ALevel CS C09 Security, privacy and data integrity

Data integrity: a requirement for data to be **accurate** and **up to date**
Data privacy: a requirement for data to be **available only to authorised users**
Data security: a requirement for data to be available for use when needed, ensures that **only authorised users have access to the system** and **data can be recovered if lost or corrupted**.
two primary aims of system security measures:
1. to ensure that the system continues to carry out the tasks users need
2. to ensure that only authorised users have access to the system.

The threats to the security of a system:
1. individual user not taking appropriate care
2. internal mismanagement
3. natural disasters
4. unauthorised intrusion into the system by an individual
5. malicious soft ware entering the system.

Malware: malicious software that has the intention of causing harm to a system or its contents
Types of malware-containing program code:
1. **virus**: tries to replicate itself inside other executable code
2. **worm**: runs independently and transfers itself to other network hosts
3. **logic bomb**: stays inactive until some condition is met
4. **Trojan horse**: replaces all or part of a previously useful program
5. **spyware**: collects information and transmits it to another system
6. **bot**: takes control of another computer and uses it to launch attacks.

Malware can be classified in terms of the activity:
1. **phishing**: sending an email or electronic message from an apparently legitimate source requesting confidential information
2. **pharming**: setting up a bogus website which appears to be a legitimate site
3. **keylogger**: recording keyboard usage by the legitimate user of the system.

Similarities between phishing and pharming:
1. both malware software
2. both collect personal data
3. via fake website
4. the data are then used illegally

Differences between phishing and pharming:
1. phishing use emails or direct users to fake website
2. pharming misdirected browser to a bogue website, by modifying entries on a DNS server. by being installed on your computer

System vulnerability arising from user activity
do not involve malware:
1. The use of **weak passwords** and particularly those which have a direct connection to the user.
2. A legitimate user not recognising a phishing or pharming attack and, as a result, **giving away sensitive information**.

actions that might introduce malware:
1. attaching a portable storage device
2. opening an email attachment
3. accessing a website
4. downloading a file from the Internet.

System vulnerability arising from within the system itself
1. Operating systems often **lack good security**. Operation system need regular update.
2. In the past, commonly used application packages allowed **macro viruses to spread**, but this particular problem is now largely under control.
3. A very specific vulnerability is **buffer overflow**.

Security measures for protecting data

Validation: a check that data entered is of the correct type and format; it does not guarantee that data is accurate
validation type:
1. presence check: an entry field is not left blank
2. format check: a date has to be dd/mm/yyyy
3. length check: a telephone number
4. range check: the month in a date must not exceed 12
5. limit check: a maximum number of years for a person's age
6. type check: only a numeric value for the month in a date
7. existence check: a file exists with the filename referred to in the data entry.

Security measures for protecting computer systems

Disaster recovery: to protect continuity of operation.
If an organisation has a full system always ready to replace the normally operational one, it is referred to as a 'hot site'.
Safe system update: A special case of system vulnerability arises when there is a major update of hardware and/ or soft ware. Organisations may need to have the original system and its replacement running in parallel for a period to ensure continuity of service.
User authentication
Authentication: verification of a user's identity
methods of authentication:
1. biometric methods
2. security tokens

Good practice:
1. not leaving the computer switched on when unattended
2. not allowing someone else to observe you accessing the computer
3. not writing down details of how you access it.
4. a policy banning the use of such devices or at least limiting their use.

Firewall: **hardware** or **software** that **monitors and controls network traffic**
Data must enter the system, but it can be inspected immediately. A firewall can inspect the system addresses identified in the transmission of data, but can sometimes also inspect the data itself to check for anything unusual or inappropriate.
Firewall protect method:
1. Prevents unauthorised access to the data
2. Monitors incoming and outgoing traffic
3. Blocks transmissions from unauthorised sources / websites / ports
3. Maintains an allow list / deny list of IP addresses

Digital signature: check the identity of the sender.
Anti-virus software and intrusion detection:
1. **anti-virus software**: This carries out regular scans to detect any malware and to remove or deactivate it.
2. **intrusion detection system**: take as input an audit record of system use and look for examples that do not match expected system activity.

Security measures for protecting data

Recovering from data loss:
reasons for accidental loss of data:
1. a disk or tape gets corrupted
2. a disk or tape is destroyed
3. the system crashes
4. the file is erased or overwritten by mistake
5. the location of the file is forgotten.

backup procedure:
1. a full backup is made at regular intervals, perhaps weekly
2. at least two generations of full backup are kept in storage
3. incremental backups are made on a daily basis.

backup program: effectively freezes the file store while data is being copied.
disk-mirroring strategy: data is simultaneously stored on two disk systems during the normal operation of the system.
Restricting access to data
authorisation policy: definition of a user's access rights to system components
Protecting data content
Data can be **encrypted** to ensure data security.

Verification: confirmation of data received by a system. verification means getting the user to confirm that the data entered was what was intended to be entered.
one-bit parity check:
1. At the transmitting end, the number of 1s in the seven-bit code is counted.
2. If the count gives an even number, the parity bit is set to 0.
3. If the count gives an odd number, the parity bit is set to 1.
4. This is repeated for every byte in the transmission.
5. At the receiving end, the number of 1s in the eight-bit code is counted.
6. If the count gives an even number, the byte is accepted.
7. This is repeated for every byte in the transmission.

two bits to be flipped in an individual byte, which would mean that the transmission is incorrect but the parity check is passed.
checksum:
1. At the transmitting end a block is defined as a number of bytes.
2. The sum of these binary numbers in a block is calculated and supplied as a checksum value in the transmission.
3. The receiver does the same calculation and checks the sum of the numbers with the checksum value transmitted for each block in turn.

an error can be detected but its position in the transmission cannot be determined.