means congestion or routing change.
Jitter	Variation in latency.	Affects voice and video quality.
Packet loss	Percentage of packets that never arrive.	Any steady loss indicates a fault.
Error rate	CRC errors, collisions and discards on an interface.	Bad cable, duplex mismatch or failing NIC.
Availability (uptime)	Fraction of time a device or service is working.	Failing to meet the SLA.

Table 7.11: Choosing a tool by symptom

Symptom	First tool to use
Link light is off	Cable tester, swap cable and port
Cannot reach the Internet	ipconfig / ifconfig, ping ladder, traceroute
Web site names do not open	nslookup, dig
Service not reachable	netstat, nmap, telnet host port
Network slow	Monitoring graphs, Wireshark, interface error counters
Suspicious traffic	Wireshark/tcpdump, firewall and sys log logs

7.3 Internal Security

7.3.1 Why Internal Security Matters

Many people imagine security as a wall that keeps outsiders away. In reality a large share of security incidents begins **inside** the organisation, caused by careless mistakes, weak passwords, curious users or dishonest employees. Internal security means controlling who may enter the system, what they may do once inside, and how they behave. Three goals, together called the **CIA triad**, guide all security work.

Table 7.12: The CIA triad

Goal	Meaning	Example control
Confidentiality	Only authorised persons can read the information.	Passwords, permissions, encryption
Integrity	Information is accurate and cannot be changed without authority.	Write permissions, checksums, backups
Availability	Systems and data are usable when needed.	Redundancy, UPS, backups, DoS protection

Access control is built on three actions called **AAA: Authentication** (proving who you are, for example by password), **Authorization** (deciding what you may do) and **Accounting** (recording what you did).

7.3.2 Account Security

The user account is the main gateway into a system, so it has to be protected carefully. A stolen or guessable password gives an attacker all the rights of that user. Account security rests on the following measures.

Strong password policy

Table 7.13: Typical password policy

Policy setting	Recommended practice
Minimum length	At least 8 characters; 12 or more for administrators.
Complexity	Mix of upper and lower case, digits and symbols; not a dictionary word, name or date of birth.
Maximum age	Change at regular intervals, commonly 60 to 90 days.
Password history	Remember several previous passwords so they cannot be reused immediately.
Account lockout	Lock the account for a period after a few failed attempts (for example 3 to 5).
Default passwords	Change vendor defaults on routers, switches and software immediately.

- **Least privilege:** give every user only the rights needed for the job, and give administrators a separate normal account for daily work.
- **Disable or delete** unused, guest and former employees' accounts without delay.
- **No shared accounts:** each person must be identifiable so that actions can be traced.
- **Protect stored passwords:** systems should store only salted hashes (as in `/etc/shadow`), never the plain text.
- **Audit logs:** review failed and unusual logins (for example in `/var/log/secure` or the Windows Security log).
- **Separation of duties:** no one person should control a whole critical process.

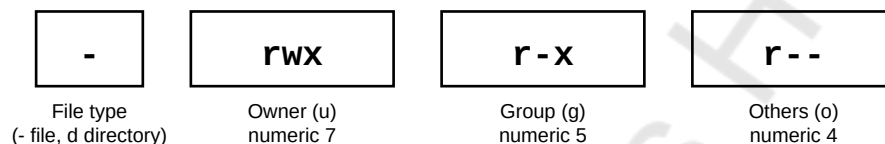
Table 7.14: Authentication factors

Authentication factor	Description	Example
Something you know	Secret memorised by the user.	Password, PIN
Something you have	Physical object held by the user.	Smart card, token, mobile phone
Something you are	Biological characteristic (biometrics).	Fingerprint, iris, face

Multi-factor authentication (MFA) combines two or more different factors, for example a password plus a one-time code sent to a phone. Even if the password is stolen, the attacker still lacks the second factor. On Linux, the commands learnt in Chapter 6 enforce these rules: `chage` for ageing, `usermod -L` for locking and `passwd -e` for forcing a change.

7.3.3 File and Directory Permissions

Permissions control which users can read, change or run a file. Correct permissions protect **confidentiality** and **integrity** even when an attacker or a careless user has an account on the machine. In UNIX/Linux every file has an owner, a group and permissions for owner, group and others.



Equivalent command: `chmod 754 filename`

Fig. 7.5: Reading a permission string such as `-rwxr-xr--`

Table 7.15: Meaning of permissions

Symbol	On a file	On a directory	Value
r (read)	View contents	List the names inside	4
w (write)	Modify contents	Create, delete or rename files inside	2
x (execute)	Run as a program	Enter the directory (Cd) and access files	1

# <code>chmod 640 payroll.txt</code>	owner rw-, group r--, others none
# <code>chmod o-rwx /home/student</code>	remove all rights for others
# <code>chown alice:accounts payroll.txt</code>	
# <code>umask 027</code>	new files 640, new directories 750
# <code>find / -type f -perm -0002</code>	find world-writable files
# <code>find / -perm -4000 -type f</code>	list SUID programs (audit)

The **umask** subtracts permissions from the default modes (666 for files and 777 for directories). With `umask 022`, new files become 644 and new directories 755; with `umask 027`, they become 640 and 750. Dangerous settings include mode 777 on data, world-writable directories without the sticky bit, and unnecessary SUID programs.

Windows (NTFS) permissions

On NTFS volumes permissions are kept in an **Access Control List (ACL)**, a list of entries naming a user or group and the rights granted or denied. Permissions are normally **inherited** from the parent folder. When a folder is shared over the network, the **share** permissions and the **NTFS** permissions are both checked, and the **more restrictive** result applies. An explicit **Deny** overrides any Allow.

Table 7.16: Standard NTFS permissions

NTFS permission	Allows
Full control	Everything, including changing permissions and taking ownership.
Modify	Read, write, delete and execute.
Read & execute	View and run files.
List folder contents	See names of files and subfolders.
Read	View files and attributes.
Write	Create files and change content.

- Assign permissions to **groups**, not to individual users; put users into groups.
- Remove the default "Everyone" full control entries.
- Sensitive files should also be **encrypted** (for example EFS on Windows, GPG on Linux) and backed up.
- Review permissions periodically, especially after staff changes.

7.3.4 Practices and User Education

Technology alone cannot protect a network because the user is often the weakest link. Attackers know that tricking a person is easier than breaking an encryption algorithm. For this reason every organisation needs a written **security policy** and a programme to teach users about it.

Table 7.17: Important security policies

Policy	Contents
Acceptable Use Policy (AUP)	What users may and may not do with computers, e-mail and the Internet.
Password policy	Length, complexity, ageing, and no sharing of passwords.
Remote access / BYOD policy	Rules for VPN use and for personal devices connected to the network.
Backup and recovery policy	What is backed up, how often, where stored, and how restore is tested.
Incident response policy	Who must be told and what steps are taken when an incident happens.

Table 7.18: Common social-engineering techniques

Social engineering	Description
Phishing	Fake e-mail or web site that tricks the user into giving passwords or opening malware.
Pretexting	Attacker invents a believable story (such as posing as help desk) to obtain information.
Tailgating / piggybacking	Following an authorised person through a secured door.
Shoulder surfing	Watching someone type a password or read a screen.
Dumpster diving	Searching rubbish for papers or media containing sensitive data.
Baiting	Leaving an infected USB drive where someone will plug it in.

What users should be taught

- Never reveal passwords, even to someone claiming to be from the IT department.
- Recognise suspicious e-mails; do not click unknown links or open unexpected attachments.
- Lock the screen when leaving the desk and keep a clean desk.
- Do not install unauthorised software or plug in unknown USB drives.
- Use only approved methods to transfer or store confidential data.
- Report lost devices, suspected infections and strange behaviour at once.

Good administrative practices

- Keep operating systems and applications **patched** (see Chapter 6 for up2date and RPM).
- Take regular **backups** and test the restore (the 3-2-1 rule: three copies, two media types, one off-site).
- Enable **logging and auditing**, and actually read the logs.
- Disable unused services and ports; use secure protocols (SSH, HTTPS) in place of telnet and FTP.
- Protect equipment physically: locked server rooms, cable locks, controlled visitor access.
- Repeat training regularly and test users with simulated phishing messages.

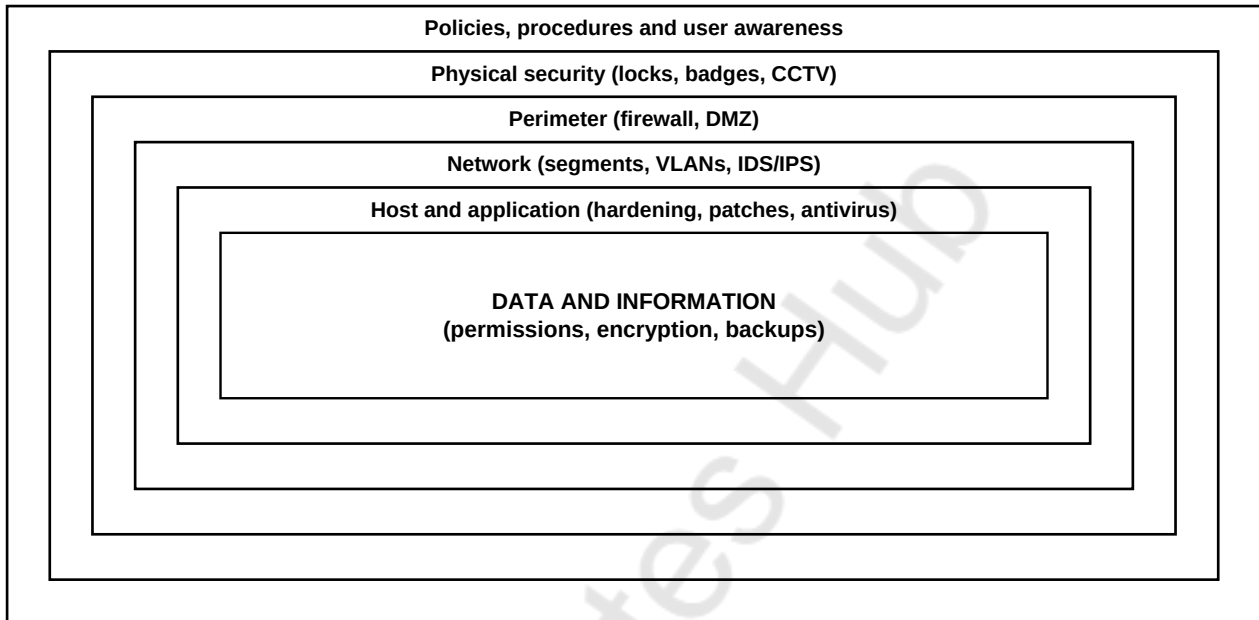


Fig. 7.6: Defence in depth: layers of protection around the data

7.4 External Threats

7.4.1 Who Attacks and How

An **external threat** comes from outside the organisation. The attackers differ in skill and motive: **hackers** and crackers who like the challenge, **script kiddies** who run tools written by others, criminals who want money, **hacktivists** with political goals, competitors, and state-sponsored groups. Most attacks follow a pattern.

1. **Reconnaissance:** collecting public information about the target (domain names, IP ranges, employees).
2. **Scanning:** using tools such as nmap to find live hosts, open ports and software versions.
3. **Gaining access:** exploiting a weakness or a stolen password.
4. **Maintaining access:** installing a back door or rootkit so that the attacker can return.
5. **Covering tracks:** erasing or modifying logs.

7.4.2 Front Door Threats

A **front door threat** is an attack made through the normal, authorised entrance of a system: the login prompt, a public network service or a user. The attacker tries to be accepted as a legitimate user, or finds a flaw in the service that is deliberately exposed to the network.

- **Password guessing:** a *brute-force* attack tries every combination; a *dictionary* attack tries likely words.
- **Stolen or default credentials:** passwords obtained by phishing, key-loggers or from factory settings.
- **Exploiting vulnerable services:** bugs such as buffer overflows in web, mail or FTP servers that are not patched.
- **Social engineering:** persuading a user or the help desk to reveal access.
- **Session hijacking and sniffing:** capturing credentials sent in clear text (telnet, FTP, HTTP).
- **Man-in-the-middle:** secretly relaying and altering communication between two parties.

Countermeasures

- Strong passwords, lockout after failures, and multi-factor authentication.
- Prompt patching and removal of unnecessary services.
- Firewalls to allow only required ports; encryption (SSH, TLS, VPN).
- Logging, intrusion detection and user training.

7.4.3 Back Door Threats

A **back door** is a hidden way of entering a system that **bypasses the normal authentication and security controls**. It may be planted by an attacker after the first break-in (so that access remains even if the password is changed), left by a programmer as a "maintenance hook", or installed by a Trojan horse or rootkit. Back doors can also be physical or architectural, such as an unauthorised modem, a rogue wireless access point or an unprotected remote-administration service.

- Trojan programs that open a hidden listening port for remote control.
- **Rootkits** that hide the attacker's files and processes from the administrator.
- Extra user accounts or SSH keys quietly added by an intruder.
- Debug or maintenance accounts with hard-coded passwords in software or devices.
- Unauthorised access points or modems connected by staff.

Table 7.19: Front door versus back door threats

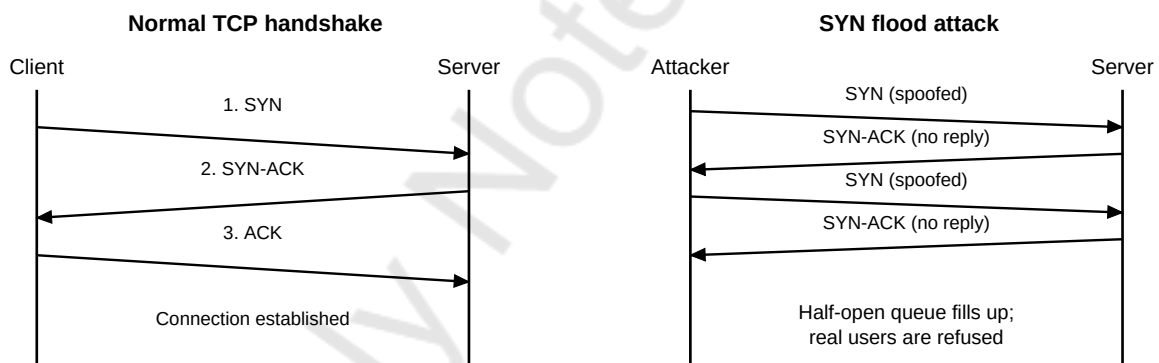
Aspect	Front door	Back door
Entry point	Normal login or public service.	Hidden path that skips normal checks.
Attacker's goal	Gain first access.	Keep access or regain it later.
Typical methods	Password guessing, exploits, phishing.	Trojans, rootkits, secret accounts, rogue devices.
Defence	Strong authentication, patches, firewall.	Integrity checks, port audits, malware scans, change control.

Detecting back doors: list listening ports regularly and look for unknown ones; run rootkit scanners; compare critical files with known-good checksums using integrity checkers such as Tripwire or AIDE; review accounts and startup entries; and scan the network for unauthorised devices.

```
# netstat -tulnp          every listening port with its program
# nmap -sT -p 1-65535 localhost full TCP port scan of own machine
# rpm -Va                verify installed files against the RPM database
# chkrootkit             scan for known rootkits
# awk -F: '$3==0' /etc/passwd accounts with UID 0 (should be only root)
```

7.4.4 Denial of Service Threats

A **Denial of Service (DoS)** attack does not try to steal data. It tries to make a system, service or network **unavailable** to its legitimate users by exhausting its resources: bandwidth, memory, CPU or connection tables. When the attack is launched from many computers at once, usually infected machines of a **botnet**, it is a **Distributed DoS (DDoS)** and is much harder to block, because the traffic comes from thousands of sources.

*Fig. 7.7: Normal TCP three-way handshake compared with a SYN flood***Table 7.20: Common denial-of-service attacks**

Attack	How it works
SYN flood	Many SYN packets (often with spoofed source addresses) leave half-open connections that fill the server's queue.
ICMP (ping) flood	Huge numbers of ping requests consume bandwidth and CPU.
Smurf attack	Ping sent to a broadcast address with the victim's spoofed address, so all hosts reply to the victim.
Ping of death	Oversized or malformed ICMP packet crashes unpatched systems.
UDP flood	Large volume of UDP packets sent to random ports.
Reflection / amplification	Small requests to open DNS or NTP servers produce large replies sent to the victim.
HTTP flood	Many legitimate-looking web requests overload a web server.

Countermeasures

- Filter spoofed addresses at the edge (ingress and egress filtering) and block directed broadcasts.
- Enable **SYN cookies** and limit connection rates, for example `sysctl -w net.ipv4.tcp_syncookies=1` on Linux.
- Use firewalls and IPS devices with rate limiting; keep systems patched.
- Provide spare bandwidth and redundancy; use CDN or DDoS-scrubbing services for public sites.
- Prepare an incident plan and contact the ISP quickly when an attack starts.

7.4.5 Viruses, Worms and Other Malicious Code

Malware (malicious software) is any program designed to damage a system, steal information or give an attacker control. Different kinds of malware are distinguished by how they spread and what they do.

Table 7.21: Types of malicious code

Type	Description and spread	Example
Virus	Attaches itself to a host file or program and spreads when the infected file is run or shared; needs user action.	Melissa (macro virus, 1999)
Worm	Self-replicating program that spreads across networks on its own without a host file, using vulnerabilities or e-mail.	Slammer (2003), Blaster (2003), Conficker (2008)
Trojan horse	Appears to be useful software but carries a hidden harmful function; does not replicate itself.	Fake utilities, remote-access Trojans
Logic / time bomb	Code lying dormant until a condition or date is met, then destroys or alters data.	Disgruntled-employee scripts
Rootkit	Hides the presence of malware and gives privileged access.	Kernel-level rootkits
Spyware / keylogger	Secretly records activity or keystrokes and reports to an attacker.	Banking keyloggers
Adware	Displays unwanted advertisements and tracks browsing.	Browser toolbars
Ransomware	Encrypts the victim's files and demands payment for the key.	WannaCry (2017)
Botnet / bot	Infected machines controlled remotely , used for DDoS and spam.	Mirai (IoT botnet, 2016)
Macro virus	Virus written in the macro language of office documents.	Melissa
Boot-sector virus	Infects the boot sector of a disk and loads before the operating system.	Older floppy-disk viruses

Table 7.22: Virus compared with worm

Feature	Virus	Worm
Needs a host file	Yes	No
Spreads by	Users running or sharing infected files	Automatically over the network
User action needed	Usually yes	Often none
Main damage	Corrupts files and programs	Consumes bandwidth; creates back doors; may carry payloads

A virus has a typical life cycle: a **dormant** phase where it hides, a **propagation** phase in which it copies itself into other files, a **triggering** phase when a condition occurs, and an **execution** phase in which the harmful payload is delivered. Antivirus software defends by comparing files with **signatures** of known malware, by **heuristic** analysis of suspicious structure, and by **behaviour monitoring**.

Protection against malware

- Install antivirus software, keep signatures updated, and scan e-mail and downloads.
- Apply security patches quickly; most worms exploit known, already-fixed flaws.
- Disable macros and automatic execution from removable media.

- Run users with limited rights so that malware cannot change the whole system.
- Segment the network so that an infection cannot spread everywhere.
- Keep offline backups; they are the best defence against ransomware.
- Teach users not to open unexpected attachments or install pirated software.

7.4.6 Defensive Technologies

A **firewall** examines traffic between networks and permits or blocks it according to rules. A **DMZ** (demilitarised zone) is a separate segment for public servers such as web and mail. If a DMZ server is compromised, the inner firewall still protects the internal LAN.

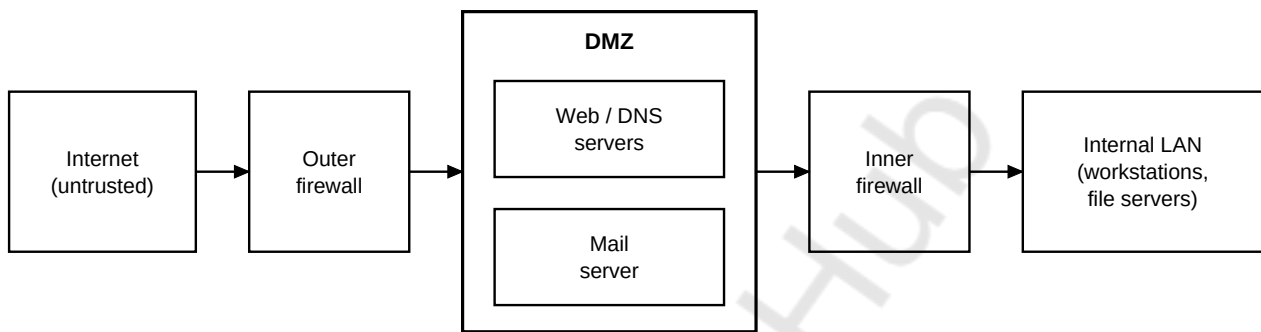


Fig. 7.8: A network with two firewalls and a DMZ

Table 7.23: Types of firewall

Firewall type	Operation
Packet filter	Checks each packet's source and destination address, port and protocol against a rule list.
Stateful inspection	Tracks connections, so that only replies to permitted requests are allowed in.
Proxy / application gateway	Acts on behalf of clients and inspects application-level content.
Next-generation firewall	Adds application awareness, intrusion prevention and content filtering.

Table 7.24: IDS compared with IPS

Feature	IDS (Intrusion Detection System)	IPS (Intrusion Prevention System)
Placement	Monitors a copy of traffic (passive).	Sits in the traffic path (inline).
Action	Detects and raises an alert.	Detects and blocks the attack automatically.
Risk	Attack is not stopped by the device.	False positives may block legitimate traffic.

# iptables -P INPUT DROP	default: deny incoming
# iptables -A INPUT -i lo -j ACCEPT	allow loopback
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT	
# iptables -A INPUT -p tcp --dport 22 -j ACCEPT	allow SSH
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT	allow web
# service iptables save	keep rules after reboot

- **VPN** and encryption (IPsec, SSL/TLS, SSH) protect data crossing untrusted networks.
- **Vulnerability scanning** and **penetration testing** find weaknesses before attackers do.
- **Honeypots** are decoy systems that attract attackers and reveal their methods.
- **Network Address Translation (NAT)** hides internal addresses from the outside.

7.4.7 Incident Response

Table 7.25: Incident response phases

Phase	Activities
1. Preparation	Policies, trained team, tools, backups, contact lists.
2. Identification	Confirm that an incident has occurred; determine the scope.
3. Containment	Isolate affected systems (unplug from the network) to stop the spread.
4. Eradication	Remove malware, close the weakness, reset passwords.
5. Recovery	Restore from clean backups, monitor closely.
6. Lessons learned	Document what happened and improve defences.

Quick Revision Tables

Table 7.26: Threats and countermeasures at a glance

Threat	Main countermeasures
Password guessing	Strong passwords, logout, MFA
Phishing / social engineering	User training, mail filtering, verification procedures
Unpatched service exploit	Patch management, vulnerability scans, firewall
Back door / rootkit	Integrity checkers, port audits, antivirus, change control
DoS / DDoS	Rate limiting, SYN cookies, filtering, spare bandwidth, scrubbing service
Virus / worm	Antivirus, patches, least privilege, segmentation
Ransomware	Offline backups, patching, user training
Insider misuse	Least privilege, auditing, separation of duties, policy
Eavesdropping	Encryption (SSH, TLS, VPN), switched network

Table 7.27: Key terms

Term	One-line definition
Troubleshooting	Systematic process of finding and fixing faults.
Segmenting	Dividing the problem into layers or parts.
Isolating	Narrowing down to the single faulty component.
Baseline	Recorded picture of normal network behaviour.
SNMP	Protocol for monitoring and managing devices.
Least privilege	Give only the access needed to do the job.
CIA triad	Confidentiality, Integrity, Availability.
DMZ	Segment for public servers between two firewalls.
Botnet	Group of infected machines controlled by an attacker.

Practice Questions

Part A: Multiple Choice Questions (15)

Choose the correct option. The answer key is given after the questions.

- Q1.** What is the first step of the structured troubleshooting method?
- (a) Replace the suspected hardware
 - (b) Identify the problem
 - (c) Reboot every device
 - (d) Write the final report
- Q2.** Which troubleshooting approach begins at the Physical layer and moves upward?
- (a) Top-down
 - (b) Substitution
 - (c) Follow the path
 - (d) Bottom-up
- Q3.** Which problem should normally receive the highest priority?
- (a) Failure of a core link or server that affects all users
 - (b) A user's screen saver does not start
 - (c) A request for a new desktop wallpaper
 - (d) A printer prints in the wrong colour
- Q4.** The command ping 127.0.0.1 is used to test:
- (a) The Internet connection
 - (b) The default gateway
 - (c) The local TCP/IP software
 - (d) A remote DNS server
- Q5.** Which hardware tool sends a pulse and measures reflections to locate a break in a cable?
- (a) Crimper
 - (b) TDR
 - (c) Loopback plug
 - (d) Tone generator
- Q6.** Which command shows the list of routers (hops) between a host and its destination?
- (a) traceroute / tracert
 - (b) arp
 - (c) netstat
 - (d) hostname
- Q7.** Which protocol is used by a management station to monitor and manage network devices?
- (a) SMTP
 - (b) FTP
 - (c) ICMP
 - (d) SNMP

- Q8.** The principle of giving users only the access they need to do their work is called:
- (a) Defence in depth
 - (b) Non-repudiation
 - (c) Least privilege
 - (d) Redundancy
- Q9.** The command `chmod 754` file gives the 'others' category:
- (a) Read permission only
 - (b) Read and write permission
 - (c) Execute permission only
 - (d) No permission
- Q10.** Which of the following is an example of the authentication factor 'something you are'?
- (a) Password
 - (b) Smart card
 - (c) PIN
 - (d) Fingerprint
- Q11.** Tricking a person into revealing confidential information is known as:
- (a) Spoofing
 - (b) Social engineering
 - (c) Sniffing
 - (d) Fragmentation
- Q12.** A hidden method of entering a system that bypasses normal authentication is called a:
- (a) Front door attack
 - (b) DMZ
 - (c) Back door
 - (d) Honeypot
- Q13.** An attack that sends many TCP SYN packets and leaves half-open connections is a:
- (a) SYN flood
 - (b) Phishing attack
 - (c) Keylogger
 - (d) Rootkit
- Q14.** Which malicious program replicates itself across a network without needing a host file?
- (a) Logic bomb
 - (b) Macro virus
 - (c) Trojan horse
 - (d) Worm
- Q15.** A DMZ is a network segment that:
- (a) Stores only backup tapes
 - (b) Hosts public-facing servers between the outside and the internal network
 - (c) Is used only for wireless users
 - (d) Encrypts all traffic automatically

Answer Key for MCQs

Q.	Answer	Reason
1	(b)	Step 1 is to identify the problem by collecting information.
2	(d)	Bottom-up starts at layer 1 (physical) and works upward.
3	(a)	Impact on all users and critical services sets the highest priority.
4	(c)	127.0.0.1 is the loopback address and tests the local TCP/IP stack.
5	(b)	A Time Domain Reflectometer finds cable breaks from reflected pulses.
6	(a)	<code>tracert</code> (Linux) or <code>tracert</code> (Windows) lists each hop.
7	(d)	SNMP lets the NMS poll agents and receive traps.
8	(c)	Least privilege limits damage from mistakes and attacks.
9	(a)	7=rwx for owner, 5=r-x for group, 4=r-- for others.
10	(d)	Biometrics such as a fingerprint are 'something you are'.
11	(b)	Social engineering manipulates people rather than systems.
12	(c)	A back door skips the normal login and security controls.
13	(a)	A SYN flood exhausts the server's half-open connection queue.
14	(d)	Worms are self-replicating and spread over networks on their own.
15	(b)	The DMZ holds public servers so that the internal LAN stays protected.

Part B: Broad Questions (10)

1. What is troubleshooting? Explain the seven steps of a systematic troubleshooting method with a neat flow diagram.
2. Describe the different troubleshooting approaches (top-down, bottom-up, divide and conquer, follow the path, substitution and comparison) and state when each is most suitable.
3. Explain what is meant by segmenting and isolating a network problem. Using the OSI model and a ladder of ping tests, show how a connectivity fault is located.
4. How are priorities assigned to network problems? Discuss the factors involved and give an example priority scheme with response times.
5. List and explain the hardware tools used in network troubleshooting. Describe the working of a cable tester, TDR and tone generator.
6. Discuss the software tools used for troubleshooting (ping, traceroute, nslookup, netstat, nmap, Wireshark). Show with examples what each tool reveals.
7. Explain how SNMP-based network monitoring works. Describe the manager, agent, MIB and trap with a diagram, and explain the importance of a baseline.
8. What is internal security? Explain the CIA triad and discuss account security measures including password policy, least privilege and multi-factor authentication.
9. Explain file and directory permissions in UNIX/Linux and Windows NTFS. Why are correct permissions important? Describe umask and the special permission bits.
10. Distinguish between front door and back door threats. Describe denial-of-service attacks, viruses, worms and other malicious code, and explain the countermeasures including firewalls, DMZ and IDS/IPS.

End of Notes