Introductory Lesson

Understanding http: the postcard problem

Browser sends info like a postcard that anyone can read

Hackers use tools like burp suite to See or change info

https enforces security making sealed envelopes

Defensive http response headers

Sites like twitter add special rules like only open certain file types or only open within current site

Stay within the platform (no redirecting) because the environment is controlled by the platform keeping users safer

Login pages: bank vault door

Hackers can try endless passwords (brute force)

Guess usernames

Steal session tokens and log in as you

Access controls across layers

UI – hide delete user buttons from regular staff

Code – check If user is actually admin before executing commands

Database – ensure users only see their own data

What is XSS (cross site scripting)

Website unknowingly delivers malicious code to users

How apps block XSS

Output encoding: emojis and symbols etc turn to code to show but code wont run

Avoid raw user data in scripts

Sanitization removes harmful code

CSRF

Website can’t tell real requests from forged ones so it accepts both

Session tokens used to mask as you

Defences to CSRF

Anti-CSRF tokens are secret codes in each request

X frame options are used so no one can frame checkout pages

Critical questions like are you sure are prompted

Cryptography

A diagram of a computer script

AI-generated content may be incorrect.SQL injection

HTTP Security Considerations

* Proxy Server Settings: proxy sits between user and internet which filters, monitors and caches content; misconfigured proxy can allow attackers to slip through

A diagram of a server

AI-generated content may be incorrect.

* Without a proxy users can reach harmful sites and attackers can scan networks
* Correctly configured proxy keeps apps controlled and secure

Threat of Untrusted Data

* Apps get data from forms, cookies, URLs and uploads where attackers can hide harmful commands and scripts in inputs

Examples of Risks:

* XSS: scripts hidden in comments
* SQL Injection: harmful queries via input
* Database Inputs: data from users can be tainted

Threat: an attack that occurs (someone breaks in)

Risk: the action executed (someone stealing)

Vulnerability: weak spot (not locking your door)

How XSS attack works

A diagram of a cross-site scripting attack

AI-generated content may be incorrect.