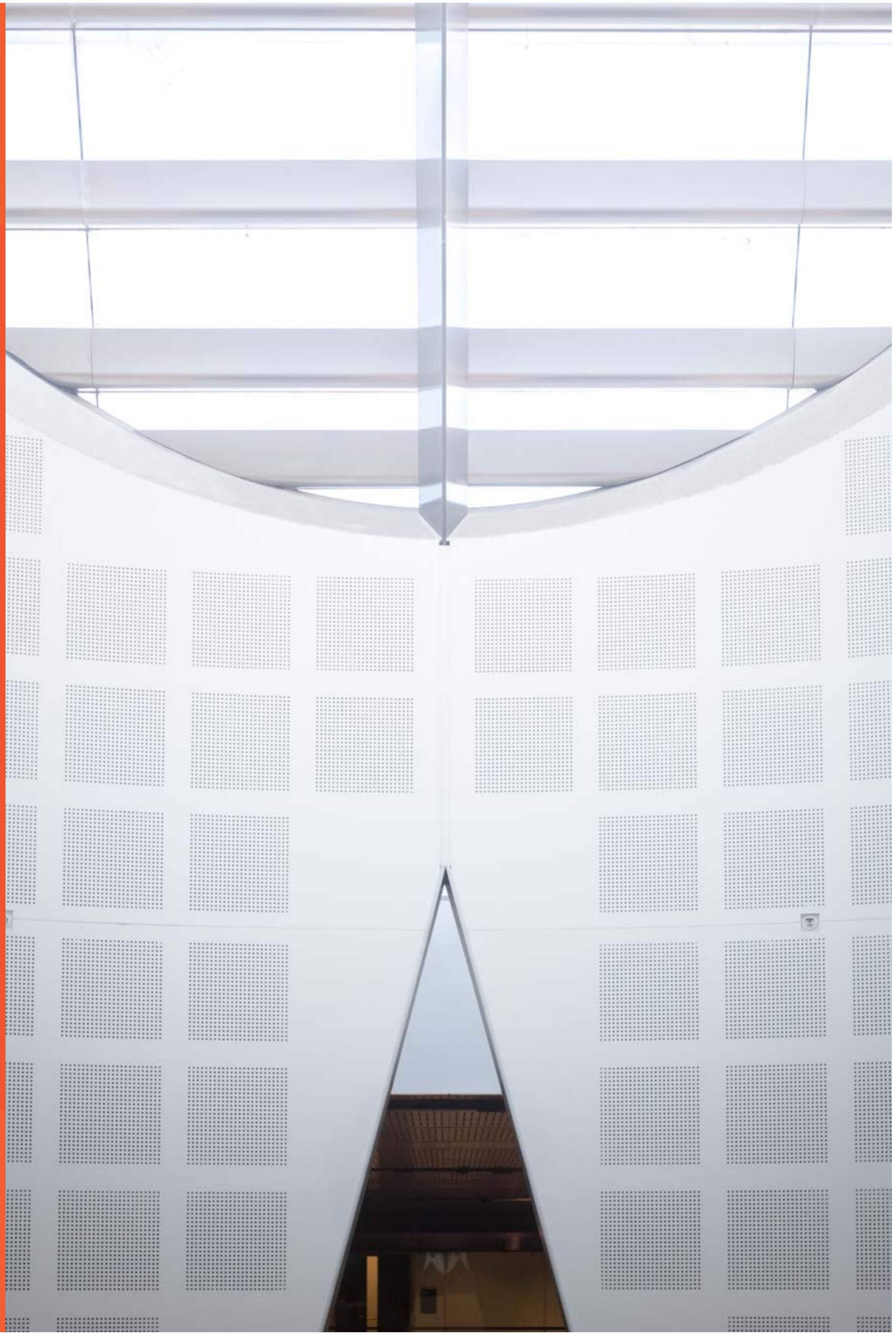
INFO2222

Security Mindset, Terminology, Threat Models

Presented by

Luke Anderson

THE UNIVERSITY OF SYDNEY



Overview

Today's agenda

- Defining security
  - High level
  - Achieving a security mindset
  - Mapping the security space
  - Systems – people and technology

- Critical terminology and security goals

- Threat modelling
  - Context is always key
  - Focusing limited resources

Defining Security

High level

- We want to build and maintain systems that remain safe to use and dependable in the face of malice, error, and mischance.

- Security is often described using military or game-playing terms.
  - Arms race between attacker and defender
  - Resource game – raising the bar high can deflect or deter attacks
  - Attack/offence vs. defence

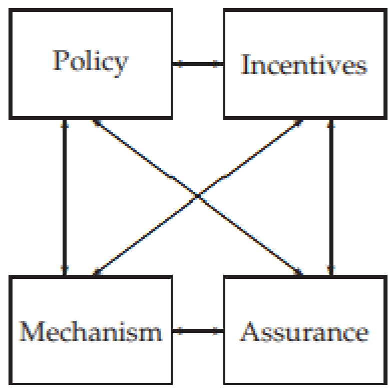
- Nuance:
  - (Software) Engineering: “making things happen”
  - Security Engineering: “making sure certain things do not happen”

Mindset for achieving security

- Understand the big picture, but…
  - Also understand specifics in complex systems.
  - Know that "the devil is in the detail"
- Requires you to
  - Think like an attacker.
    - What would you attack? How? Why?
    - What is the end game? Money? Infamy?
  - Think like a user.
    - Want to get a job done without computer system being in the way
    - Poor usability stands in the way – leads to things such as users clicking through complex warnings…
- Security by design, not bolted on afterwards

Thinking about security (cf. R. Anderson)

- Good security engineering requires four key areas:
  - Policy - What you are supposed to achieve.
  - Mechanisms - How you implement policy
    - Technical controls, cryptography, operational security, etc.
  - Assurance - Amount of reliance you place on a control.
  - Incentive - How to motivate those following policy.



Systems

- The day-to-day operations of society depend on systems where **people** use **technology** to perform *activities*.
  - Businesses, non-profits, governments, individuals, ...

- Definition of system: many things. Need to define precisely before discussing what is to be achieved!
  - **Product or component –** can be software or hardware
  - Collection of the above, **plus operating system, communication, anything that belongs to an organization's infrastructure**
  - The above, **plus applications** (browsers, accounting software, etc...)
  - Any, or all, of the above **plus IT staff**
  - Any, or all, of the above **plus internal users and management**
  - Any, or all, of the above **plus external users and customers...**

Security of Systems

- Security is about keeping systems working as intended.
  - Failure could endanger lives: planes, power plants, etc.
  - Failure could erode societal stability: banking, insurances, ...
  - Failure could destroy a life: identify theft, etc.

Technology in a system

- Hardware
  - Processing
  - Storage
  - Peripherals
- Software
  - Operating systems
  - Files and databases
  - Middleware
  - Applications
- Networks
  - The interconnection of computer systems

People in a system

- Security depends on **people** and their **behaviours**, maybe even more than on technology.
- Designers and developers.
  - Internal and external
  - Vendors of software and hardware.
- Operators and administrators.
- Users
  - Inside the organisation
  - Outsiders (clients, suppliers, partners)
- Other stakeholders
  - People about whom data is kept and society more broadly

Forgetting people is a recipe for failure

- People are not machines.
  - Intrinsic and extrinsic goals: get job done, be liked, self-interested
  - Venue for Social Engineering!
- People don't always do what they are told.
  - Forgetful – "what did you eat 8 days ago?"
  - Instructions may not align with their motivations and goals
  - Intentional maliciousness – insiders can be most damaging, they have the keys.
- We all differ in many ways:
  - Aptitudes
  - Attitudes
  - Engagement
  - Training
  - Priorities

Risk Management

- Security can be viewed through the lense of **risk management**
  - understanding the assets – and risks to the assets.
- Pragmatic, approximate approach:
  - Multiply the quantifiable amount of the potential loss by the probability that the loss will occur.
  - Compare against the costs of security measures to protect against the risk.
- This is called **risk analysis**, which can be applied at the level of the individual, the enterprise, the nation…
- Often highly problematic:
  - What is the probability that a loss will occur? How was this computed?
  - **Always** ask such questions – hard data is better than gut feeling because **human psychology is very bad at estimating risk**
  - **FUD: Fear, Uncertainty, Doubt – clouds thinking**

Common adages

- Security through obscurity does not equal security.
  - Hiding something **does not make it safe – only harder to find**
  - **Always** assume attacker **can find** it, **never ever rely on it** as a defence.
  - Intention behind hiding is to cause the attacker some **frustration**; maybe they will go away before trying anything else. But who knows?

- Security is a trade-off. Nothing is 100% secure.
  - Only ever raising the bar of cost, time, money for attacker.
  - New discoveries every day, tomorrow may bring down your wall.
- Defence in depth:
  - Avoid the eggshell model – one hard layer, gooey insides.
  - Layer controls such that failure of a single control is **not** a full collapse of a system.

Critical Terminology and Security Goals

Key Terms

Vulnerability

Flaw

Threat

Attack

Control

Trusted

Trustworthy

Key Goals

Confidentiality

Integrity

Availability

Auditability

Anonymity/Pseudonymity

Non-Repudiation

Vulnerability

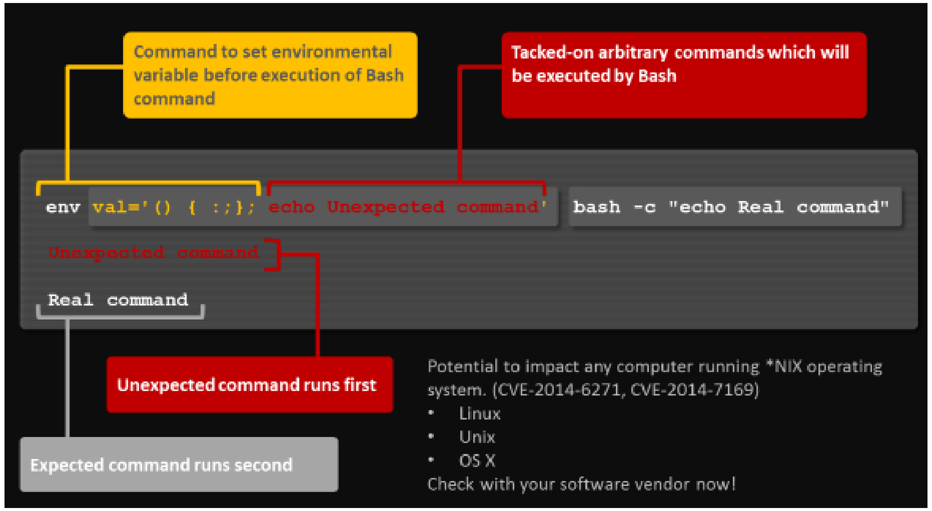
- Weakness in system at **implementation** level.
  - Hardware, software, data, people, …
  - Often due to coding, procedures, or to people's practices.
- Vulnerabilities may be known or unknown.
  - Vendors may be able to address them in patches – need to keep systems updated
  - Patches can also break interoperability – careful checks needed!
  - Industry has disclosure processes – responsible disclosure
  - Zero Days: 0 time between disclosure and use in the wild
- Classic example: buffer overflow
- War Stories: Heartbleed and Shellshock

Heartbleed

- Vulnerability in the OpenSSL **implementation** of TLS.
  - Transport Layer security is responsible for the padlock icon in the browser – ensures that communication between the computer and server is encrypted.
  - Bug introduced into the code in 2012, disclosed in 2014.
  - Due to the lack of a bounds check:
    - Allowed an attacker to receive data from memory they were not supposed to see.
    - “Read” beyond what they were supposed to.
    - This could mean any secret information of the server.
      - Keys? ✓
      - Passwords? ✓
      - User data? ✓
      - ☠ ✓

Shellshock

– A vulnerability in the **implementation** of Bash.
  – **Bash:** Unix shell on most linux distributions, macOS and today, Windows 10 ☺
  – Allowed remote code execution:
    • This means an external attacker can run any code they wish on the machine.



Flaws

Weakness in the system at a **design level.**

- Hardware, software, data, people, …
- Due to design decisions
- These **cannot always be addressed** after shipping the product
- Examples:

Spectre & Meltdown

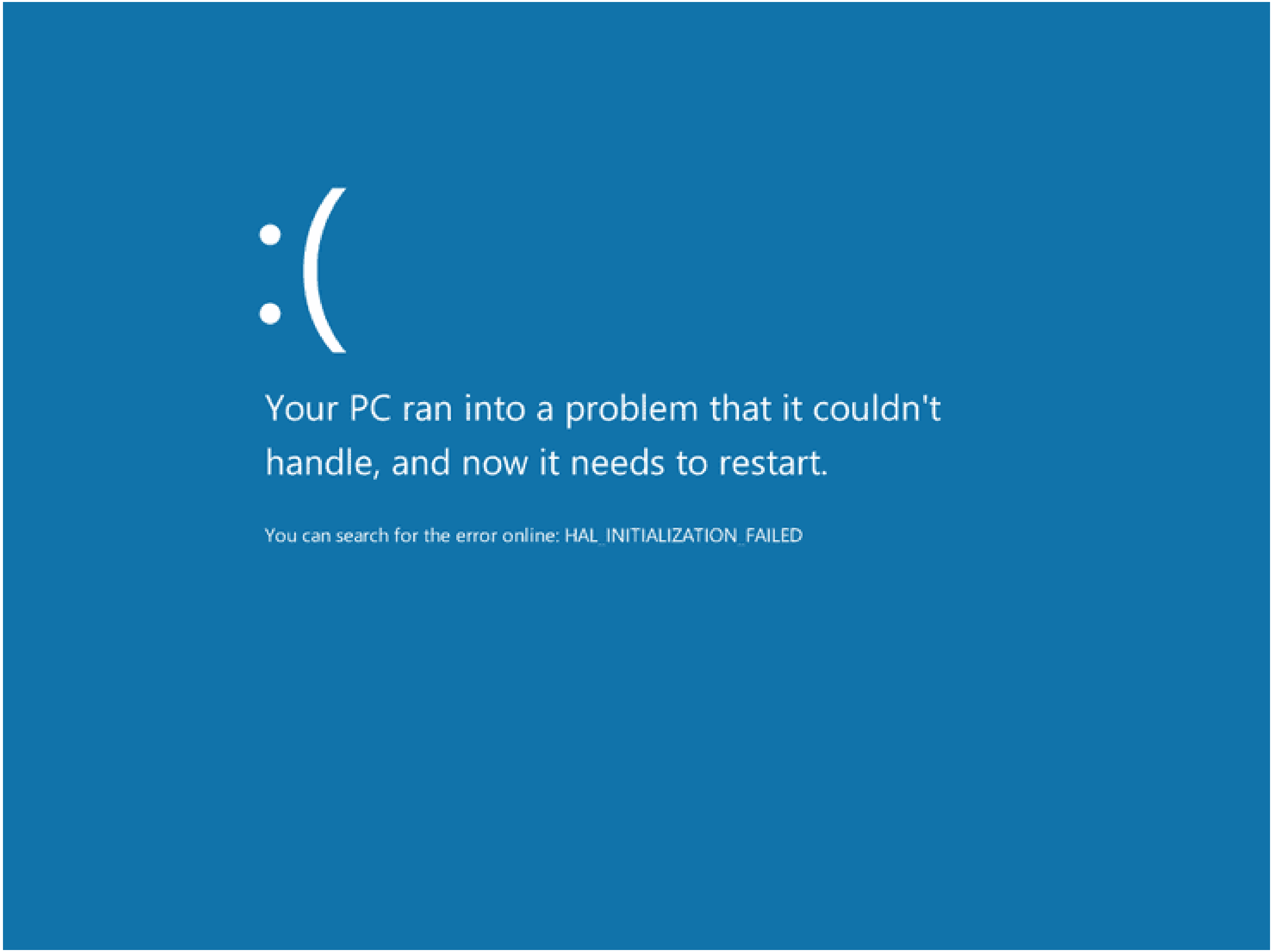
- Fundamental **design** issues with nearly every computer chip (CPU) produced in the last 20 years – and found in 2017.
- Both flaws allow a malicious program to gain access to data it shouldn't be able to see by two methods:
  - **Speculative Execution**
    - Modern processors would compute both the "if" and "else" result of a statement before knowing the outcome/path.
    - May speculatively "choose" path based on previous executions.
    - Reverts the unchosen part later.
  - **Caching**
    - To speed up access data to in memory, we use a cache.
    - 'Protected memory' accessed may be put in the cache before the privilege check is done i.e. the "if".
    - Get protected data from cache before it is reverted.

Meltdown

- Meltdown is the variant of this concept that:
  - "melts" the normal security boundaries *between* programs running.
  - Allows access to other programs memory and special data.
  - Only works on specific Intel Chips.
  - Example:
    - An attacker runs JavaScript in your Chrome browser that reads data from you logging into Spotify.

Spectre

- Spectre is more universal:
  - Allows reading privileged data in the same program.
  - Works on nearly all chips available at the time of discovery.
  - Called Spectre for:
    - “speculative execution”
    - “it will haunt us for some time”
      - It is not a simple fix.


:(
Your PC ran into a problem that it couldn't handle, and now it needs to restart.
You can search for the error online: HAL_INITIALIZATION_FAILED

Meltdown & Spectre Patch = BSOD

- Microsoft released a patch for Meltdown and Spectre…
  - Saw many systems blue screening.
  - Realised many AV vendors were using syscalls that were not officially supported.
    - This caused the system to crash.
  - Customers would not receive the MS patch UNTIL the **anti-virus company** supported it.
  - An example of the complex ecosystem. Hundreds of AV, customers not knowing how to manual 'set the registry key' to get the update.
  - Public Google Doc collating which AV supported it:

				Last update: 5th January 2018 @09:54am GMT
Vendor	**Product**	**Sets registry key**	**Supported**	**Comment**
AVAST		Y	Y	Fixed.
Avira		Y	Y	Fixed.
BitDefender		**N**	N	Fix this evening or tomorrow
Carbon Black		**N**	N	Assessing impact
Cisco	AMP	**N**	N	In testing
CrowdStrike	Falcon	**N**	Y	Registry key change scheduled for Monday
Cylance	PROTECT	**N**	Y	Manual registry key setting
Cyren	F-PROT	**N**	N	Working on a fix, cannot set registry key thru usual update

Threat

- Term has **different meanings, depending on context**
  - Historical reasons – you need to know both
- **Threat Modelling**
  - Important activity to understand & defend against weaknesses
  - Applied at the design and development stages
  - Could reasonably be called "**attack modelling**" because it focuses on what attackers could do, and how you would defend against that

- **Further common meaning**: the sum of more high-level environmental factors to which a system is exposed
  - E.g. banking systems are exposed to both organised crime as well as state-level attacks in case of war
  - Thinking about threats in this way is useful to estimate likelihood of certain attacks, and attacker's capabilities

Attack

- Activity that **intends to cause harm** to the system
  - *Example*: installing a “key logger” on a machine that records everything the user types
  - *Example*: sending many requests at once to keep a system so busy that it can't offer service to real users (Denial-of-Service)

- An attack can occur even if there isn't a vulnerability
  - *Example*: attempt to login by sending a guess as the password.

Controls

- A measure to defend the system by avoiding risk or attack, detecting it, or mitigating it
  - Preventive – e.g. authentication of users
  - Detective – e.g. Intrusion Detection Systems (IDS)
  - Corrective – e.g. Denial-of-Service protections
- Can be technical, but do not have to be:
  - Physical security
  - Operational security
  - Regulatory/governance

Trust/trusted/trustworthy

- Trust in a system is the expectation that the system will **operate as intended.**

- The terms **trusted** and **trustworthy** have a particular meaning in the context of computer security.
  - The definitions come from the point of view of system analysis

- A system component is called **trusted** if its failure would compromise the system's security
  - I.e. as in "we put out trust in it not failing – and if we are not sure, then it's our job to make it trusted"
- A **trustworthy** component if we can indeed safely assume that it is not going to fail

Check: trusted or trustworthy

- Taking the previous two definitions and applying them to warnings to users - should we say:

"Do not click on links to untrusted websites!"

or

"Do not click on links to untrustworthy websites!"

Check: trusted or trustworthy

– *What is a double-agent of the NSA?*

Security goals

- A **security policy** describes the **security goals** that a system is meant to achieve. Typical goals are:
  - Confidentiality (but see next slide)
  - Integrity
  - Authorization
  - Authenticity
  - Non-repudiation
  - Accountability
  - Auditability
  - Anonymity/Pseudonymity
  - Availability
- Typically, a system aims at meeting a **well-chosen** subset of these goals.

Secrecy/confidentiality/privacy

- These terms relate to information that should not be accessible except by those who are supposed to know it. We use them in the following way:

- Secrecy is a **technical term –** the effect of mechanisms to limit accessibility of information to the intended group

- **Confidentiality** involves the **obligation** to protect secrecy **–** it is a possible **security goal for a system**

- Privacy is the capability to **protect personal information** and prevent invasion of **personal space**
  - **System may achieve privacy** by meeting certain security goals

Integrity

- Security goal: must be able to verify that information has not been altered (or altered in a non-permissible way)

- Examples:
  - Protect integrity of database entries
  - Protect integrity of information during online banking – customer and bank must be able to detect malicious attempts to change information

- Note careful wording: key is detecting illegitimate alterations
  - An attacker in the network can always alter the data - in global networks such as the Internet, this is hard to prevent
  - The important thing is that sender and/or receiver can reliably detect it and react to it

Authenticity, Authorisation, Accountability

- Authentication and Authentication
  - Authenticity means that the origin of a message can be determined and verified.
  - Authentication means that the parties in a communication can be determined to be who they claim to be

- Authorisation:
  - An entity is assigned a privilege (to carry out an action, access data, etc.), and this authorization can be verified
  - Access Control carries out the authorization check
  - Question: Does Access Control need Authentication?

Non-repudiation and Accountability

- Users often desire a way to make sure that they can convince others about what happened.
  - Can be very hard to achieve in practice, especially over computer networks
  - An attack could involve doing something and denying it was done, leading to confusion about the correct state.
  - *Example*: you send a message to transfer money to someone, then say you didn't and demand the money be repaid to you by the bank

- Accountability
  - Means that it is possible to map the outcome of an action or state change to the entity that caused it
  - E.g. change in database can be correctly traced to the user who caused it

Auditability

- Organisations often need to find out what happened.
  - Who sent which messages
  - Who accessed what data
  - How data got into the current state – data provenance

- Legal requirements (forensics) may exist
  - *Example*: Enron, tax affairs, identifying the criminal

- Audit information is also very useful after a security failure has occurred.
  - Learn how it happened
  - So controls can be introduced in future

War Story – CBA Fraud

In 2017, the Commonwealth Bank became embroiled in a fraud scandal – primarily facilitating fraud.

- **May 21 2015:** two were raided and arrested and $3 million in banking receipts were found, many for CBA accounts.
- Managed to launder over 1.7 million over 7.5 months.
- How did they evade detection?
  - Used multiple branches depositing less than the $10,000 threshold each time i.e. 9900.
  - This is a legal threshold: deposits over 10k are logged and reported to the government.
  - Meant CBA were in hot water as they weren't meeting regulatory requirements due to a system technicality.

Availability

- The system should provide service for the intended users.
  - Requests are a) processed and b) this happens within an acceptable time

- If the system isn't available, damage to the organisation comes from lack of normal functioning.

- Opponents may attack availability for many reasons.
  - Malice – e.g. attack against competition
  - Blackmail – e.g. extortion of money
  - Warfare – take out the opponent's infrastructure
  - Accidental side-effect – e.g. application is sensitive to malformed input and gets hit by an (otherwise harmless) Internet scan

War Story – Dyn & Mirai Botnet

Dyn, a major DNS (Domain Name System) infrastructure provider was offline for most of October 21.

- **DNS:** Translates your human readable URL (www.google.com) to an IP address – so you can talk to the web server.
- Brought down access to sites like Twitter, Reddit, Netflix etc.
- Attacked by the Mirai Botnet
  - Made up of IoT devices i.e. DVR players, cameras.
  - Traffic flow of 1.2 Tbps
  - 100,000+ malicious endpoints/bots.
  - Most powerful at the time

Anonymity

- Human desire to be able to do things without being identified.
  - E.g. accessing websites with sensitive topics – like to stay unknown
  - Keep knowledge from other groups – business, family, government...
  - The user's goal for anonymity may conflict with other goals, such as auditability.

- Different forms of anonymity:
  - Network anonymity: no-one can identify the user in the network
  - Data anonymity: no-one can re-identify a person in an "anonymized" data set

- Perfect anonymity is impossible to achieve in practice
  - Near-global observation of network traffic breaks anonymity
  - No such thing as "anonymized data set" that is simultaneously useful!

Achieving the goals

- To achieve security goals in spite of opponents, systems have controls.

- Perfect security does not exist in practice
  - Every system still has vulnerabilities
  - Learn from security in banks, airlines, etc.
  - Get the design stage (as) right (as you possibly can) - it's a good investment
- Every control has drawbacks: cost, inconvenience, usability!
  - A decision is needed about which control to follow.
  - Someone must compare the drawbacks and the benefits

Trade-offs

- Every control has costs as well as benefits.

- There are financial costs
  - pay for security products, pay for security advice, pay operators for time doing security rather than other services…

- Security controls make it harder for attackers.
  - Some things can't be done, or can't be done easily
  - So they also make it harder for normal users to do their normal work.
  - This reduces the value of the IT systems for their owners.

- Also costs against ease of use in many cases.

Threat Modelling

Context matters

- Security is ultimately contextual.
  - What are you trying to protect?
  - What are you trying to protect from?
  - For how long?
  - Example: how would you store secrets **long-term** – 100s of years?

- **Threat modelling** is a structured way to think about and communicate the actual threats, concrete attacks, kind of attackers and incentives they have.

- Never exhaustive, the application provides an understanding of the context.

Context matters

- *Example:* The lock on your house's door (a control) is not to prevent against a robber, it is to prevent a normal person just walking in.
  - *If you have an actual threat of a general robber – you install window bars as well.*
  - *Still doesn't help if the attacker has a bulldozer – but those guys are rares*
- *Example:* Face ID is a control to prevent your friends and family from accessing your phone.
  - If you are held at a border, you are still going to unlock your phone

- It all depends on what you need to protect, how valuable it is to you and to others, what capabilities the attacker may reasonably have

Types of attackers

- Wide range of motivations:
  - Amateur enthusiasts demonstrating their skills - script kiddies were famous in the early 2000s
  - Blackhats: attackers with malicious intent
  - Whitehats: attackers who are paid to test a system for vulnerabilities
  - Sometimes money is a motivation, sometimes a political statement
  - Governments engage in certain activities for espionage or warfare.

- Not attackers:
  - People making mistakes – lack of intention! May be mistaken for attackers, of course.
  - Hackers per se – "to hack" originally meant to get a system to do things it was not designed for
  - Read up on the word – the negative connotation today is a recent thing!

Levels of organization

- Some attackers are computing experts
  - Often considerable knowledge about hardware, software, networks
  - Much can be legitimately learned:
    - Ethical Hacking
    - Capture The Flag competitions
- Organized crime is a reality:
  - Tool sets written for attacking systems
  - Experts advertising their services in underground forums
  - Sale of illegally obtained data etc.
- Dual–use tool sets: what is a tool to repair, debug & test systems in one set of hands is a tool set in malicious hands
  - Law in most countries recognizes that and requires malicious intent, not just ownership and use of such tools

Types of attacks

These violate a key security goal. Can you tell which one?

- Unauthorised intrusion into a system
- Intercepting messages on the wire
- Flooding a server with requests
- Unauthorised modification of data
- Falsification

Types of attacks

These violate a key security goal. Can you tell which one?

- Unauthorised intrusion into a system (Authorization, possibly Confidentiality, Integrity, and more)
- Intercepting messages on the wire (Confidentiality)
- Flooding a server with requests (Availability)
- Unauthorised modification of data (Integrity)
- Falsification (Integrity, Authenticity)

Incentives For Attacks

What are we actually trying to protect?

- Monetary values
- Reputation
- Machines – botnet usage, framing users etc.
- Information
  - Business secrets
  - Personally identifiable information (PII)
- Access and authority

Social Engineering

One typical method of attacking a system is performing social engineering to gain knowledge or access.

- Exploits the human as opposed to the machine
  - Psychology – our willingness to help
  - Relationships
  - Common beliefs, expectations or social norms
- Example:
  - Walking up to a card access door once someone opens it, hoping they hold the door for you expecting you also have access.
  - *Bonus*: Pretending to be on the phone so they can't ask.
  - *Bonus*: Pulling out your wallet/a card to make it seem like you have a valid card.
  - *Bonus:* Be carrying things that would make it hard for you – people want to help.

More in week 9 ☺

War Story – Frank Abagnale

One of the most well known impostors for his actions between the age of 15 and 21. Assumed no fewer than eight identities:

- **Airline Pilot:** acquired a uniform by calling and claiming he was a pilot who lost it. Estimated to have flown for free as a passenger on more then 250 flights. Controlled the plane on some occasions.
- **Physician**: Supervised resident interns at a hospital and did not do any actual doctor's work.
- **Attorney**– Forged a law transcript, passed the bar exam (a US exam required to practice law).
- Later, Frank worked for the security industry and the FBI
- Film: Catch me if you can (sorry, not during lecture hours)

Summary

- Systems involve hardware and software components, and also people. The latter are as important as the tech.

- Security tries to protect assets.

- There are many different security goals for different stakeholders.

- Security involves tradeoffs, and decisions must be made.
  - These should be informed decisions and to prevent against your threat model.

- Know the terminology we will use in this unit!