2. Security Literacy and Thinking

COMP6441 • KC Notes

2.1 Cyber Security Literacy

- **Reconnaissance** (recon) – gathering information and learning about a target
- **Active** recon: seeking information that **can be detected or identified** by the target
    - Engages with the target to get more information
- **Passive** recon: collecting information **without engaging** with the target
- The **little things mean a lot** – knowing the coffee shop nearby, overhearing phone calls can slowly open up to larger amounts of information
    - Other examples: dumpster diving, paper shredding, burning, hard drive disposal
    - All these can be reassembled and reconstructed

2.2 Think like an Engineer: Problems

- Hacking will be around forever. Why?
    - **Complexity** – misuse and abuse of unexpected and out of spec behaviour
        - The **Aneroid barometer task** – when asked to measure the height of a building "using an aneroid barometer", you could easily sell the barometer and buy a piece of string, or time and drop the barometer!
    - **Asymmetry** – attack one component, defend all components (Week 1)
    - **Weakest Link** – when one link breaks, the whole security breaks
    - **Hubris** – companies think like a defender (Week 1)
    - **Abuse of trust, human weakness** (Week 3), **psychology**
- **M&Ms** – companies believe that if you have one big protection system on the outside, everything inside is secure
    - That barrier is like an M&M – a brittle crusty layer that becomes **a single point of failure**
    - Difficulties in defining the boundary
    - Does not prevent attacks from inside
    - Examples: internal access points that bypass a firewall
- **Secrets** are similarly a single point of failure
- **Kerckhoff's Principle**: A cryptosystem should be secure **even if everything except the key is public**.
- **Security through obscurity** and **security theatre** – making things *seem* complicated and difficult

2.3 Think like an Engineer: Solutions

- To build a secure system:
    - Make sure the system is checked by others and yourself
    - **Ask the right questions** and figure out what you want to achieve first.
        - Sun Tsu: A poor general goes into battle and wants to win. A good general **plans how he is going to win without going to battle**.
        - Most systems have a lack of process and is cunning-less
    - Don't rely on obscurity as it is brittle.
- **Type I and Type II errors** – false positives and false negatives
    - When something is actually true but the **test says it is false**, or something is actually false but **the test says it is true**.
    - One of these situations is usually worse, but when we reduce the error of one we increase the error of the other. The best solution is **to reduce overall error**, but it is typically difficult.
    - Examples: making it easier/harder to get the dole/social services, jail, refugees entry/refusal, biometrics in airports

2.4 Crypto Literacy

- **Crypto literacy**: coming up with cryptography systems that work like magic
- **Protocols:**
    - **Confidentiality**: everyone can feel and see the system but only one person can do something with it, e.g. Japanese puzzle boxes
    - **Integrity**: messages cannot be tampered with
    - **Authentication**: how you know the message came from the owner
- **Primitives**: the building blocks to ensure CIA.

2.4 Confidentiality

- **Cipher**: a secret way of writing to ensure confidentiality
    - **Steganography**: hiding the fact that it is a message, e.g. tattooing a message onto a slave's head, pin pricks in a newspaper.
        - Easy to find, key can't be changed
    - **Substitution cipher**: replacing a letter with another (NSA app)
        - Frequency analysis and patterns
    - **Transposition cipher**: keeping the letters but rearranging their position (Rail fence)
        - Frequency of specific letters
    - **Old codes**
- **New codes** are judged by **entropy** – the amount of chaos in something
    - Considering pairs of letters – 26^2 possible pairs of letters
    - Bits of security – amount of work to brute force a cipher, usually best when the universe runs out (Week 3)