Why Secure Programming Matters

Week 1, Lecture 2 — COMP09031

1

From bug to breach

TalkTalk — October 2015

- ~156,959 customer records exfiltrated; 15,656 with bank sort code + account number.
- Cost: ~£77m to the business; £400k ICO fine (then a record).
- Technical root cause: SQL injection on three legacy pages.
  - Inherited from the 2009 Tiscali UK acquisition.
  - Database: unpatched MySQL with a fix available since 2012.
  - Two earlier successful injection probes (July, September 2015) had gone undetected.
- ICO verdict read like a syllabus checklist: no inventory, no monitoring, no patching, no parameterised queries.

The bug-to-breach pipeline

Developer decision → unmonitored vulnerability → undetected probe → ICO report.

Every stage is software behaving as written.

British Airways — August 2018

- 22 lines of skimming JavaScript injected into Modernizr 2.6.2 on ba.com.
- Ran undetected from 22 August to 5 September 2018.
- ~429,612 individuals affected; ~244,000 had full card + CVV stolen.
- ICO fine: £20m (down from initial £183m proposal due to mitigation + COVID-19 hardship).
- Tell-tale: a single network call to baways.com after each successful payment.

MOVEit Transfer — May 2023

- Cl0p ransomware group exploited a previously-unknown SQL injection in Progress Software's MFT product.
- ~2,500 organisations and ~93 million individuals affected.
- Vulnerable component: a single API DLL, moveitisapi.dll.
  - Crafted action=m2 + x-silock-transaction header concatenated into SQL.
  - Custom ASP.NET web shell LEMURLOOT dropped for persistence.
- Same shape, eight years on. SQL injection is still the unsanitised string concatenated into a query.

2

The numbers, briefly

What the 2025 reports say

- Verizon DBIR 2025 — vulnerability exploitation as initial access doubled year-on-year.
- Mandiant M-Trends 2025 — global median dwell time keeps falling, but that's because attackers are encrypting faster.
- IBM Cost of a Data Breach 2025 — global average ~$4.88m; healthcare and finance worst hit.
- Cost-of-fix curve.
  - Boehm 1981 is the original; many later citations are unreliable.
  - Don't quote a clean 100× number — quote a range.
  - The Register / Bossavit critique is real; flag it verbally.

Embedded interlude — OWASP Top 10 movement

3

Why good developers write insecure code

Three studies that should have changed practice

- Acar et al., IEEE S&P 2016.
  - Information sources matter — copy-paste-from-Stack-Overflow is statistically less secure than copy-paste-from-official-docs.
- Fischer et al., IEEE S&P 2017.
  - 1,161 Android apps had Stack Overflow code with known security flaws — 15.4% of the apps studied.
- Jallow et al., IEEE S&P 2024.
  - Stack Overflow snippets evolve; copies in deployed software don't follow the upstream fixes.
- Pattern: developers reach for fast answers; security is a non-functional concern that loses to deadlines.

4

The AI coding assistant question

Four years of evidence, all uncomfortable

- Pearce et al. 2022 (IEEE S&P, Distinguished Paper).
  - 89 scenarios drawn from CWE Top 25, 1,689 generated programs.
  - ~40% vulnerable.
- Perry et al. 2023 (ACM CCS).
  - 47 developers; AI-assisted half wrote LESS secure code AND were MORE confident in it.
- Snyk 2024 AI Code Security Report (n=537).
  - 75% of devs believed AI code was MORE secure than human code.
  - 80% admitted bypassing org security policies to use AI tools.
- Veracode 2025 GenAI Code Security Report.
  - 100+ LLMs tested across Java/Python/C#/JS — 45% of samples introduced an OWASP Top 10 bug.
  - Security performance was FLAT across model sizes. Bigger ≠ safer.

Healthy scepticism, not abstinence

Perry 2023's positive signal: the developers who engaged with their prompts — rephrasing, reading critically, treating the assistant as a junior pair — produced code with fewer vulnerabilities.

5

Cultural change is possible

Two cultural inflections worth remembering

- Trustworthy Computing — January 2002.
  - Bill Gates internal memo to all Microsoft FTEs.
  - 'Choose security over features when forced to choose.'
  - Two-month dev pause; the Security Development Lifecycle followed.
- Heartbleed and the Core Infrastructure Initiative — April 2014.
  - OpenSSL vulnerability that exposed memory of millions of TLS-protected services.
  - Catalysed industry-funded support for critical OSS — the OpenSSF eventually grew out of this.
- Both started with one team that had been embarrassed.

What's next

- Practical (W1 P): Build your lab — Docker, Python venv, git, VS Code, the legality of the toolkit.
- W2 L1: Threat Modelling — STRIDE on a real DFD.
- Reading: Chapter 2 of the book, plus the Pearce 2022 paper if you want the methodology depth.