Security measures

Security, privacy, integrity

What is the difference between data security, privacy and integrity?

Term	Definition	Key focus	Example
Data Security	Protecting data from **unauthorised access**, theft, or attacks	Preventing breaches or leaks	Using encryption, firewalls, and passwords to protect stored data
Data Privacy	Ensuring data is **collected, stored, and shared** in a way that respects the user's rights	Controlling who can access and use personal data	Asking for user consent before collecting or sharing personal information
Data Integrity	Ensuring data is **accurate, complete, and unaltered** during storage or transfer	Maintaining correctness and reliability	Using checksums or validation rules to detect accidental or unauthorised changes

- **Security** = keeping data safe from threats (e.g. hackers)
- **Privacy** = making sure data is used **fairly** and **with consent**
- **Integrity** = making sure data stays **accurate** and **unchanged**

Protection techniques

User accounts

- User accounts are designed to **control access** to computer systems
- They help protect data and system resources by ensuring only **authorised individuals** can log in and perform specific tasks

Passwords

- Passwords are a **digital lock** to prevent unauthorised access to an account
- They are often stored as an **encrypted/ciphered** text entry in a database, ensuring that even with unauthorised access to a database, a hacker would not be able to gain access to the individual passwords of users

Authentication

- Authentication is the process of **ensuring that a system is secure** by asking the user to **complete tasks to prove they are an authorised** user of the system
- Authentication is done because **bots can submit data in online forms**
- Authentication can be done in several ways, including:
  - **Digital signatures**
  - **Biometrics**

Digital signatures

- A digital signature is a secure way to **prove that a digital message or document was sent by a specific person** and that it **has not been altered**
- It acts like a **digital stamp of approval**, confirming:
  - **Who sent the data** (authentication)
  - **That the data hasn't been changed** (integrity)

Biometrics

- Biometrics are an individuals **personal characteristics used to identify them**, such as
  - **Fingerprints**
  - **Iris/retina scans (eyes)**
  - **Voice recognition**
- Biometrics provide a **very secure method** of confirming a users **identity** before allowing access/permission to a computer system
- Biometric measures are often **used on mobile devices** to provide secure access

Firewall

- A firewall is a **barrier** between a network and the internet
- A firewall prevents **unwanted traffic** from entering a network by filtering requests to ensure they are **legitimate**
- It can be both **hardware** and **software** and they are often used together to provide stronger security to a network
  - **Hardware firewalls** will protect the whole network and prevent unauthorised traffic
  - **Software firewalls** will protect the individual devices on the network, monitoring the data going to and from each computer

Anti-virus software

- Anti-virus software is a term used to describe a combination of different software to prevent computers from being susceptible to **viruses** and other **malicious software**

- Anti-virus **scans** through **email** attachments, **websites** and downloaded **files** to search for issues
- Anti-virus software has a list of known malware **signatures to block** immediately if they try to access your device in any way
- Anti-virus will also perform **checks for updates** to ensure the database of known issues is up to date

Anti-spyware software

- Anti-spyware is a type of **security software** designed to **detect, block, and remove spyware** from a computer system
- Spyware is a type of **malicious software (malware)** that secretly gathers information about a user without their knowledge
- It can:
  - **Record keystrokes** (e.g. passwords, credit card numbers)
  - **Monitor browsing habits**
  - **Access files** and send them to a third party

Encryption

- Encryption is the process of **converting data into a secret code** so that **only authorised users** can read it
- It protects sensitive information (like passwords, personal data, or messages) from **unauthorised access**, especially when data is stored or sent over a network

Type	Description
Symmetric	The **same key** is used to encrypt and decrypt. Fast but key must be shared securely
Asymmetric	Uses a **public key** for encryption and a **private key** for decryption. More secure for sending data

Worked Example

A company has several security measures in place to prevent unauthorised access to the data on its computers.

Describe the difference between the security and privacy of data. **[2]**

Answer

- Security protects data against loss [1 mark]
- Privacy protects data against unauthorised access [1 mark]

Threats

Network & internet risks

What are common network & internet risks?

Hackers

- A **hacker** is someone who gains **unauthorised access** to computer systems or networks
- Hackers often **exploit security weaknesses** to:
  - **Steal sensitive data**
  - **Gain control of systems**
  - **Cause damage or disruption**
- Hackers are considered **cybercriminals** when their actions are illegal or harmful.

How do hackers gain access?

- Hackers look for opportunities or vulnerabilities in systems, such as:
  - **Unpatched software** – Missing security updates can leave systems exposed
  - **Out-of-date anti-malware** – Older protection can fail to detect new threats
  - **Weak passwords** – Simple or reused passwords are easy to guess or crack

Effects of a hacker attack

- When a hacker successfully gains access, it can lead to:

Effect	Impact
Data breaches	Personal or company information is leaked or stolen
Malware installation	Hackers may install viruses or spyware to cause further harm
Data loss	Important files may be deleted or corrupted
Identity theft	Stolen personal data can be used to impersonate individuals
Financial loss	Hackers may access bank accounts or demand ransom payments

How can hacking be prevented?

- There are several methods to reduce the risk of being targeted by hackers:
  - Use **strong passwords** that are long and hard to guess

- Enable **two-factor authentication** for extra login security
- Install and regularly update **anti-malware software**
- Use **firewalls** to block unauthorised access to the network
- Keep all **software up to date** with security patches

Phishing

- Phishing is a form of social engineering
- It involves sending **fraudulent, legitimate-looking emails** to a large number of email addresses, claiming to be from a **reputable company** or trusted source to try and **gain access** to your details
- Phishing often tries to **coax** the user to click on a login button to enter their details

What are the effects of phishing?

- The creator of the email can gain unauthorised access to **personal data** such as **login information**, **bank accounts** and more
- Phishing can lead to **identity theft** or **fraudulent activity** on credit cards and bank accounts

How can phishing be prevented?

- Phishing can be prevented by:
  - **Anti-spam filters** to avoid fraudulent emails arriving in a user's inbox
  - **Training staff** to recognise fraudulent emails and to avoid opening attachments from unrecognised senders
  - **User access levels** to prevent staff from being able to open files-types such as executable (**.exe)** files and batch (**.bat**) files

Pharming

- Pharming is typing a website address into a browser and it is **redirected to a 'fake' website** to trick a user into typing in sensitive information such as passwords
- An attacker attempts to **alter DNS** settings or **change a users browser settings** to redirect users to the fraudulent website

What are the effects of pharming?

- The creator of the malicious content can gain unauthorised access to **personal data** such as **login information, bank accounts** and more
- Pharming can lead to **identity theft** or **fraudulent activity** on credit cards and bank accounts

How can pharming be prevented?

- Pharming can be prevented by:

- **Keeping anti-malware software up to date**
- **Checking URLs regularly**
- **Make sure the padlock icon is visible**

Malware

- Malware (**mal**icious soft**ware**) is the term used for any software that has been created with malicious intent to cause harm to a computer system
- Examples of issues caused by malware include
  - Files being **deleted**, **corrupted** or **encrypted**
  - Internet connection becoming **slow** or **unusable**
  - Computer **crashing** or **shutting down**
- Malware can exist in many forms, each designed to perform its role in different ways

Malware	What it Does

Computer virus	▪ A program which can **replicate itself** on a user's computer. It contains code that will cause **unwanted and unexpected events** to occur
	▪ Examples of issues a user may experience are
	  ▪ **Corrupt** files
	  ▪ **Delete** data
	  ▪ **Prevent** applications from running correctly

Trojan	▪ Sometimes also called a **Trojan Horse** ▪ Trojans **disguise** themselves as **legitimate software** but contain malicious code in the background
Spyware	▪ Software which will allow a person to **spy** on the users' **activities** on their devices ▪ This form of software will be embedded into other software such as games or programs that have been downloaded from **illegitimate sources** ▪ Spyware can **record** your screen, log your **keystrokes** to gain access to **passwords** and more

How can malware be prevented?

- To protect against the threat of malware:
  - **Ensure code is written correctly**
  - **Keep anti-malware software up to date**
  - **Install a firewall**
  - **Educate users**

Data protection

Data security techniques

Access rights

- Access rights control **what users can see or do** on a network
- They ensure that users can **access the files and resources they need** for their role, and **can't access information they shouldn't**
- Access rights are assigned based on a user's **role**, **responsibilities**, or **security clearance**

Access right	What it means
Full access	The user can **open, create, edit, and delete** files or folders
Read-only access	The user can **open and view** files but **cannot edit or delete** them
No access	The user **cannot see or interact with** the file or folder at all

Example: access rights on a school network

User group	Access rights
Administrators	**Unrestricted access** – Can access and control all files, folders, and settings
Teaching Staff	**Partially restricted access** – Can access student data but not other staff files
Students	**Restricted access** – Can only access **their own files and folders**

- Users can be **grouped** (e.g. "Year 11", "Staff", "Admin") and given access rights as a group
- Access rights can be assigned to **files**, **folders**, or **whole systems**
- This helps protect **confidential data**, improves **network security**, and supports **efficient collaboration**

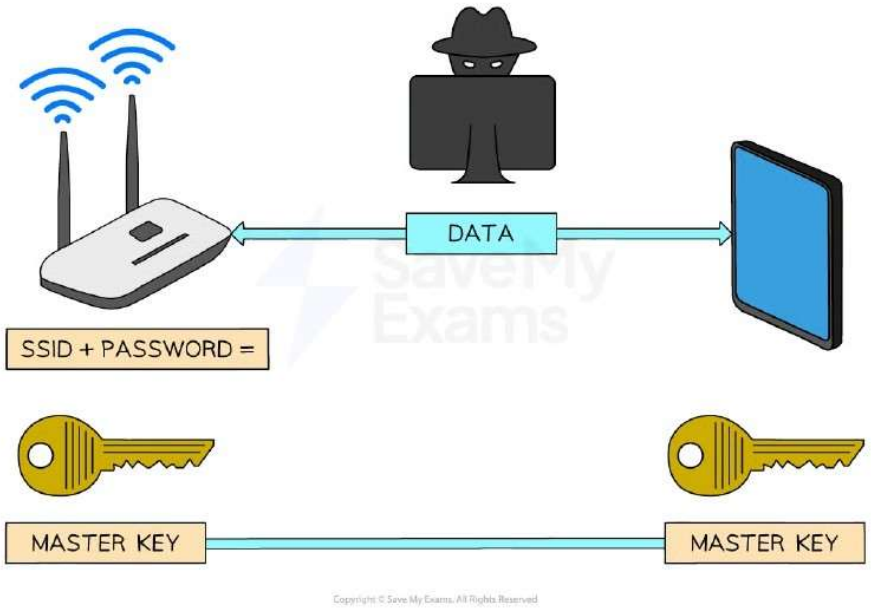
Encryption

- Encryption is a method of **scrambling data** before being transmitted across a network in order to protect the contents from **unauthorised access**

- While encryption is important on both **wired** and **wireless** networks, it's even more critical on wireless networks due to the data being transmitted over radio waves, making it easy to **intercept**

How is wireless data encrypted?

- Wireless networks are identified by a 'Service Set Identifier' (**SSID**) which along with a password is used to create a '**master key**'
- When devices connect to the **same wireless network** using the SSID and password they are given a copy of the master key
- The master key is used to **encrypt** data into '**cipher text**', before being transmitted
- The receiver uses the same master key to **decrypt** the cipher text back to '**plain text**'
- To guarantee the security of data, the **master key is never transmitted**. Without it, any intercepted data is rendered useless
- Wireless networks use dedicated **protocols** like **WPA2** specifically designed for **Wi-Fi** security



How is wired data encrypted?

- Wired networks are encrypted in a very similar way to a wireless network, using a **master key to encrypt** data and the **same key to decrypt** data
- Encryption on a wired network differs slightly as it is often left to **individual applications to decide** how encryption is used, for example **HTTPS**

Integrity methods

Validation & verification

What is validation?

- Validation is an **automated process** where **a computer checks** if a user input is **sensible** and meets the program's requirements
- There are **seven categories** of validation which can be carried out on **fields** and **data types,** these are
  - **Range check**
  - **Format check**
  - **Length check**
  - **Presence check**
  - **Existence check**
  - **Limit check**
  - **Check digit**
- There can be occasions where **more than one type of validation will be used** on a field
- An example of this could be a password field which could have a **length**, **presence** and **type** check on it

What is verification?

- Verification is the **act of checking data is accurate** when being **transferred** or **entered** into a system
- Verification methods include:
  - **Parity check (transfer)**
  - **Checksum (transfer)**
  - **Double entry checking (entry)**
  - **Visual checks (entry)**

Validation methods

Validation type	Definition	Example
Range check	Ensures a number falls within a set range	Validating that a percentage is between 0 and 100

Limit check	Checks that a value does not exceed a maximum limit	Ensuring no more than 5 items can be bought in a special offer
Length check	Checks the length of a string	Confirming a PIN is exactly 4 digits long
Type check	Checks that the input is of the correct data type	Ensuring age is entered as a whole number
Presence check	Checks that data has been entered and the field is not blank	Making sure a registration form is not submitted with blank fields
Existence check	Checks that a referenced value exists in a database or list	Confirming a student ID entered exists in the school records
Format check	Ensures data is in the correct format (e.g. patterns)	Making sure an email address contains '@' and a domain like '.com'

What is a check digit?

- A check digit is the **last digit** included in a code or sequence, used to **detect errors** in **numeric data entry**
- Examples of errors that a check digit can help to identify are:
  - **Incorrect digits entered**
  - **Omitted or extra digits**
  - **Phonetic errors**
- Added to the **end of a numerical sequence** they ensure **validity** of the data
- Calculated using **standardised algorithms** to ensure widespread **compatibility**
- Examples of where check digits can be used include:
  - **ISBN book numbers**
  - **Barcodes**

ISBN book numbers

- Each book has a **unique** ISBN number that **identifies** the book
- A standard ISBN number may be ten digits, for example, 965–448–765–9
- The check digit value is the **final digit** (9 in this example).
- This number is chosen specifically so that when the algorithm is completed the result is a whole number (an integer) with no remainder parts

- A check digit algorithm is performed on the ISBN number and **if the result is a whole number,** then the **ISBN is valid**

Barcodes

- Barcodes consist of **black and white lines** which can be scanned using barcode scanners
- Barcode scanners **shine a laser** on the black and white lines which **reflect light** into the scanner
- The scanner reads the **distance between these lines as numbers** and can identify the item
- Barcodes also use a **set of digits** to **uniquely identify each item**
- The **final digit** on a barcode is usually the **check digit**, this can be used to **validate** and **authenticate** an item

Verification methods

What is a parity check?

- A parity check determines whether **bits in a transmission** have been **corrupted**
- Every byte transmitted has **one of its bits** allocated as a **parity bit**
- The sender and receiver must **agree** before transmission whether they are using **odd** or **even parity**
- If **odd parity** is used then there must be an **odd number of 1's in the byte**, including the parity bit
- If **even parity** is used then there must be an **even number of 1's in the byte**, including the parity bit
- The **value of the parity bit** is determined by **counting the number of 1's in the byte**, including the parity bit
- If the number of 1's **does not match the agreed parity** then an **error has occurred**
- Parity checks **only check that an error has occurred**, they do not reveal where the error(s) occurred

Even parity

- Below is an arbitrary binary string

EVEN Parity bit	Byte						
0	1	0	1	1	0	1	0

- If an **even parity bit** is used then **all bits** in the byte, including the parity bit, must **add up to an even number**

- There are four 1's in the byte
- This means the parity bit must be 0 otherwise the whole byte, including the parity bit, would add up to five which is an odd number

Odd parity

- Below is an arbitrary binary string

ODD Parity bit	Byte						
1	1	0	1	1	0	1	0

- If an **odd parity bit** is used then **all bits** in the byte, including the parity bit, must **add up to an odd number**
  - There are four 1's in the byte. This means the parity bit must be a 1 otherwise the whole byte, including the parity bit, would add up to four which is an even number
- The table below shows a number of examples of the agreed parity between a sender and receiver and the parity bit used for each byte

Example #	Agreed parity	Parity bit	Main bit string							Total number of 1's
#1	ODD	0	1	1	0	1	0	1	1	5
#2	EVEN	1	0	0	0	1	0	0	0	2
#3	EVEN	1	0	1	0	1	1	1	1	6
#4	ODD	1	0	1	1	1	0	0	1	5
#5	ODD	1	1	0	1	0	1	0	1	5
#6	EVEN	0	1	0	0	1	1	1	0	4

- Example #1: The agreed parity is **odd**. All of the 1's in the main bit string are added **(5)**. As this number is **odd already** the **parity bit** is set to **0** so the whole byte **stays odd**
- Example #2: The agreed parity is **even**. All of the 1's in the main bit string are added **(1)**. As this number is **odd** the **parity bit** is set to **1** to make the **total number of 1's even (2)**
- Example #6: The agreed parity is **even**. All of the 1's in the main bit string are added **(4)**. As this number is **even already** the **parity bit** is set to **0** so the whole byte **stays even**

How do errors occur?

- When using parity bits, an **error** occurs when the number of **total bits does not match the agreed parity**

- Bits can be **flipped** or **changed** due to **interference** on a wire or wirelessly due to **weather** or **other signals**

Example #	Agreed parity	Parity bit	Main bit string							Total number of 1's	Error
#1	ODD	1	1	1	0	1	0	1	1	6	Error
#2	EVEN	1	0	0	0	1	0	0	0	2	No error
#3	EVEN	1	0	1	1	1	1	1	1	7	Error
#4	ODD	1	0	1	1	1	0	0	1	5	No error
#5	ODD	1	1	0	1	0	1	1	1	6	Error
#6	EVEN	0	1	0	0	0	1	1	0	3	Error

- Parity checks are **quick** and **easy to implement** but **fail to detect bit swaps** that cause the parity to remain the same

What are parity byte & block checks?

- Parity blocks and parity bytes can be used to **check an error has occurred** and **where the error is located**
- Parity checks on their own **do not pinpoint where errors in data exist**, only that an error has occurred
- A parity block consists of **a block of data** with the number of 1's totalled **horizontally** and **vertically**
- A parity byte is also sent with the data which contains the **parity bits** from the **vertical parity calculation**
- Below is a parity block with a parity byte at the bottom and a parity bit column in the second column

ODD	Parity bit	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
Byte 1	0	1	1	0	1	0	1	1
Byte 2	0	0	0	0	1	0	0	0
Byte 3	1	0	1	0	1	1	1	1

Byte 4	1	0	1	1	1	0	0	1
Byte 5	1	1	0	1	0	1	0	1
Byte 6	1	1	0	0	1	1	1	0
Byte 7	0	0	1	1	1	1	1	0
Byte 8	0	1	0	1	1	0	0	0
Parity byte	0	1	1	1	1	1	1	1

- The above table uses **odd parity**
- Each **byte row** calculates the **horizontal parity** as a parity bit **as normal**
- Each **bit column** calculates the **vertical parity** for each row, the **parity byte**
- It is **calculated before transmission** and **sent with the parity block**
- Each **parity bit tracks if a flip error** occurred in a byte while the **parity byte** calculates if an **error occurred in a bit column**
- By **cross referencing** both **horizontal and vertical** parity values the **error can be pinpointed**
- In the above example the **byte 3 / bit 5 cell** is the error and **should be a 0 instead**
- The error could be **fixed automatically or a retransmission request could be sent** to the sender

What is a checksum?

- A checksum is a value that can be used to determine if data has been **corrupted or altered**
- It indicates whether data **differs from its original form** but **does not** specify where
- Checksums are **calculated using an algorithm** and the value is **added to the transmission**
- The receiving device **re-calculates** the checksum and **compares** to the original
- If the **checksums do not match**, it is assumed an error has occurred

What is double entry checking?

- Double entry checking involves **entering the data twice** in separate input boxes and then comparing the data to ensure they both match
- If they do not, an error message is shown

What is a visual check?

- A visual check involves the user **visually checking the data on the screen**
- A popup or message then **asks if the data is correct** before proceeding
- If it isn't the **user then enters the data again**