

The Internet and its Uses

IGCSE Computer Science ■ Topic 5

IGCSE Computer Science



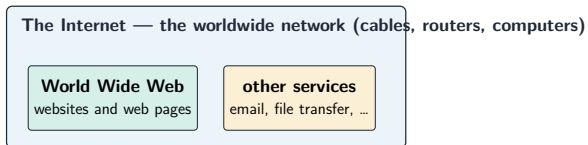
What we will cover

- 1 Internet vs the web
- 2 URL & the browser
- 3 How a page reaches you
- 4 Cookies & blockchain
- 5 Cyber threats
- 6 Keeping data safe





Internet vs the world wide web



- The **internet** is the **infrastructure** – the huge worldwide network of computers and the cables, routers and connections that join them.
- The **world wide web** (WWW) is a collection of **websites** and web pages that you view using the internet.

The internet and the world wide web



A router joins your home network to the internet and sends data to the right place

A URL is an address in three parts

`https://` `www.example.com` `/index.html`

protocol domain name path to the
page on the server

the parts of a URL

A **uniform resource locator** (URL) is a text address you type into a **web browser**. In `https://www.example.com/books/index.html`:

- `https` – the **protocol**, the rules for transferring the page
- `www.example.com` – the **domain name** of the web server
- `/books/index.html` – the file's path on that server

What a browser actually does

rendering

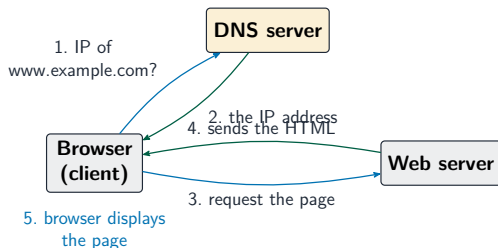
turns the page's **hypertext markup language** (HTML) into what you see

displaying

puts the finished page on the screen and follows the links you click

It also **manages how the page is presented** – layout, text and images – which is why the same page can look different in two browsers.

How a web page reaches you



- 1 you type a URL and press enter – a **name**, not a number
- 2 the browser asks a **DNS** for the IP
- 3 the DNS returns the **IP address**
- 4 the browser contacts the **web server**
- 5 the server sends back the HTML
- 6 the browser renders the page

DNS is the internet's phone book – name in, IP address out.



Cookies

	session	persistent
kept on close? used for	no basket, short-term	yes remembering you between visits

A **cookie** is a small text file a site stores on your device to remember things – logins, preferences, basket contents.

Session cookies die when you close the browser; **persistent** ones survive to recognise you next time.

A cookie is a small text file



a cookie

a small text file
stored on your computer
remembers settings / login

A small text file the site stores on [your](#) computer. It can hold your preferences, the contents of a basket, or [personal details](#) such as a login or address, so you do not type them again – and it can also track what you look at.

What a cookie is for

What it holds

A small text file the site stores in your browser – a session id, a language choice, items in a basket.

Track user preferences

Sites use them to **track user preferences** between visits, so you stay logged in and the settings persist.

The other use

Third-party cookies follow a visitor *between* sites to build an advertising profile – the reason for consent banners.

Privacy settings

Privacy settings in a browser control which cookies are accepted, and clearing them logs you out of everything.

A cookie cannot run code. What makes it sensitive is not what it does, but what it lets a site *remember*.

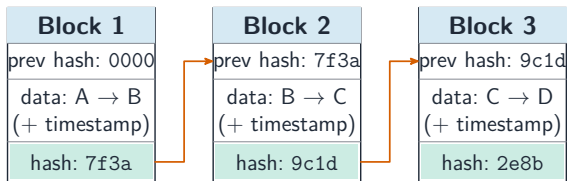
Digital currency

A **digital currency** is money that exists only in **electronic form** .

Stored and moved by computer, and spendable online. The problem is **trust**: what stops someone spending the same money twice, or editing the records? That is the question blockchain exists to answer.

Blockchain

each block stores the previous block's hash



A **blockchain** is a shared **digital ledger** of every transaction. Each block holds the data, a **timestamp**, a **hash**, and the **hash of the block before**.

Change one old block and its hash changes – breaking every block after it. That chained hash is what makes tampering **obvious**.



Cyber threats

brute force

try many passwords fast until one works

data interception

“listen in” to steal data in transit

DDoS

flood a server until it cannot respond

malware

virus, worm, Trojan,
spyware, ransomware

phishing / social engineering – trick a person into giving up secrets

Cyber security locks data and systems against malware, phishing and unauthorised access – the padlock is the whole job in one picture.

The cyber-security threats, by what each does

DDoS

flooding a server with so many requests that it **cannot respond**, so the website goes down

hacking

gaining access to a system **without permission**

pharming

secret code sends you to a **fake website even when you type the correct address**

phishing

fake emails or messages that **trick you** into giving away private details

brute-force attack

trying every password until one works

data interception

reading data as it travels across a network

Phishing needs **you** to click; pharming does not. That difference is why one is beaten by care and the other only by technical defences.

Three attacks, and the words behind them

Brute force attack

A **brute force attack** tries every combination until a password works – which is why length beats complexity.

Distributed denial of service

A **distributed denial of service** (DDoS) attack floods a server from many machines at once, so real users cannot get through.

DNS

The **domain name service** turns a name into an IP address; poisoning it sends visitors to the attacker's server instead.

Hash value

A **hash value** is a fixed-length fingerprint of data. Passwords are stored hashed, so a stolen database does not hand over the passwords.

For each threat name the target, the method and one defence. That triple answers almost every question in this topic.

Malware – know the differences

virus	attaches to a file; spreads when opened
worm	spreads over a network on its own
Trojan	pretends to be useful software
spyware	secretly records what you do
ransomware	locks files, demands payment

The key distinction: a **worm** needs no file and no user action – it copies itself across the network. A **virus** needs a file to be opened.

The malware types

virus

attaches to a file and needs a person to run it

worm

copies itself across a network with **no** human action

trojan horse

disguised as something useful, so you install it yourself

spyware

records what you do – often a **keylogger** capturing what you type

adware

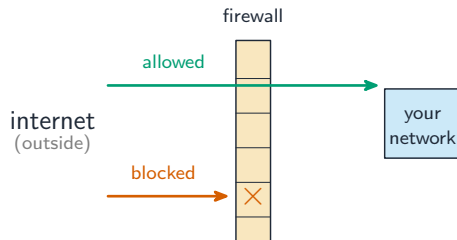
floods you with unwanted adverts

ransomware

encrypts your files and demands payment for the key

These attacks **steal personal data, delete or change files, stop a service working, cost money, and make users lose trust in a company.**

Keeping data safe



- **firewall** : checks traffic in/out, blocks the disallowed
- **authentication** : password, **biometrics** , two-step
- **access levels** : see only what you need
- anti-malware, updates, **proxy server**

Against **phishing** the defence is a person: check the spelling, the urgent tone, and the real **URL** behind a link.

How data is kept safe

firewall

checks traffic entering and leaving, and blocks what breaks the rules

anti-malware

anti-virus and **anti-spyware** – scan for and remove harmful software

authentication

proving who you are: a strong password, **two-step verification**, or **biometrics** (fingerprint or face)

encryption

scrambles data so an interceptor cannot read it – it does not stop theft, it makes theft useless

automatic updates

updates **fix weak points** in software, and automating them means they are not forgotten

reading carefully

bad **spelling** or an **urgent tone** marks a phishing message, and a strange **URL** marks an unsafe link

The last two cost nothing and stop the commonest attack there is. Technical defences cannot help a user who types their password into a fake page.

Worked example – breaking down a URL

`https://www.example.com/books/index.html`

- `https` is the **protocol** – the rules used to transfer the page; the `s` means the transfer is encrypted.
- `www.example.com` is the **domain name**, which identifies the web server and which a DNS server turns into an IP address.
- `/books/index.html` is the **file path and file name** of the resource on that server.

Name each part *with its job*. “The first bit” and “the last bit” describe the URL without explaining anything.

Exam tips

- The **internet** is the worldwide network (the infrastructure); the **world wide web** is the pages you view on it – do not mix the two up.
- Learn how a page loads: the browser asks a **DNS** for the site's IP address, then requests the page from the **web server**, which returns the HTML.
- Match the malware types: **virus** (attaches to a file), **worm** (spreads by itself across a network), **Trojan horse** (pretends to be useful), **spyware**, **ransomware**.
- Match each threat to a defence: phishing/pharming → check the URL; brute force → strong passwords + two-step verification; interception → encryption; hacking/DDoS → a firewall.
- A **session** cookie is deleted when you close the browser; a **persistent** cookie is saved to remember you between visits.



Topic 5 – key takeaways

1 Net vs web

internet is the network; web runs on it

2 DNS

name in, IP address out

3 Blockchain

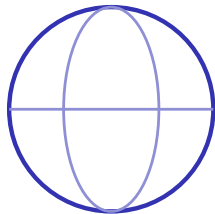
chained hashes make tampering obvious

4 Safety

firewall, authentication, access levels

Check yourself

- 1 What does a DNS turn a URL into?
- 2 Which malware spreads with no user action?
- 3 Which cookie survives closing the browser?
- 4 What checks traffic in and out of a network?



Answers 1. an IP address 2. a worm 3. persistent 4. a firewall