Topic 05: The Internet and Its Uses

IGCSE Computer Science 0478 | SenpaiCorner | Mr. Wei

Learning Objectives

By the end of this topic you should be able to:

- Distinguish between the internet and the World Wide Web; give examples of services on each.
- Describe the components of a URL (protocol, domain name, file path).
- Explain the difference between HTTP and HTTPS; state the role of SSL/TLS.
- Identify the two layers of TLS and describe how a secure connection is established.
- State the purpose and functions of a web browser.
- Describe the step-by-step process by which a web page is located, requested, and displayed (URL $\rightarrow$ DNS $\rightarrow$ IP address $\rightarrow$ web server $\rightarrow$ HTML rendered).
- Distinguish between session cookies and persistent cookies; describe their uses and privacy concerns.
- Define digital currency; describe the key properties of a blockchain/digital ledger.
- Identify and describe cyber-security threats: brute-force attack, data interception, DDoS attack, hacking, malware types (virus, worm, Trojan horse, spyware, adware, ransomware), pharming, phishing, social engineering.
- Describe cyber-security solutions: access levels, anti-malware, authentication methods, automated updates, firewalls, privacy settings, proxy servers, SSL.

5.1 The Internet and the World Wide Web

Internet vs. World Wide Web

The **internet** is the global network of interconnected devices and infrastructure. It provides the underlying communication system using TCP/IP. Services delivered over the internet include email, VoIP, cloud storage, file transfer, and online gaming.

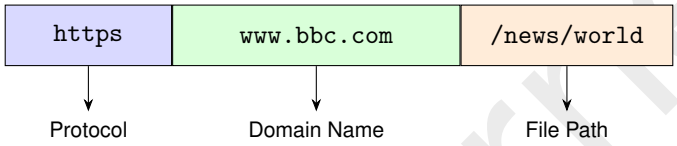
The **World Wide Web (WWW)** is a collection of websites and web pages built *on top of* the internet. It is accessed using web browsers. Visiting a website (e.g. `www.wikipedia.org`) is using the WWW; sending an email uses the internet but not the WWW.

Internet vs. WWW — Do Not Conflate

The internet is the global physical and logical network infrastructure. The WWW is one *service* that runs on top of that infrastructure. Email, VoIP, and cloud storage are internet services that are **not** part of the WWW. Cambridge questions specifically test this distinction — a student who writes "the internet is websites" will lose the mark.

Uniform Resource Locator (URL)

A **URL** is the text-based address that specifies the location of a resource on the WWW. It tells the browser which protocol to use, where to find the resource (domain name), and which specific file or resource to retrieve.



- **Protocol** — specifies the rules of communication (e.g. `https`, `http`).
- **Domain name** — the address of the web server (e.g. `www.bbc.com`).
- **File path** — the location of the specific resource on the server (e.g. `/news/world`).

Without URLs, the WWW would require users to memorise numeric IP addresses. URLs also appear in hyperlinks, APIs, and email `mailto:` links.

HTTP and HTTPS

HTTP (Hypertext Transfer Protocol) is the set of rules for transferring web pages between a browser (client) and a web server. Data is sent as plain text and can be read if intercepted. HTTP operates on port 80.

HTTPS (Hypertext Transfer Protocol Secure) adds encryption to HTTP using SSL/TLS. It operates on port 443. HTTPS provides:

- **Confidentiality** — data is hidden from eavesdroppers.
- **Integrity** — data cannot be tampered with during transmission.
- **Authentication** — the website's identity is confirmed via a digital certificate.

HTTPS is essential for banking, online shopping, and login pages. The padlock icon in the browser address bar indicates an HTTPS connection.

HTTP vs. HTTPS — Quick Recall

HTTP = plain text, port 80, not secure. **HTTPS** = encrypted, port 443, secure. For a 2-mark question: state the difference (plain vs. encrypted) *and* give a consequence (data can / cannot be intercepted by a third party). Students can verify HTTPS by checking for the padlock icon or the `https://` prefix.

Transport Layer Security (TLS)

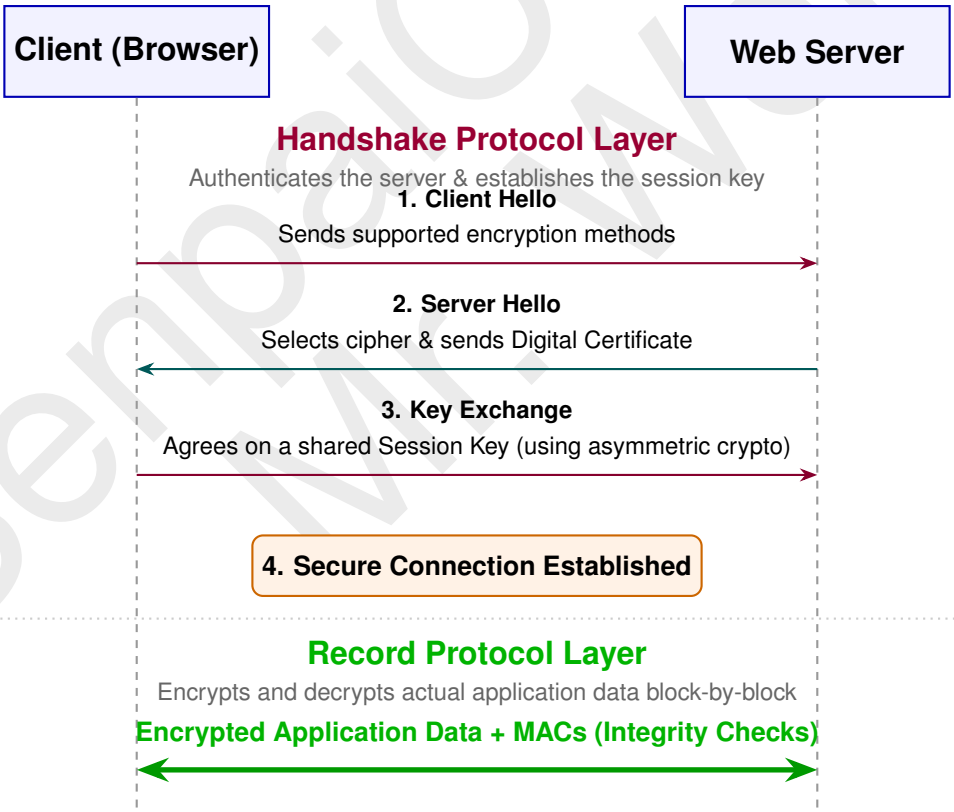
TLS (Transport Layer Security) is a security protocol that ensures safe communication over the internet. It is the modern successor to SSL. TLS provides three guarantees:

- **Encryption** — hides data from eavesdroppers.
- **Authentication** — verifies the server is genuine via digital certificates.
- **Integrity** — ensures data is not altered during transmission.

TLS has **two layers**:

TLS Layer	Function
Handshake Protocol	Runs *before* secure communication starts. Authenticates the server using a digital certificate, agrees on the encryption method (cipher), and establishes a shared session key.
Record Protocol	Handles the actual encryption and decryption of application data. Splits data into blocks, encrypts each block, and adds integrity checks (Message Authentication Codes — MACs).

TLS Handshake — four steps (simplified):



1. **Client Hello** — the browser sends its supported encryption methods to the server.
2. **Server Hello** — the server selects an encryption method and sends its digital certificate.
3. **Key Exchange** — the client and server agree on a session key using public/private key cryptography.
4. **Secure Connection Established** — all further communication is encrypted using the session key (Record Protocol takes over).

TLS Exam Recall — Two Layers

Cambridge past papers regularly ask: "Identify the two layers of TLS." The answer is always **Handshake** and **Record**. *Handshake* = authentication and key setup. *Record* = encrypts actual data in transit. For a 5-to-6-mark "explain how data is sent securely using TLS" question, cover: certificate request, certificate sent, certificate validated, signal back to server, session key generated, data encrypted — both layers must be implied.

Web Browser

A **web browser** is software that renders HTML into visual web pages (e.g., Chrome, Edge, Safari, Firefox).

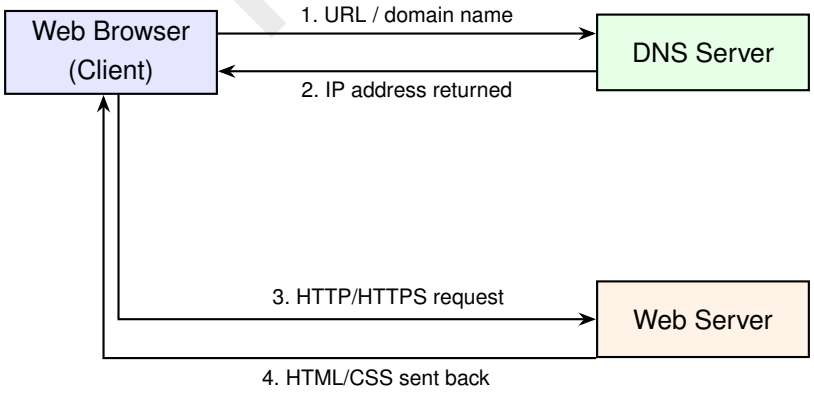
Functions of a web browser:

- Send a URL/domain name to a DNS server.
- Request web pages from a web server IP.
- Receive web page data (HTML, CSS, JS).
- Convert/render HTML to display the page.
- Manage protocols (HTTP/S) & verify SSL.
- Store bookmarks and favourites.
- Record a history of visited sites.
- Allow multiple tabs.
- Store cookies.
- Provide navigation tools and address bar.
- Run client-side scripts (JavaScript).
- Allow files to be downloaded.

How Web Pages Are Located, Retrieved and Displayed

When a user types a URL into a browser, the following sequence occurs:

1. The browser sends the URL/domain name to a **DNS (Domain Name Server)**.
2. The DNS searches its database and returns the corresponding **IP address**. If the URL is not found, the request is forwarded to another DNS.
3. The browser sends an **HTTP/HTTPS request** to the web server at that IP address.
4. The web server locates the page and sends back **HTML, CSS, and JavaScript**.
5. The browser **renders the HTML** to display the web page on screen. Any SSL/TLS certificates are exchanged and authenticated before data transmission begins.



Webpage Request — Worked Example

A student types `www.youtube.com` into Chrome. Chrome sends the domain name to a DNS server. The DNS returns the IP address (e.g. 142.250.190.14). Chrome sends an HTTPS request to that IP address. The YouTube web server sends back HTML, CSS, and JavaScript. Chrome renders this data and displays the homepage. SSL/TLS certificates are exchanged and authenticated before any data is transmitted.

Missing the DNS Step

A common exam error is to write: "the browser sends the URL directly to the web server." This is incorrect. The URL first goes to the **DNS server** to be translated into an IP address. Only then does the browser contact the web server. Always include the DNS step in any "describe how a web page is requested" question.

Cookies

A **cookie** is a small text file sent by a web server and stored on the user's device by the browser.

Type	Storage location	Lifespan
Session cookie	Stored in RAM/memory	Deleted when the browser is closed
Persistent cookie	Stored on the hard drive/secondary storage	Remains until expiry date or user deletes it

Uses of cookies:

- Save personal details (name, email, payment details).
- Track user preferences (e.g. theme, language).
- Store login details ("remember me" feature).
- Keep shopping cart items between visits.
- Targeted advertising based on browsing habits.

Privacy concerns:

- Users may not know what data is collected, so they may feel their privacy is affected.
- A profile could be built about the user, potentially leading to identity theft.
- Sensitive data stored in cookies could be intercepted during transmission.
- Other websites or hackers could gain access to cookies stored on the device.
- Payment information could be stolen and used by a third party.

Cookies Do Not Contain Viruses

Cookies are plain text files. They cannot execute code and cannot carry viruses. However, cookies *can* store sensitive data that may be exploited if intercepted or accessed by an unauthorised party. This is a frequently tested misconception in IGCSE papers.

Cookies — How Payment Details Are Stored and Auto-entered

Mark scheme pattern for a 3-mark "describe how cookies store and auto-enter payment details" question:

- The web server sends a cookie file to the user's browser.
- The user's payment details are stored (encrypted) in the text file on the hard drive/secondary storage.
- When the user revisits the website, the web server requests the cookie file and the browser sends it back, allowing payment details to be automatically entered.

5.2 Digital Currency

Concept and Uses

Digital currency is a currency that exists only electronically — it has no physical form.

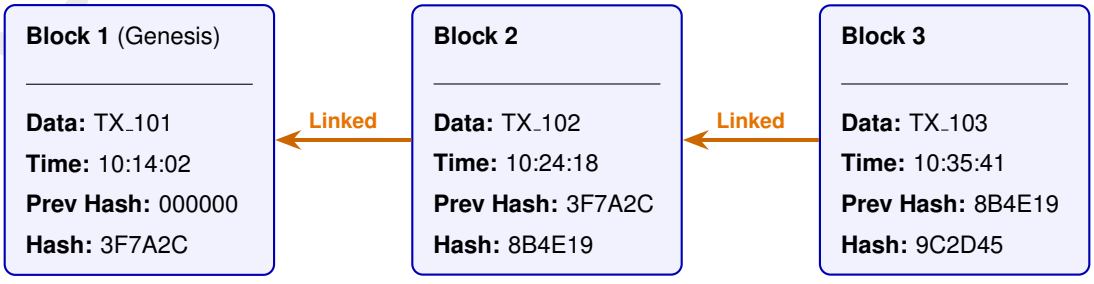
Type	Examples	Notes
Cryptocurrency	Bitcoin, Ethereum	Decentralised; not government-backed
E-wallet / platform	PayPal, Apple Pay	Centralised payment platforms
CBDC	Digital pound, digital yuan	Centralised; government-backed

Uses of digital currency:

- Online shopping (e.g. Shopee, Amazon).
- Peer-to-peer transfers (e.g. PayPal, Bitcoin wallet to wallet).
- In-game purchases (e.g. Roblox Robux, Fortnite V-Bucks).

Blockchain

A **blockchain** is a distributed digital ledger used to record transactions. Copies of the ledger are stored across many computers (nodes) in a network. It is **decentralised** — no single authority controls it.



Immutability / Tamper-Proofing:
If data in Block 2 is altered, its Hash (`8B4E19`) instantly changes.
Block 3 will no longer match the chain, alerting all nodes in the decentralized network.

Each **block** in the chain contains:

- Transaction data (sender, receiver, amount).

- A timestamp (when the transaction occurred).
- A cryptographic hash — a unique digital fingerprint of the block's contents.
- The hash of the *previous* block, linking all blocks into a chain.

Key properties:

- **Immutability** — records cannot be changed once added to the chain.
- **Transparency** — all network participants can view the transactions.
- **Security** — cryptography prevents tampering.
- **Trustless** — no need for banks or central intermediaries.

New transactions are verified by network participants before a block is added. Once added, it cannot be altered. Even if one node is compromised, the record survives on all other nodes.

Blockchain — Exam Definition Sentence

Blockchain = digital ledger + decentralisation + immutability + timestamps. For a 1-mark "give the name of this process" question — "a digital ledger with time-stamped transactions that cannot be altered" — the answer is simply **blockchain**. For a "state what is meant by digital currency" question — the answer is: "a currency that only exists electronically/virtually."

5.3 Cyber Security

Cyber-Security Threats

Threat	Description	Example
Brute-force attack	Trial and error to guess a password or encryption key; all combinations are repeatedly entered until the correct one is found. Can be carried out manually or automatically by software.	Trying thousands of password combinations per second to access a Gmail account.
Data interception	Unauthorised monitoring/capture of data as it travels across a network. Particularly common on unsecured public Wi-Fi.	Sniffing login credentials at a coffee-shop Wi-Fi hotspot.
DDoS attack	Many compromised devices (a botnet) simultaneously flood a server with requests, overloading it until the website becomes unavailable. No data is stolen; the aim is disruption.	An e-commerce site crashing during a major sale due to an attack.
Hacking	Gaining unauthorised access to systems or data in order to steal, view, alter, delete, or encrypt it.	Breaking into a bank database to steal customer records.
Virus	Malicious code/software that replicates itself when the user runs an infected file. Can corrupt or delete data and fill up storage space.	A Word document containing a macro virus.

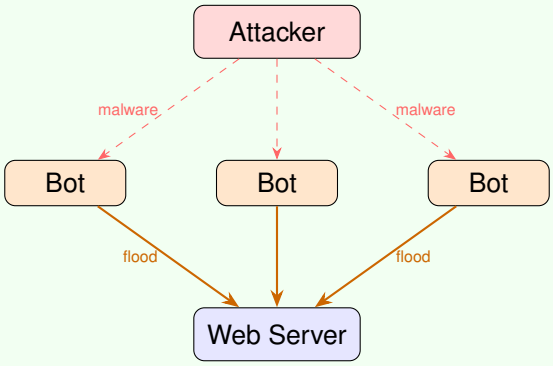
Threat	Description	Example
Worm	Self-replicating malware that spreads across networks *without user action*. Takes up bandwidth; can open back doors and install other malware.	The ILOVEYOU worm spreading through email contacts automatically.
Trojan horse	Software/code hidden within apparently legitimate software. When downloaded or installed, the concealed malware is also installed.	A "free game" download that secretly installs spyware.
Spyware	Secretly records all key presses and transmits them to a third party, who analyses the data to find passwords and login details.	A keylogger capturing online banking credentials.
Adware	Generates and displays unwanted adverts; can reduce device performance, redirect internet searches, or contain other malware.	Pop-up ads appearing constantly after installing free software.
Ransomware	Stops a user accessing their computer/data by encrypting the files; a fee must be paid to decrypt/restore access.	WannaCry ransomware attack (2017) hitting hospitals and companies worldwide.
Pharming	Malicious code downloaded onto the user's computer redirects them to a fake website even when they type the correct URL.	Typing `paypal.com` but being sent to a fake look-alike site that steals login details.
Phishing	A legitimate-looking fake email is sent to the user; the user clicks a link and is directed to a fake website where they enter personal data that is stolen by a third party.	An email "from a bank" asking users to "verify" account credentials via a link.
Social engineering	Manipulating / deceiving / tricking people — rather than systems — to obtain data or force them to make an error. Exploits trust, fear, or curiosity.	A scammer posing as IT support calling employees to ask for their passwords.

Pharming vs. Phishing — The Key Difference

Both are designed to steal personal data and both direct the user to a fake website. The difference: **phishing** requires the user to follow a link in a fake *email*. **Pharming** uses malicious code *already installed on the user's computer* to redirect them from the correct URL they typed. Cambridge past papers test this with "give one similarity and two differences" — a common 3-mark format.

DDoS Attack — How It Works

A hacker sends malware to many unprotected computers, turning each one into a **bot**. The collection of bots is a **botnet**. When the hacker initiates the attack, all bots simultaneously send repeated requests to the target web server. The server becomes flooded with traffic, cannot handle the requests, and fails. Legitimate users can no longer access the website.



Server overloaded — website becomes unavailable

Cyber-Security Solutions

Solution	Description	Example
Access levels	Users are assigned permissions based on their role; they can only read or edit data they need. Limits the risk of insider attacks and reduces the impact of a compromised account.	Students cannot alter admin records on a school server.
Anti-malware software	Detects, blocks, and removes malware (viruses, spyware, ransomware). Must be updated regularly to recognise new threats.	Windows Defender alerting to a Trojan horse download.
Authentication methods	Verifies user identity before granting access. Methods: username/password (basic), biometrics (fingerprint, face — hard to fake), two-step verification/2FA (password plus a one-time code sent by SMS or app).	Online banking login requiring password and an SMS code.
Automated software updates	Keeps the OS and applications up to date, patching known vulnerabilities before attackers can exploit them.	Windows Update installing the latest security patches.
Check email spelling/tone	Phishing emails often contain poor grammar, unusual urgency, or misspelt sender addresses. Recognising these signs prevents falling for the scam.	Spotting "PayPall" instead of "PayPal" in a sender address.

Solution	Description	Example
Check URL links	Verify the site uses HTTPS and that the domain spelling is correct before entering personal data. Look for the padlock icon.	`amazon.com` is genuine; `amaz0n.com` is fake.
Firewalls	Hardware or software that monitors and filters incoming and outgoing network traffic against defined criteria (whitelist/blacklist). Can be hardware or software. Keeps a log of traffic and blocks unauthorised access attempts.	A company firewall blocking port scans by an external hacker.
Privacy settings	Restrict who can view or share personal data on online platforms.	Limiting social media posts to "Friends only."
Proxy servers	Act as intermediaries between users/clients and the internet. Can hide IP addresses, cache web pages for faster access, filter traffic by whitelist/blacklist, and absorb a DoS/DDoS attack before it reaches the real web server.	A school proxy server blocking access to gaming sites.
SSL	Security protocol that establishes an encrypted connection between browser and web server using asymmetric encryption and digital certificates. Browsers request the SSL certificate, validate it, and send a signal back before data transmission begins.	Padlock icon on an online shopping checkout page.

Threat-to-Solution Mapping — Exam Technique

Always connect a specific threat to appropriate solutions. Examiners reward both *technical* and *human-behavioural* measures:

- **Brute-force** → strong/complex password; two-step verification; limit login attempts; biometrics.
- **Phishing** → check URL links; check email spelling/tone; never click unsolicited links.
- **Malware** → anti-malware software; automated updates; download only from trusted sources; firewall.
- **DDoS** → proxy server; firewall; users scan computers with anti-malware.
- **Data interception** → SSL/TLS; HTTPS; encryption.

Firewall vs. Proxy Server — The Examiner's Distinction

Similarities: both filter/check traffic; both can maintain a whitelist/blacklist; both can be hardware or software; both block unauthorised access; both keep a log of traffic. **Key differences:** A **proxy server** can *hide the user's IP address* and can *cache* web pages for faster access — a firewall does neither. A proxy server absorbs a DoS/DDoS attack before it reaches the web server; a firewall checks packet types but does not route requests on behalf of the server.

SSL Certificate Check — How a Browser Verifies a Secure Website

Mark scheme pattern for a 4-mark "describe how a browser checks a website is secure" question:

- The browser requests that the web server identify itself / requests to view the SSL certificate.
- The web server sends a copy of its digital certificate to the browser.
- The browser checks whether the certificate is authentic / trustworthy.
- If valid, the browser sends a signal back to the web server to confirm, and encrypted data transmission begins.

Social Engineering Is Not Hacking

Social engineering exploits *people*, not computer systems. It does not involve breaking into or cracking a computer. A student who writes "social engineering involves hacking into a computer" will not score the mark. The correct description: manipulating, deceiving, or tricking people to obtain data or to force them to make an error.