COMP6441 Security Engineering and Cyber Security

Student Course Textbook 19T2

Week 1

1. Security Engineering

What makes a good security engineer?

- Good at **Analysis**
    - Don't just say the first thing that comes to you or your first reaction
    - Think from an attacker's perspective, not just defender's
    - Strengths, disadvantages, recommendations
- Need an awareness and be able to form other disciplines.
    - Learn from history
    - Be able to identify features that look familiar and build on previous solutions
    - Learn from mistakes
- Case studies
    - When things go wrong, we need to look at it and learn from it and how to prevent it in the future.
    - Don't just allocate blame
    - Look into what happened, the outcomes, reasons why things occurred the way they did, recommendations
- Decisions
    - Need to be able to weigh up the points for and against for all options, and make a good decision.
- See what's important
    - Identify important details in a mess of complexity
    - When a situation is complex, it's hard to know where to begin.
    - Push the non-important stuff to the side, and focus on what matters
- Certainty
    - Good security engineer is never really confident that they're right.
    - You're never done. Don't just stop when you think you have the best answer
- Humans
    - Need to understand humans
    - Many issues occur in the interface between humans and technology, so if you're just looking at the technological side of things you're not getting the full picture
    - Complexity = Overlook mistakes and vulnerabilities

Vulnerabilities vs Exploits vs Threats

- Vulnerability - weakness in a system
- Exploit - scripts or other methods utilising a vulnerability to compromise a system
- Threat model - what you are up against

2. Security Through Obscurity

Security through Obscurity: Relying on secrecy or complexity for security.This was an issue for a variety of reasons including:

1. The ability to audit and assure of a complex products security - Increasing complexity means that it is hard to analyse whether a system's security is actually improving when changes are made - because it is harder to understand does not necessarily mean it is more secure.
2. Additionally, this added complexity may actually induce more unforeseen vulnerabilities.

Kerckhoff's principles

1. The system must be practically, if not mathematically, indecipherable;
2. **It should not require secrecy, and it should not be a problem if it falls into enemy hands;**
3. It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will;
4. It must be applicable to telegraph communications;
5. It must be portable, and should not require several persons to handle or operate;
6. Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.

We ***should not rely on obscurity*** to make things secure. Good security happens when the enemy knows everything you know but they can’t decode what you can.

3. Cryptography Principles

Confidentiality: Ensuring that only authorised people can access certain information.
Integrity: Maintaining the “trustworthiness” of a message. I.e. ensuring the message is not altered or destroyed by an unauthorised party.
Authentication: an act, process, or method of showing something to be real, true, or genuine. E.g: logging into CSE lab machines with both zID and zPass
// Availability(which Richard not prefer to be the “A” in C.I.A. Principles): The message or information is in an available condition. E.g. a DDOS attack is an example of an attack that compromises availability.
Non-repudiation - Not being able to refute that something has happened - someone cannot deny the validity of something.

Steganography: Hiding the existence of the message

- Issue: as soon as the method/secret is known, the message AND all past messages are useless
- Example: Greeks under siege would tattoo messages on the shaved heads of slaves, the message would be hidden by hair grown after

Codes

- Codes vs Ciphers: Codes generally operate on *semantics*, (frog means danger) meaning, while ciphers operate on *syntax*, symbols. A code is stored as a mapping in a codebook, while ciphers transform individual symbols according to an algorithm.
- Caesar Cipher is using substitution - shift the letters in the alphabet by x amount
- Substitution Cipher: each letter is mapped to another letter.
- Transposition Cipher: keep the letters the same, but change the order
- Jefferson's wheel: set of wheels with the alphabet on them.
  - Order the wheels and line up the plaintext to encode
  - Write down a lined up gibberish line
  - To decrypt, recipient arranges disks in agreed order, rotates to encrypted message and looks for plaintext
- Playfair cipher: encrypt pair of letters instead of single letters
  - "Digraphs" - pairs of letters (which also don't occur at the same frequency)
  - Typically, J is removed because it is low on the English frequency table
  - Repeated letters in the same group are usually separated by an X
- Polybius square: put the letters of the alphabet onto a grid and replace each letter with its coordinates on the grid (row then column). Sometimes two letters are in same grid
- Vigenere cipher: a method of encrypting plaintext using a series of interwoven Caesar ciphers, based on the letters of a keyword

Bits and dits
- Useful when scale of size changes all the time
- Dits = "decimal digits"

4. Example Case Study: The Halifax Explosion (1917)

What happened:
- The incident caused the largest man made explosion prior to the atomic bomb
- S*S Mont-Blanc* was carrying a large shipment of explosives through port Halifax
- A misunderstanding between Norwegian vessel *SS Imo* and French cargo ship S*S Mont-Blanc* caused a collision course
- Sailors abandoned ship prior to collision and escaped the explosion
- Resulting explosion killed 2000 people and injured 9000
- Vince Coleman, a local train dispatcher was notified of the explosion about to occur and stayed at the dispatch station to prevent trains from entering Halifax saving 1000s and allowing stopped trains to be used for the recovery effort

How could this have been avoided?

Human Weakness of the Week: Adjusting our frame of mind.
- Examples include the Halifax incident where bystanders did not evacuate/take any action after seeing sailors running through the town in fear, as well as 9/11 where those in the second building failed to listen to evacuation messages quickly enough.
- This is often because people are caught in their daily routine, unable to quickly shift between the mind frame of work or having a meal at a cafe, to emergency mindset.
- Confirmation bias

Week 2

Security by Design

- 4 primary colours for a security engineer
    - Trust
    - Secrets
    - Humans
    - Engineering - risk, complexity
- Trust
    - Start with a healthy dose of scepticism when you begin
    - Trust no one, and don't build security on someone else's perceptions of trust
    - Want to make sensible choices with the aim of perfection
    - Never believe anyone who says something is perfect
    - Engineering is really about how can we achieve almost perfection with limited resource and time
- Defence in depth
    - System should be able to work if something fails e.g. submarine division into compartments
    - ***Want to avoid a single point of failure –*** once barrier is broken, everything fails
    - Concentric rings of walls for castles - if one wall breaks, limits areas of access while defending from other barriers
    - Segmentation of confidential data
    - Uni doesn't really do this
- Security by design
    - Build security in from the beginning, instead of adding security when the breaches occurs (responsive security)
- Apes vs ants – represent 2 different paths of complexity
    - Apes – complexity between interactions between individuals
        - Have the capability from breaking from role within society and have freedom
        - Brain develops to allow us to work in larger groups and perform more complex tasks
    - Ants – themselves are not complex, dumb
        - Their society + colony is complex
        - No ant by itself is the one making the decisions, but the colony works as a whole
        - Colony is rigid + unwavering following of roles = predictable
        - e.g. human wearing a castle suit
- Bell LaPadula – levels of confidentiality
    - Not allowing people to have the choice to access specific parts of info
    - Relies on complete human obedience
    - Can only increase controls + penalty to deter
    - Could have an idiot on top = single point of failure
- Physical security
    - Everything we deal with is in a virtual world – within it, can't really tell anything

    - If someone can access the hardware they can do anything
    - Everything we run is running on the machine – if someone interferes with the machine = game over
    - Physical security - the thing we trust all the time as programmers
    - Consider hardware security measures before the software security measures
    - Hardware attacks - have become increasingly more common e.g. Huawei
    - Passwords - can retrieve it by setting up a camera around the space they type it, check the keyboard for imprints
    - Side channels - for every cyber action, there is a corresponding physical trace. “Every contact leaves a trace in the physical world”.
    - Side channel attack - based on information gained from the implementation of a system rather than weakness in the implemented algorithm itself

Vigenere (Caesar + Password)

- Take a password and shift the encrypted message by the alphabetical order of the password. I.e if the pwd is as long as the plain text = one time pad.

    e.g. ABBA would give the shift (0,1,1,0) on repeat, so “Hello” would become

    H+0(A) = H
    E+1(B) = F
    L+1(B) = M
    L+0(A) = L
    O+0(A) = O
- The Kasiski test
    - Used to defeat ciphers where the password/offset repeats, such as the Vignere cypher
    - Find the length of the key by looking for repetitions (longer repetitions are better)
    - Line up each section, then solve the lined up letters as single substitution cipher
- The Index of Coincidence
    - The coincidence index is an indicator used in cryptoanalysis which makes it possible to evaluate the global distribution of letters in encrypted message for a given alphabet. **If the index is high,** (similar to plain text) then the message has probably been encrypted using **a transposition cipher** or a monoalphabetic substitution. I**f the index is low** (similar to random text), then the message has probably been encrypted using a **polyalphabetic cipher** (letter can be replaced by multiple other ones)
    - The easiest ways to determine if you have a monoalphabetic substitution cipher. The encoded message should have the same coincidence index as the language it was encoded in. (E.g. English texts typically have a coincidence index of 1.73)
    - Formula

$$\mathbf{IC} = c \times \left(\left(\frac{n_{\mathrm{a}}}{N} \times \frac{n_{\mathrm{a}} - 1}{N - 1} \right) + \left(\frac{n_{\mathrm{b}}}{N} \times \frac{n_{\mathrm{b}} - 1}{N - 1} \right) + \ldots + \left(\frac{n_{\mathrm{z}}}{N} \times \frac{n_{\mathrm{z}} - 1}{N - 1} \right) \right)$$

$$\mathbf{IC} = \frac{\sum_{i=1}^{c} n_i(n_i - 1)}{N(N-1)/c}$$

        - Where c is the normalizing coefficient (26 for English), n_a is the number of times the letter "a" appears in the text, and N is the length of the text.
    - Manual Calculation
        - Line the text up with a copy of itself shifted slightly.
        - Count the number of times the letter in the first copy matches the second copy.
        - Divide this by the number of aligned pairs of letters.
        - Multiply that by the size of the alphabet (26 for english)
        - The result is an estimate of the Coincidence Index

Enigma Machine

- Enigma Machine was an encryption device used by Nazi Germany in WW2 to protect commercial, diplomatic and military communication. Enigma has an electromechanical rotor mechanism that scrambles the 26 letters of the alphabet
- caesar plus almost infinite password (password extender)
- Each key press causes one or more rotors to step by a letter, changing the substitution
- Understand the weakness and how it was used that allowed it to be cracked
    - Repeatedly used stereotypical expressions
    - Repetition of message key
    - Easily guessed keys
    - Re-transmitting a message on different cipher networks (solving a bad cipher of same message -> solved that enigma key)
    - Not allowing repetitions (e.g. a letter cannot map to itself), reduced possibilities

One Time Pad

- In cryptography, the one-time pad (OTP) is an encryption technique that cannot be cracked, but requires the use of a one-time pre-shared key the same size as, or longer than, the message being sent.
- In this technique, a plain text is paired with a random secret key (also referred to as a one-time pad). Then, each bit or character of the plaintext is encrypted by combining it with the corresponding bit or character from the pad using modular addition.
- **Like Vignere** but the password is the length of the message and completely random, once used the cipher should be discarded

Type 1 & Type 2 Errors

- Type 1: False Positive - reject a true null hypothesis
    - E.g. a medical test saying you **do** have a disease when in reality you **do not**

- Type 2: False Negative - fail to reject a false null hypothesis
    - E.g. a medical test saying you **do not** have a disease when in reality you **do**
- There is a tradeoff between the two

Null hypothesis: statement that there is no relationship between two measured phenomena
Richards example: Sydney Airport Passport Gates sometimes letting people through with the wrong passport, and sometimes not letting people through with the correct passport.

Approximate English letter frequency table

Letter	Frequency	Letter	Frequency
E	12.02	M	2.61
T	9.10	F	2.30
A	8.12	Y	2.11
O	7.68	W	2.09
I	7.31	G	2.03
N	6.95	P	1.82
S	6.28	B	1.49
R	6.02	V	1.11
H	5.92	K	0.69
D	4.32	X	0.17
L	3.98	Q	0.11
U	2.88	J	0.10
C	2.71	Z	0.07

Insiders

Often securities can be broken from insiders because of self-interest, corruption, etc. There are many examples in espionage between Americans and Russians.

Other

- **Entropy**: randomness. If there's order in the data, it's hard to get rid of it even if we encrypt it. Better codes hide the data/information better
- **Edit distance**: distance between valid sentences (very large for English)
- **Work ratio:** ratio between the work a legitimate recipient has to do to decrypt the message, and an interceptor trying to decrypt the message. We want the ratio to be as big as possible
- **Repeat/playback attack**: exploit system by repeatedly transmitting an authenticated message
- **Tempest attack**: Analysing electromagnetic radiation from a distance. Dutch elections
- **Number stations**: radio shortwave station that just repeats numbers. Countries sent messages to their spies which they decoded with one-time-pads. Side channel was the amount of activity on the number stations

How to differentiate between ciphers

1. **Check letter frequencies** - this will tell us whether the cipher is a transposition cipher or not. If the letter frequencies obey english standards then likely the cipher is transposition.

2. **Check coincidence index -** if the coincidence index is around 1.73, then we believe the cipher is likely mono-alphabetic substitution.

3. **Check the periodic CI -** if periodically the CI spikes, the cipher is likely poly-alphabetic substitution.

Week 3

Risk

Is/ought = fact / should be that way. Ought is often a moral/ethical question

Risk is invisible. You don't know you took a risk unless it goes wrong. *Risk* is always *bad regardless* of whether it happens or not. you can't see the intelligence behind it but you see the outcomes.

Humans are bad at assessing risk due to the *low probability* of it happening. When estimating risks we tend to evaluate the risks that we are secure against, it is hard to get an accurate picture of the actual probability of it happening. For example, if you were to get your laptop fixed at a shop, there may be a 1/100 chance of the repair destroying your laptop. To the unlucky 1 person, that chance seems to be 100%. To the 99 out of 100 people who didn't have their laptop destroyed, the chance of repair destroying your laptop seems to be 0% because it didn't and more experiences can we get a more accurate representation of risk. Allocate your resources appropriately across risks.

Humans assess risk based on intuition and past experiences. From Skinners' Pigeon Experiment, observers of the pigeons found that pigeons develop superstitious behaviour, believing in acting in a particular way or committing a certain action, food would arrive. Humans are bad at making decisions about risk based on so called "past life experiences" - it's difficult to evaluate if an event is low probability, because most likely you won't have observed it. Statistically, people underestimate low probability events, because it's unlikely past experience accurately estimates the actual probability.

Low-probability high-impact event e.g. asteroids. We tend to obsess about such events or ignore it.

Past experience usually underestimates low probability events
- But we don't care about low probability events unless they are high impact

Risk matrix – 2D graph. Impact vs likelihood. Helpful for allocation of resources. High impact and high likelihood (focus). Low impact and low likelihood (ignore). The other ones are trickier.
Risk registers – tool for documenting risks and actions to manage each risk

Compliance – involves ability to demonstrate something
- Prevents forgetting things that need to be done
- Sets a minimum standard
- Unlikely policies to comply with are good enough to deal with present risks
- *Compliance Culture*

Measure info in bits:
1 bit of info distinguishes between two cases
2 bits -> 4 cases

- Bits of security - a cipher takes n operations to solve. This can be written in terms of 2^x where x is the number of bits of security required to solve a problem

Centralisation – Systems with a single point of failure often have very high-level security, but if it goes down it has a huge impact.

Space-time tradeoff – is the case when an algorithm/program can increase space usage to decrease time spent or vice versa
- Exploiting space-time tradeoff, an attacker can roughly halve the effective number of bits.

Work factor – work factor is defined as the amount of effort required to break down a cryptosystem
- Increase the amount of work for someone to do a bad thing so that it's not profitable

Dealing with Risk

1. Prevention
Try to prevent an event from happening by removing the vulnerabilities which allow the event to occur. I.e. if a fire is likely to occur in a building because people smoke in the building, then implement rules to prevent people from smoking inside the building.

2. Limitation
If the event cannot be prevented, then try and minimise it by limiting how bad the situation can get. I.e. if a fire does happen to occur inside a building, a method of limitation would be the installation of fire separation walls beforehand, so that the effect of the fire is limited to certain areas. Another method of limitation is by lowering the probability of the event occurring if the event cannot be fully prevented.

3.Passing the risk to a 3rd Party
Similar to Limitation, we try and limit the effects of the risk by shifting the responsibility off to another party. I.e. we can limit the losses occurred from the fire by having insurance, which can help with the recovery from the fire, through financial aid.

4.Wearing the risk
If a risk cannot be fully prevented, it is necessary to just wear the risk and in the case that the bad event does happen, the methods used to minimise the impact will help reduce the severity of the risk so that the worst case scenario does not occur.

Public Key Cryptography

The **problem** with private key cryptography is that if a person wanted to communicate with n people, they would need a key for each pair of people (10C2 people, so in the order of n^2)

How would these keys be established in the first place, *infeasible to pre-exchange keys*

Ralph **Merkle** proposed Secure Communications over Insecure Channels

Merkle Puzzles

Alice gives Bob 1000 sealed envelopes, with each containing a key and an identifier. Bob opens a random envelope and tells Alice the identifier for this envelope. Alice can then look this up in her list and now they can send messages using the key.

For an attacker, then would have to brute force all 1000 (on average 500) envelopes to find the one which contains the identifier they are using.

RSA

RSA was a revolutionary system as it is the first instance of public key cryptography

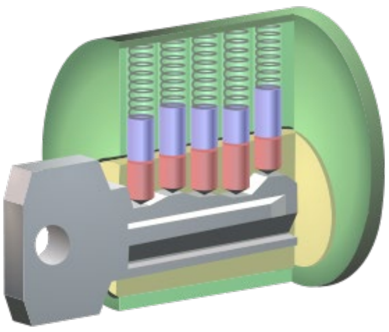
- 2 keys, one public and one private
- *Public* one can be *shared* to everyone that wants to send a message
- *Private* is *retained* by the person receiving the messages.
- The *two keys* are *mathematically related*
- Only the person with the *private* key can *decrypt* the messages
- RSA is based on modular arithmetic (ie A^B mod C = ((A mod C)^B) mod C)
- Decode key is kept secret, therefore we only need to keep 1 secret instead of say n2 keys

Calculating RSA

1. Message M has numerical equivalent ‘m’
2. Choose p,q to be different primes (LARGE)
3. Calculate $n = pq$
4. Calculate $\Phi(n) = (p-1)(q-1)$
5. Choose an integer ‘e‘ such that $0 << e << p, q$ and $gcd(e, \Phi(n)) = 1$(relatively prime / coprime)
6. Choose an integer ‘d’ such that $e \cdot d \equiv 1 \, mod \, \Phi(n)$
    a. Can be done by brute force
7. Encoding m:
    c → m^e mod (n)
8. Decoding c:
    m → c^d mod (n)
9. Receiver makes n and e public (and keeps p, q, d private)

A link to dropbox file on Public Key Cryptosystems and RSA:
https://www.dropbox.com/s/ykk1c6n1u6w6086/RSAandPublicKeyCryptography.pdf?dl=0

Lock Picking



When the correct key is inserted, the gaps between the key pins and driver pins align with the edge of the plug, called the shear line.

Methods:

<u>Lock bumping</u>

-
- Bump key inserted one notch short of full insertion. The key is ‘bumped’ inwards to force it deeper into the key way. A small force is transmitted to all the key pins. If a light rotational force is applied during the slight impact, the cylinder will turn.
- Tension wrench (pin tumbler lock picking)
    - Applies torque to the plug of a lock to hold picked pins in place. Once all pins are picked, the tension wrench is used to turn the plug and open the lock

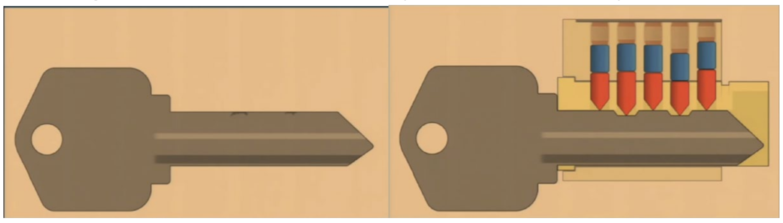
<u>Rake picks</u>

- Rapidly slide the pick past all the pins to bounce the pins until they reach the shear line
-

Shim

- Thin piece of metal, slide it into the shackle housing and pull it from side to side (inserted into the top of the lock, not through the keyhole)
-

Key impressioning

- Starting with a flat or 'blank' key, fashion the key into the shape of the lock

- Don't need anything but hands and the lock, but filing the key can help

Other

- Inserting a card in gap of a door.

Week 4

Protocols - Integrity

- *Umbrella anecdote* - umbrellas make a good defence against cameras
- *Substitution cipher example on English alphabet*
    - fac(26) combinations
    - 26! is very big (~ 4 x 10^{26})
    - 100 x $(10^3)^8$, 100~7bits, 10^3~10 bits, total **~90 bits of work**
    - 16GHz, 16 x 10^9, 2^34(operations per second) * 2^12(seconds in hour) * 2^5(hours in day) * 2^9(days in year) = 2^60 operations (60 bits) for a year of running
    - Note:
        - 1000 ≈ 1024 = 2^10
        - 1 million ≈ 2^20
        - 1 billion ≈ 2^30
- It seems like it would take *too long* for us to *brute force* a substitution cipher, but we haven't taken into account letter frequency.
    - Also, brute force is not the only way to break the cipher.
- English language has *patterns* so brute force time isn't necessarily the time it takes to crack a cipher, redundancy in English, tenses in a sentence match up
- **Entropy**: degree of *randomness*.
    - Complements patterns or order
    - Less patterns = higher entropy
    - English has a lack of entropy
- In English, there are 2^25 possible combinations for a 5 letter 'word', but only 2^13 valid words.
    - Each letter adds 2.5 bits on average.
- Passwords are more likely to use valid English words or phrases than a random string of characters
- *Brute force*: easiest way to calculate and need to find domain of work.
    - **Guided Bruteforce** = bruteforce with heuristics
- Knowledge of the language also factors into time to crack cipher
- *Telegraph example*
- Possible attacks
    - **Replay attack** - an attack on a security protocol using replay of messages from a different context into the intended (or original and expected) context, thereby fooling the honest participants into thinking they have successfully completed the protocol run
    - **Man-in-the-middle** attack - a man-in-the-middle attack (MITM) is an attack where the attacker secretly relays and possibly alters the communications between two parties who believe they are directly communicating with each other. One example of a MITM attack is active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker

- *Challenge response* - a family of protocols in which one party presents a question and another must provide a valid answer to be authenticated; we can't replay a message from one bank to another since the challenge may be different each time

Nonce - a random number that is used once, used to ensure that old communications cannot be reused in replay attacks.

Non-repudiation - unable to deny it was you who sent the message (it had to have been you who sent it)

Social Engineering

Intro

- Art of learning and lying
- Relies on human interactions
- Most common examples include phishing, ransomware, USB baiting etc.
- 95% of all attacks involve social engineering
- It takes about 146 days to detect a breach caused by social engineering

Life Cycle

1. Investigation - learn as much as possible about the victim
2. Hook - initiate the conversation with the target and build rapport
3. Play - obtain the information
4. Exit - leave the conversation without seeming suspicious

Social Engineering Vectors

- Pretexting - invented scenario to get information out from someone
- Baiting - taking advantage of curiosity or greed
- Quid Pro Quo - getting something for something
- Tailgating e.g. heavy box technique (stand in front of sliding doors with heavy boxes so people are inclined to help you and let you in)
- Phishing - digital invented scenario e.g. email to people to get their credit card details. Phone phishing is also easier now because we can automate voices

Security Questions

- Humans often struggle to produce secure and unique passwords
- Social media makes it easier to answer security questions
- Pseudo-private information - information is less private for friends and family
- Prevention and strategy
    - Always lie - consider a security question as another password
    - Use password managers and scrub your social media
    - Don't reuse security questions and answers

Principles of Persuasion

- People can be easily exploited using persuasion
- Reciprocity - when you do someone a favour, people are often obligated to return the favour
    - Can't make it look like a bribe
    - Increase the time delay between the initial gift and the later request
- Liking
    - If you can get someone to like you, it's much easier to influence them
    - Presentation, body language, establish rapport etc.
    - E.g. brands on Twitter are relying on humour to drive sales
- Social Cues
    - People believe in the social cues around them
- Authority
    - People often blindly follow authority figures
    - Can influence other people based on what you wear, what you say, how you act etc.
    - E.g. APEC 2007 Chaser's War - bought expensive cars and were able to get quite far into a secure area, even though they had fake ids

Other

- Dumpster Diving - can go through rubbish with gloves on and find lots of information
- Even after you burn confidential documents, you can still obtain some information from the ashes

Guest: Matt O'Sullivan (Sydney Morning Herald) on GIPA and FOI requests

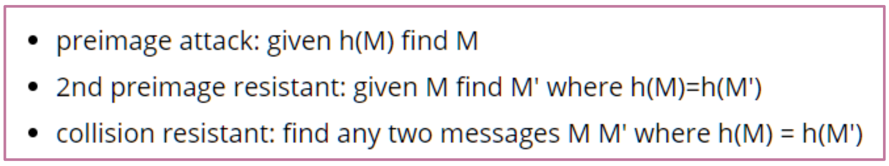
- GIPA [Government Information (Public Access)] allows members of the public access to government documents (subject to certain restrictions eg: commercial in confidence, National security etc.)
  https://www.ipc.nsw.gov.au/information-access/information-access-resources-citizens/how-do-i-access-nsw-government-information
  https://www.oaic.gov.au/freedom-of-information/foi-resources/foi-fact-sheets/foi-fact-sheet-6-how-to-apply
- Seeking a schedule of documents can be really useful, so you know what exists, and what to go for. Ministerial briefing notes are excellent summaries.
- Wording of requests is critical. Can be limited by time, or by subject.
- If your request is denied, you can appeal to the privacy commissioner.

UNSW DVCA
Professor Merlin Crossley, molecular biologist
Universities: UMelb, UNSW, Oxford, Harvard
Runs Crossley Lab - Student research about gene regulation
Married to Margot Kearns

Hashing

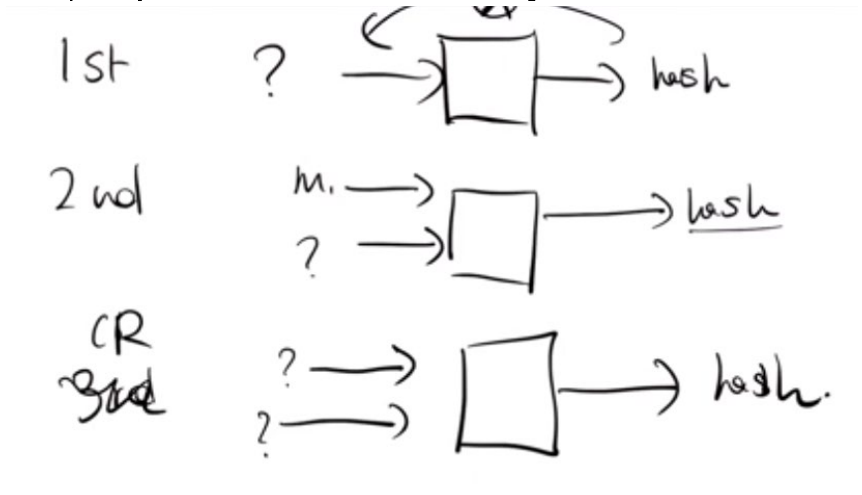
- Hashing is a process that reduces any amount of text or data into a fixed size output
- If you can find collisions you can possibly defeat the hash
- Want to set the hash output to be the biggest it can possibly be i.e. even the square root (half the number of bits) is still too much work for an attacker to do

Desired resistance against the following attacks:



- preimage attack: given h(M) find M
- 2nd preimage resistant: given M find M' where h(M)=h(M')
- collision resistant: find any two messages M M' where h(M) = h(M')

- *Preimage* attack - given the hash, find the message i.e. go **back to original** message from a hash; given *hash(x),* find *x*
  - Want hash to be preimage resistance - **Given** the **hash,** it should be **hard** to **find** the **original** message
  - Complexity on n-bit hash: 2^n. On average: 2^{n-1}.
- *2nd preimage* attack - **given** a **message** M and its hash, **find** another **message** with the **same hash** by going back from the hash; given *x* and *hash(x),* find *y* such that *hash(x) = hash(y)*
  - it should be hard to come up with a second message that has the same hash
  - Complexity on n-bit hash: 2^n. On average: 2^{n-1}.
  - More specialised version of collision attack
- *Collision attack* - **Attacker choose** any **two messages** where the **hash** is the **same;** find *x, y* such that *hash(x) = hash(y)*
  - Difference with 2nd preimage: attacker control 2 messages instead of 1
  - Want hash to be collision resistant - It should be hard to choose any two messages that result in the same hash
  - Complexity on n-bit hash: $2^{n/2}$. On average: $2^{n/2-1}$.



- Uses of hashing
  - Fingerprinting - check that the file matches the hash of the file. Quick summary that we can compare
  - Passwords
  - Proof of Work - need to do a lot of work to achieve something
    - E.g. bitcoin - you can only get a new block when you solve a puzzle, which takes a certain amount of bits of work

        - Hard to produce, easy to verify
        - Hashcash example: 20 first bits of SHA-160 must be zero
    - MAC (Message Authentication Code)
        - Write a message, append a key, hash, send message and hash
        - Receiver gets message, appends pre-shared key, and hashes to compare if message has been altered
        - Provides I and A from CIA (since hashes already provide integrity)
        - h(key | message)
            - Where key|message is key concatenated with the message
            - Vulnerable to length extension attack
        - h(message | key)
            - Vulnerable if collisions occur
        - h(key | message | key)
            - Same vulnerabilities as prev
        - HMAC; h(key | h(key | message))
            - Better than MAC

Cryptographic Hashes

A cryptographic hash has some extra properties -

- Deterministic - The same text will always **result** in the **same hash**
- Its **quick** to **compute**
- It should be **hard** to **reverse**
- Collision resistant - It's **hard** to **find two** different **texts** with the **same hash**
- **Avalanche** property - A **small change** in **input** should **dramatically** change the **output** hash
- Varying input size to fixed size output

Symmetric & Asymmetric Cryptography

Symmetric:

- two people have the same key (shared secret)
- Every unique pair needs a unique key
- If there are 100 agents wanting to speak to each other, we need 99 + 98 + 97… keys (or (n^2 - n)/2 keys, which evaluates to 4950 keys.

Asymmetric:

- Public/private key pair
- Provides authentication, non-repudiation
- Encryption is very slow
- If there are 100 agents that want to speak to each other, we would need 100 private keys (1 private key for each agent)

Week 5 - “Don’t Roll Your Own”

“Don’t roll your own” = never try to make your own cryptographic function. Will always break

Wired Equivalent Privacy (WEP)

- Stream cipher made by a group of people with good intentions, but is very insecure at what it does.
- Wired Equivalent Privacy is a **security protocol** that is designed to provide a wireless local area network (WLAN) with a level of security and privacy comparable to what is usually expected of a wired LAN
- This is secure, the key must **never** be used twice.
- WEP uses a 24 bit initialisation vector (IV) to produce 2^24 keystreams, which means that with enough trace, a key will be guaranteed to be used again. Using XOR, they can figure out the plaintext
- **Process**:
    - A stream of bits (A) is sent from one point to another
    - Another random stream of bits (B) is generated using an algorithm, such as RC4 (very insecure)
    - A (xor) B -> C (Encrypted Stream of bits)
    - However, C (xor) B -> Gives back A again (XOR can be reversed)
        - The XOR function turns A into a stream of bits with the statistical properties of randomness. (Since something random XOR something else will also result in something random)
- Danger when someone transmits the same data under the same key - same data in the same frame
- Fragmentation Attack
    - intercept packet, obtain keystream•construct IP header in 4 byte fragments•concatenate with encrypted payload•IP header points to Internet host controlled by attacker•send packets to AP (Access Point) which forwards along to Internet host
    - The big problem: data & control mixed

Hashes

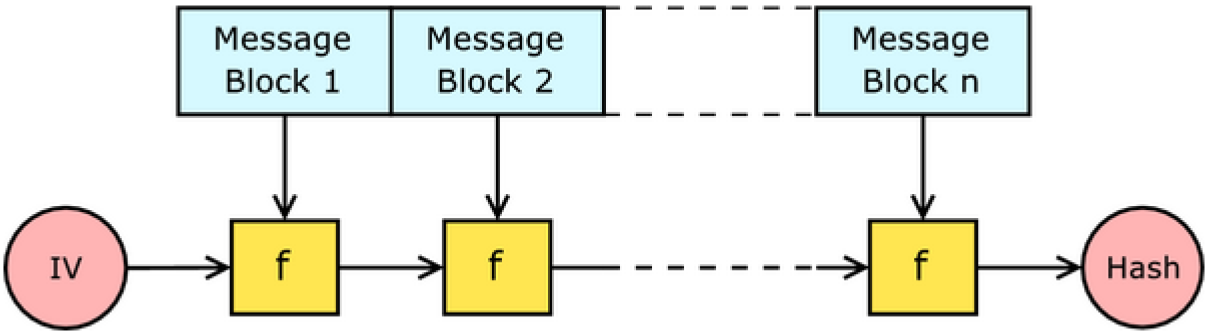
- MD5, SHA0, SHA1 (161 bit digest; high collision) - broken
- SHA2, SHA3 - not broken
    - SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256.
- SHA3 only one **not** vulnerable to length extension attacks (as it does not use Merkle-Damgard Construction) - a type of attack where an attacker can use Hash(message1) and the length of message1 to calculate Hash(message1 ‖ message2) for an attacker-controlled message2, without needing to know the content of message1

- What does it mean to be "broken"? A: Property can be violated faster than brute force (birthday attack, if you can find 2 of the same hash more frequently than root(number of possible hashes)); reveals an underlying structural flaw in the hash

- What is a **birthday attack**?
  - The birthday attack is used as an idea around how good hash functions should be designed by taking the example of birthday paradox. It says that in a room full of people (N), the amount of guesses it takes to find two people with the same birthday on average is sqrt(N). The space here is 365 > N.
  - This is because if you have N people, the probability that two people have the same birthday is 1 - Prob(Everyone has unique birthdays).
    - If you'd want this probability to be more than 0.5 (highly likely) then it turns out you need close to sqrt(n) tries at most (mathematically derived). Lower number of people => lesser guesses needed, but also lesser chances of finding collision.
  - Applying the same analogy on a hash function, it should take sqrt(n) tries to find two inputs in a group of inputs (size n) that hash to the same value. Higher the value of sqrt(n), the stronger the hash function (It takes longer to find hash collision), so that it becomes computationally infeasible to find collisons.
  - A birthday attack won't work if the passwords are salted

Merkle Damgard Construction

Block cipher which builds **collision-resistant cryptographic hash functions** from collision-resistant one-way compression functions (used in the design of MD5, SHA1, SHA2).



1. Break message into a series of blocks (512); may require padding if not divisible by 512
2. Start with IV (initialisation vector)
3. Given hash function f, it takes in IV and block 1, output goes into f again with block 2 etc.
4. After doing this with all blocks, final hash is produced

- **Bank message problem:**
    - Insert shared secret at the front of the hash, as the first block
    - MAC = h(key | data) - his hash function, key|data is the password concatenated with the message

- **Flaw - length extension attack**:
    - If someone has sent a message, and you have the hash and the message length, you can **extend the message** to say whatever you want.
    - You can take the hash you have, pass it into f again with a new message, thereby extending the original message.

Digital Signatures + RSA

- Digital Signatures **RSA** (Ron Rivest, Adi Shamir and Leonard Adleman), DSA(Digital Signature Algorithm)
- If you ever need to sign a large file, **hash the file** then **sign the encrypted hash**.
- Digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. A valid digital signature, where the prerequisites are satisfied, gives a recipient very strong reason to believe that the message was created by a known sender (authentication), and that the message was not altered in transit (integrity)
- RSA relies on people not being able to factorise the large number that is the mod, if someone is able to efficiently find the factors of a number, RSA will break.
- Signatures can be moved from one document to the other if they have the same hash
    - Based on collision attack described below

Hash Collisions

- Summary: **you can make it look like someone signed something they didn't**
- Half the number of bits in the hash size needed to do a collision attack. Why are collisions bad? (ans. above)
- **Collision resistance: It should be hard to find any two things that have the same hash**
- Example:
    - In a PDF that says "I promise I will give you $100", you receive a signature with the original document. In interest of attacker to change the amount to something higher.
    - If they find another pdf with the same hash, the signature can apply to both.
        - **How**? Change 1 bit in each document that doesn't change anything visible in the document, and keep hashing them, until you find 2 identical hashes.
    - After finding the collided hashes, there is a hash for the document that says $100, which now also maps to a document that says you'll give $1,000,000.
    - The attacker can now move the signature to the new document which says you'll give $1 mil

Key Stretching + Salting

- Types of password attacks:
    - Online: you type the password into the browser (manual)
        - can be locked out by site
    - Offline: you steal the file containing the passwords (likely hashed) and attempt to decrypt them locally
- **Salting** a hash is a random string that gets **appended** to the password before it gets hashed. Salt can be a single string, or a unique string for each user. If your salt is unique for each user, **each user's password is almost guaranteed to be unique before hashing**. The goal is to make the amount of work to decode a hash as great as possible, and to prevent attackers from using precomputed rainbow tables. (which are a collection of hashes for common passwords).
    - A **rainbow table** is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Tables are usually used in recovering a password (or credit card numbers, etc.) up to a certain length consisting of a limited set of characters. It is a practical example of a space–time tradeoff, using less computer processing time and more storage than a brute-force attack which calculates a hash on every attempt, but more processing time and less storage than a simple lookup table with one entry per hash. Use of a key derivation function that employs a salt makes this attack infeasible.
- **Key stretching** is designing a hash to be **slow**, such that it adds even more work for a hacker to decrypt a table of passwords. The added time for an individual user would be negligible by comparison. Examples are bcrypt, scrypt.

Data & Control

- Separate data and control - Users have access to data, but they shouldn't have access to control (eg. the captain crunch case)
- Separating data and control completely seems to be 'an impossible problem'.
- Famous example: Captain Crunch whistle. Something about phone lines providing free calls when a 2600hz tone is heard (user-provided audio (data) affecting control). Can be recorded and played back, or reproduced, e.g. with a whistle that was given away with Captain Crunch cereal that happened to be 2600hz.

OPSEC (Operational Security) PLEASE ADD WEEK 5 PEOPLE

- Steps
    1. Identification of Critical Information
    2. Analysis of threats
    3. Analysis of vulnerabilities
    4. Assessment of risk
    5. Application of appropriate OPSEC measures
- First formalised around the time of the Vietnam War

Passwords - PLEASE ADD WEEK 5 PEOPLE

- Salt - add a random string to the end of the password to the end of the password so the hash output will be different
  - Don’t add the same salt at the end of all your passwords
  - Don’t make your salt too short e.g. 3 ascii characters - should be at least 256 bits long
- Use a strong encryption method e.g. sCrypt, bCrypt or Argon2
- Don’t store password hints
- Password generation
  - Correcthorsebatterystaple - take 4 random words and put them together. The length will create enough entropy
  - Passphrases - long with lacky lexicon but good syntax e.g. greencloudssleepfuriously
  - Initialisation of a phrase - maybe take the first letters if you have a max character limit

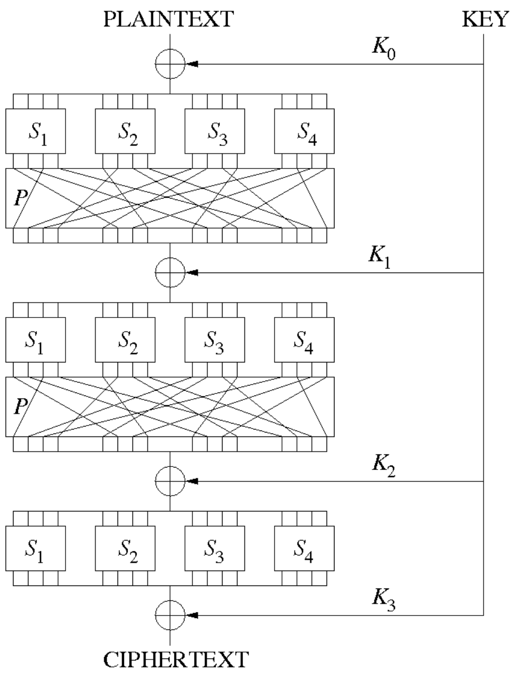
Week 6

AES (Advanced Encryption Standard)

- Uses **substitution-permutation network:**
  1. Each plaintext block is divided into sub-blocks. Round key is derived from the general key, and combined with sub-blocks in each round to form input sub-blocks.
  2. Sub-blocks go through different substitution boxes (S-boxes) or permutation boxes (P-boxes) to produce the cipher text block. S/P boxes are applied in alternating rounds. S-boxes are like substitution cipher, whereas P-boxes distribute any bit to as many S-boxes input as possible.

  * Change in one input bit to S-boxes result in huge change of output bits

  * Satisfy confusion and diffusion principle
- Richard claims that it seems to not have been broken yet in feasible time
- *Note: might need to make this section a bit more comprehensive*



Buffer Overflow

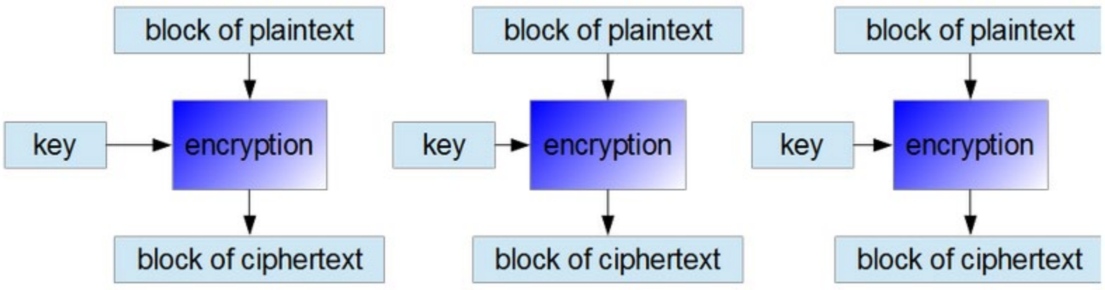
- One CPU does one thing at a time
- Rapid *context switching* occurs between processes
  - Hardware, operating system *interruptions*
- *Stack* keeps track of all processes currently interrupted
  - Lets CPU know what processes to switch back to
- The latest process in the stack is at top
  - When it finishes it is thrown away and stack pointer moves down
- When a process is uninterrupted, its info must be restored
  - Info is stored in disk
- Current running program can also store data in stack
  - Hence, why not store data in stack rather than disk
- Writing to invalid memory/*beyond end of buffer* (eg. Overflowing an array) may cause memory of an interrupted process to be overwritten
  - Simplest example is when user input overflows an array but the program doesn't check
  - The mem overwritten may contain the data for the next instruction that process needs to carry out when uninterrupted
  - To perform attack, write to *return address* of next process to wake up and tell it to go to the start of the array, where your malicious code is written
  - If you don't know where in memory your array is located, you'll need to cause a memory leak to find out

Block Modes http://www.crypto-it.net/eng/theory/modes-of-block-ciphers.html

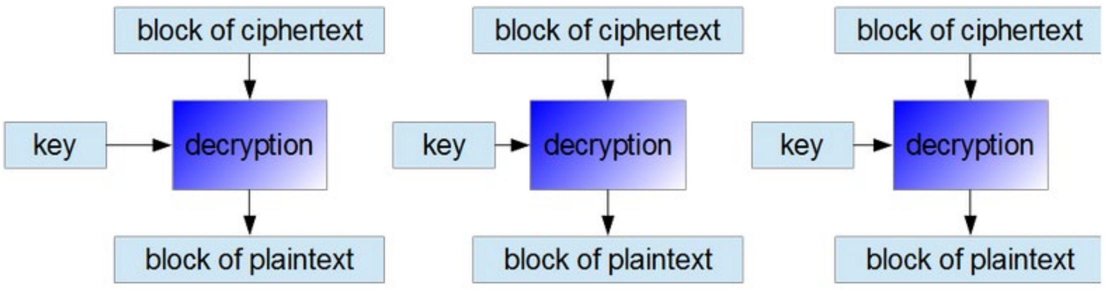
- Allow block ciphers to work with large data streams without the risk of compromising the provided security
- **initialization vector** is randomly generated used to ensure distinct ciphertexts are produced even when the same plaintext is encrypted multiple times independently with the same key

ECB (Electronic Codebook Mode)

- Each plaintext block is encrypted separately
- Each ciphertext block is decrypted separately

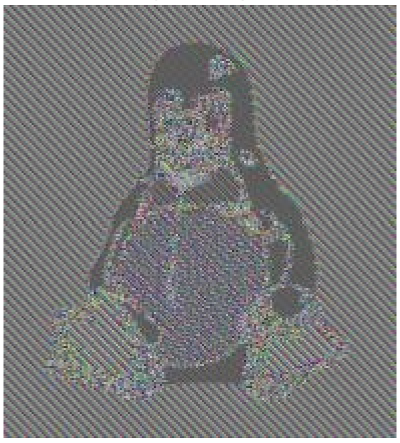


Encryption in the ECB mode



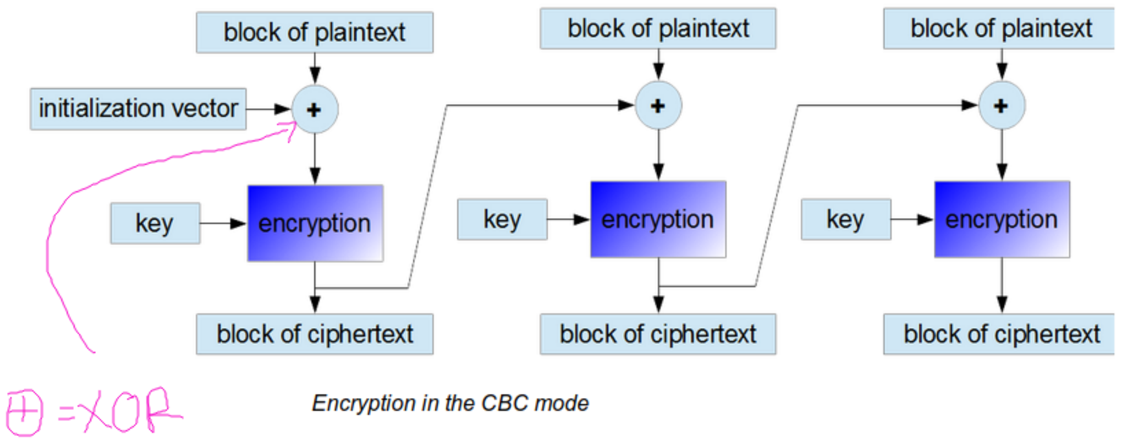
Decryption in the ECB mode

- Never use ECB cuz of the penguin image (on wikipedia) - like encrypts to like. Linux penguin logo encrypted with ECB clearly shows outline of original image

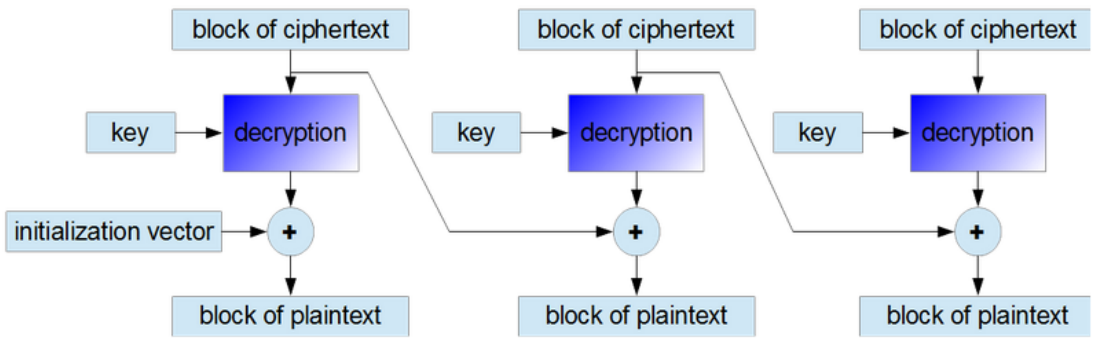


CBC (Cipher-Block Chaining)

- Each plaintext block is XOR'd with the previous ciphertext block that was produced.
- The first plaintext block is XOR'd with a random initialization vector which is the same size as the plaintext block

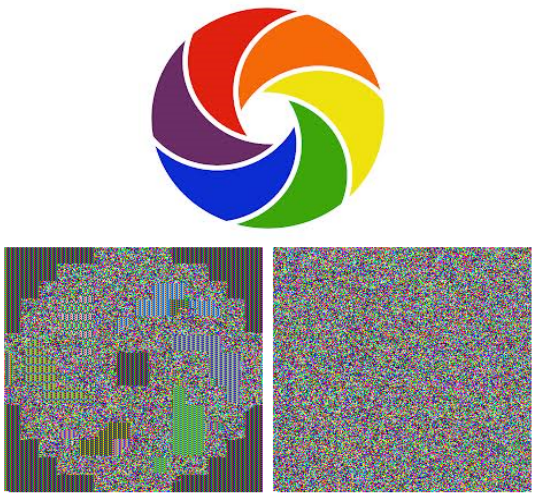


Encryption in the CBC mode



Decryption in the CBC mode

- ECB vs CBC

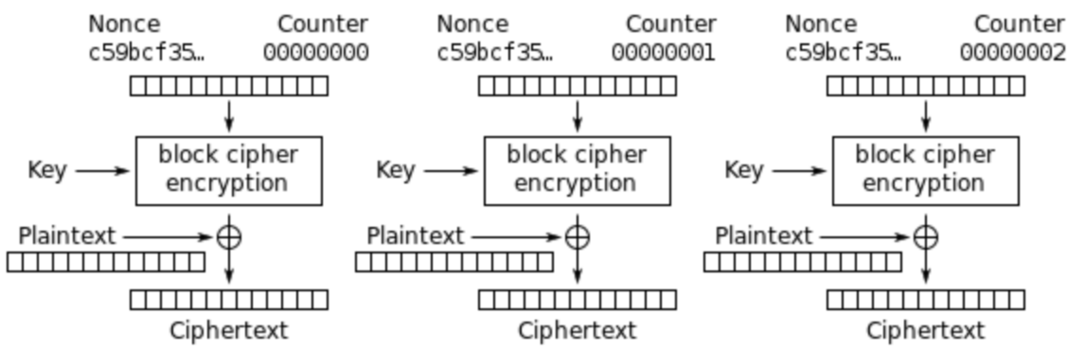


The bitmap image encrypted using DES and the same secret key. The ECB mode was used for the left image and the more complicated CBC mode was used for the right image.

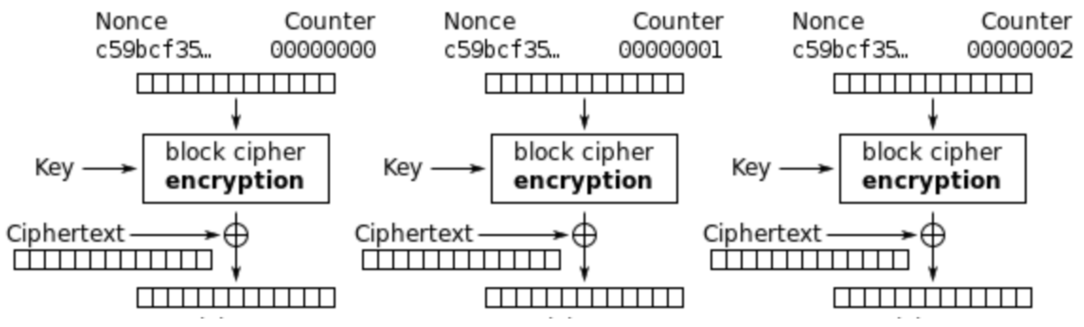
Question about one of the CIA properties

CTR (Counter Mode)

- Makes a block cipher work in a similar way to a stream cipher
- A pseudo-stream cipher
- Difference between a stream cipher and CTR: Instead of XOR with a bit of pseudorandom key stream, each bit of plaintext is combined with a bit of random nonce and counter value encrypted together.
- Since counter is increased for each subsequent block, same encryption of each block results in different bit of output. Doesn’t require previous output, **hence is parallelizable.**



Counter (CTR) mode encryption



Moore's Law 1965

- He noticed the number of transistors in a chip doubled every 1-2 years
    - Later also computing power; roughly doubles every 18 months
    - Hence you lose one bit of security every 18 months

Disk Encryption

- How Richard says encryption keys are stored:
    - Generate a random key to encrypt the disk
    - Encrypt the key (that can be decrypted with a password) and store it somewhere NOT ON THE DISK
- What Windows OS used to do:
    - Ask the disk if it can encrypt itself
    - Let the disk encrypt itself
- Forensics
    - RAM can still store data even after shutting off power (for a short period of time).
    - If RAM is frozen, data leaks even slower, and thus can be retrieved.
    - Police are then able to analyse it for say, the encryption key so that they can get to the hard disk.
    - Some links *(please don't delete me)*:
        - https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1162&context=adf
        - https://en.wikipedia.org/wiki/Cold_boot_attack

Authentication

- What is Authentication? Suppose that we are in an isolated environment, where our only job was to provide “authentication” given data from the outside world, and nothing else. How do we authenticate?
- Factors of Authentication (Multifactor authentication)
    - Defense in depth
    - Something you know
        - Secrets (e.g. password)
        - Flaw: easy to prove that you know a secret, hard to prove that no one else knows the secret
    - Something you have
        - Straight-forward: do you have something that only the correct person should have
        - Commonly implemented as 2-factor authentication (i.e. with mobile phones)
        - Flaw: we know how to intercept messages now (more of an implementation flaw)
    - Something you are
        - Something that defines you
        - Mostly implemented as biometrics
        - Flaw: easy/viable to fake e.g. biometrics (again, more of an implementation flaw)
        - Flaw: Once its leaked/stolen, it cannot be changed
    - According to Richard: These are all just different forms of “Something you know”
- Hardware bypass of authentication (again, implementation flaw). Because authentication is done with computers/electronics; theoretically, you just have to send the correct signals along the correct wire(s) and thus bypass authentication.
- These are all challenges related to authenticating identity. If we’re just authenticating information, the problem becomes much simpler: we simply just combine information with encryption.

Some stuff mentioned (in passing)

- Weaknesses arise from the implementation of the algorithm more than from the algorithm itself
- The express envelopes story - using the post office to do what you want, while it is just carrying out its normal function

Seminar 1 - Web (PLEASE ADD WEEK 6 PEOPLE)

Key concepts from Cryptocurrency Presentation

- Facilitates direct transactions between individual
- Removes the need for third party intermediaries

What is a blockchain?

- A chain of blocks
- Each block contains
  - Data
    - Contains hundreds of transactions
    - In bitcoin, you're looking at about 2000 transactions
  - Hash of the block
  - Hash of the **previous** block
- Hashes make the block **tamper resistant**

How blocks are formed

1. Transactions are grouped into a **transaction pool**
2. Miners gather transaction to form a **candidate block**
3. Candidate block is given metadata known as a **block header**
4. We hash the block with a nonce - a randomly generated number
5. If the hash is lower than a certain target value the block is added to the chain
6. Miners race to find nonces

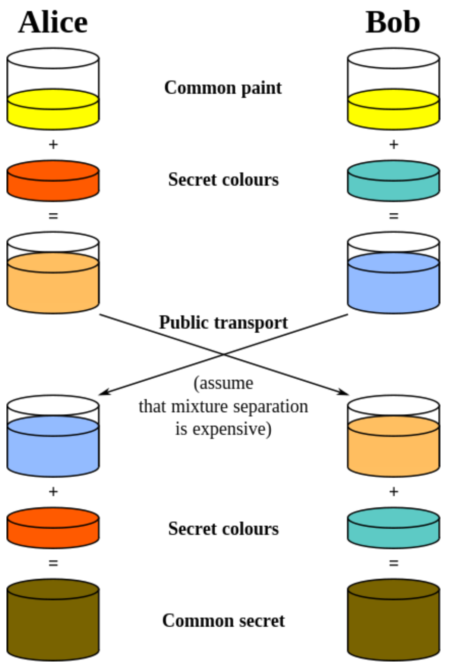
Historical flaws

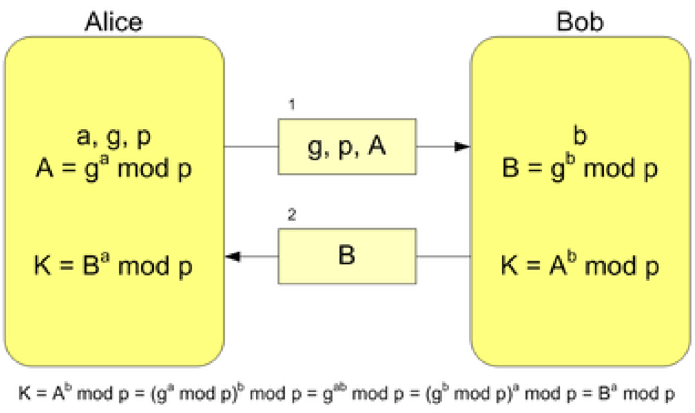
- 51% Attack
- Attack on exchanges
  - Find weakness in exchanges to gain coins
  - MtGox was subject to an SQL injection
  - User databases were attacked
- Bugs in smart contracts
  - Allow developers to program their own smart contracts
  - A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. (Blockgeeks)

Week 7

Diffie Hellman Key Exchange

Method of generating a shared secret between two people.
Here's a nice graphic representation




Alice
Bob
a, g, p
$A = g^a \bmod p$
$K = B^a \bmod p$
1
g, p, A
2
B
b
$B = g^b \bmod p$
$K = A^b \bmod p$
$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

Where:

- a, b : private key
- g : base
- p : huge prime number
- A,B : public key
- K : shared secret

Diffie Hellman does not give us **authentication** it only gives us **confidentiality** (replay attacks cannot be authenticated). With a numeric example of why this works is because (a^b)^c = (a^c)^b=a^bc, so if a is known and you tell someone a^b and they tell you a^c you cannot work out a^bc if you do not know the secret.

Check out the hall of fame for Dean Wunder's explanation on this.

- Solves the problem of sharing a secret between 2 parties at distance
- Someone can listen to everything, and still not understand anything

- Public knowledge:
  - Use 5 as the base
- Private knowledge
  - P1 chooses number 7
  - P2 chooses number 3
- P1 tells P2 5^7 = 78125
  - P2 takes 78125^3
- P2 tells P1 5^3 = 125
  - P1 takes 125^7
- The result will be the same
  - This result is known by P1 and P2 but no one else
  - So it can be used to encrypt messages now

- Reality:
  - pick a big prime number for base
  - big number for the index too
  - discrete log problem: it's hard to go backwards if you do this with big numbers
  - Use a publicly known value used as modulus to make the number smaller
  - Could use this to establish an RSA key or an AES key
- Gives us confidentiality and integrity, not authentication
- Breaking this:
  - man-in-the-middle
- Diffie-Hellman gives **Perfect Forward Secrecy** (PFS), also known as Forward Secrecy, is an encryption style known for producing temporary private key exchanges between clients and servers. For every individual session initiated by a user, a unique session key is generated. If one of these session keys is compromised, data from any other session will not be affected. **Therefore, past sessions and the information within them are protected from any future attacks.**

- Krak Des Chevaliers
  - Crusades era castle, held by the Knights Templar, had an outer ring and inner ring. Eventually fell when the Ottomans or whoever forged a letter from the head of the Templars telling them to surrender, and never fell due to weak defences
  - Also an example of defense in depth
  - Fell to rather than defence flaws

Cyber Literacy - Vulnerability

- Vulnerabilities - a potential weakness in something
- Software Bug - sometimes a vulnerability
- Exploits - attacks a vulnerability to compromise a system
    - Memory Corruption Attacks - the attacker changes what is in memory so that the program behaves differently, the most common type is a buffer overflow:
        - Buffer overflow: Attack on the stack, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.
    - How functions work in C: Functions freeze their information when a new function is called, the instruction pointer saves the location of the parent function (I think?), the stack grows
    - Format String error: submitted data of an input string is evaluated as a command by the application- Was a problem everywhere, until everyone realised and patched against it within a few months, and it wasn't an issue anymore, but is Starting to come back
        - %x attack - arbitrary read of information below our memory address
            - E.g. `printf("%x\n"); // read the next item on the stack`
        - %n attack - arbitrary write of information below our memory address
            - Value written is the number of bytes in the string so far
            - E.g `printf("1234%n\n") // put 4 in the next position on the stack`
            - When overwriting the return address of the function, it is essentially an arbitrary jump
        - To protect against format string attack
            - Don't pass a variable as the only argument into a printf
            - Provide all arguments that correspond to the format string you have written
    - People who don't take into account buffer overflows anymore are like the anti-vaxxers of security engineering
    - Printf - anything that's not properly formatted/specified just get printed literally.
        - printf("%s \n", "hello world"); - this is the "correct way" but no one does that, instead just write- printf("hello world \n");
        - Take an input example - name ← enter name
          printf(name);
        - → Enter name = "Richard%s"
          printf("Richard%s");
          C will look for an argument for the %s character, if there is no argument in the function then it will just print the next string from the stack
          → Enter "%x%x%x%x%x ….%x"
          Will print the next n addresses in hexadecimal - can get it to print stack information, this can then be made to print the canary if we know where it is on the stack. By doing this we can keep the canary there and overflow the buffer till the canary and then give the known value of the canary and continue overflowing the buffer without the canary realising there has been a buffer overflow.

        → “%n” - writes to memory as well!! - hacker used this to get root access. Can overwrite address in the stack
    - Swiss cheese analysis - sometimes the holes line up 😫

\\

- Shell code - piece of code to grant terminal console on remote device (ultimately used to escalate privileges
- NOP Sled - create a long list of No-Ops (0x90) so you can jump to the sled and then slide down to your malicious code i.e. add a bunch of NOPs to your malicious payload which will pad said payload so at any point, when the program jumps to ANY of the memory addresses of your NOPs, it will execute the NOPs and reach your malicious code.
    - There’s more sophisticated ways to do that now that look like the program is doing something when its not. Flipping backwards and forwards, adding and subtracting a constant etc.
- Vulnerabilities stored on regulated database - CVN
- Responsible disclosure
    - Consult the Vendor first
    - Then escalate to CERT (or relevant authority)
    - Still up in the air

- Cold War thinking: It’s more important to win/destroy the Russians than to have a habitable planet 1000 years down the line
    - Ozymandias: Poem excerpt
      “My name is Ozymandias, King of Kings;
      Look on my Works, ye Mighty, and despair!
      Nothing beside remains. Round the decay
      Of that colossal Wreck, boundless and bare
      The lone and level sands stretch far away”
      https://www.poetryfoundation.org/poems/46565/ozymandias

Canary - buffer overflow protection modifies the organization of data in the stackframe of a function call to include a "canary" value that, when overwritten, shows that a buffer preceding it in memory has overflowed into the canary’s address. With the canary it is not going to prevent a malicious buffer overflow. You can find where the canary is in the stack and just fill an array with the canary so that it does not get triggered and alert the computer that a buffer overflow has happened.

NOP sled is a technique used to circumvent stack randomisation in buffer overflow attacks. As stack randomisation and other runtime differences change where the program will jump, the attacker places a NOP sled in a big range of memory. This will slide the code execution down into the payload code, right next to the sled. No-operation is available in most architectures and it does nothing other than occupying memory and runtime.

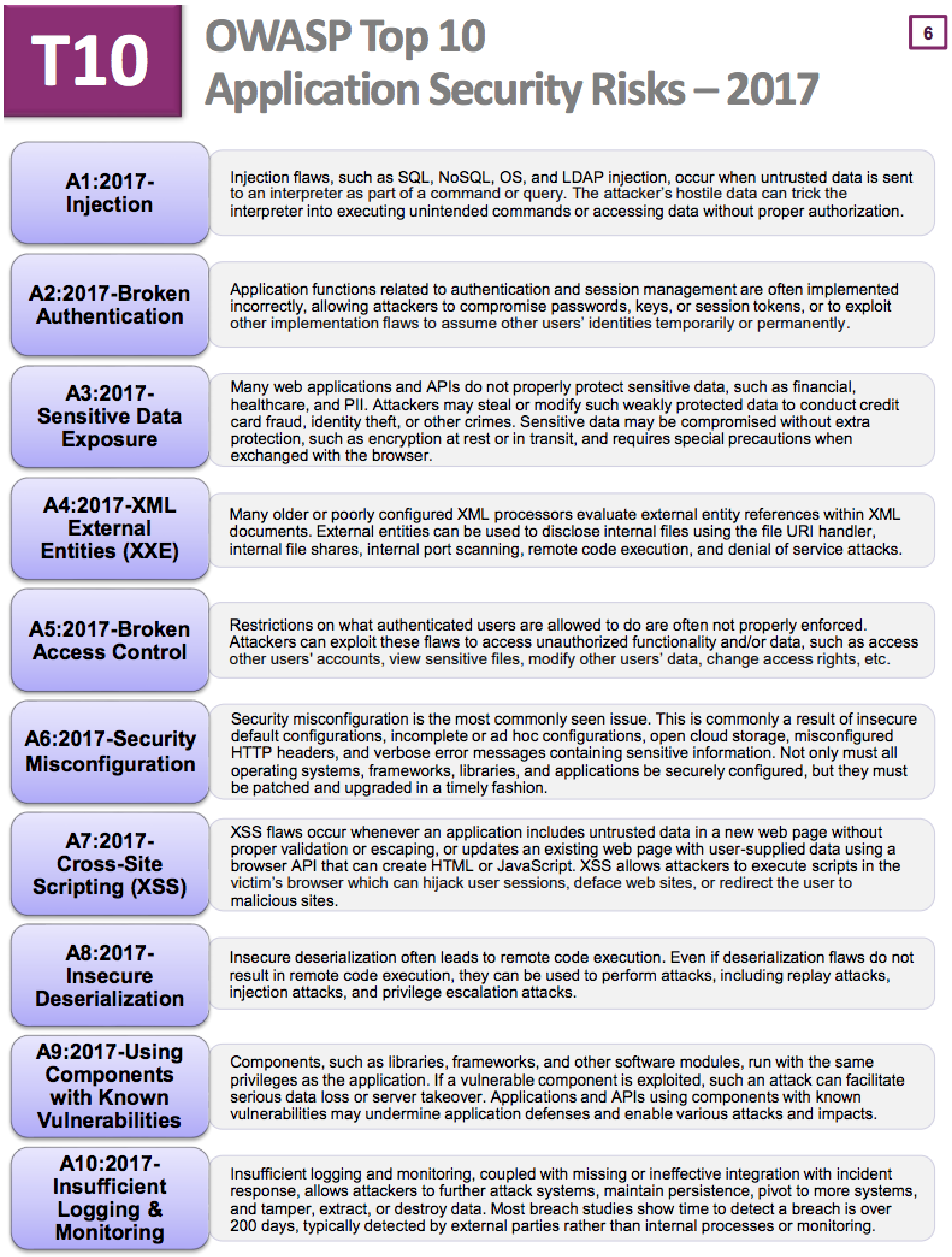
Responsible disclosure -> Vendor -> CERT (e.g. Cert Australia)

Swiss Cheese Safety

- If you slice up a block of swisse cheese and realign, the holes don't normally line up and we g
- But every now and then, the holes line up and we are vulnerable

OWASP Top 10

- **iirc** he says that “we should know this”, so we should probably have the list here for 2017:
    - Injection, broken auth, sensitive data exposure, XML external entities, broken access control, security misconfiguration, XSS, insecure deserialization, using components with known vulnerabilities, insufficient logging and monitoring



T10

OWASP Top 10 Application Security Risks – 2017

6

A1:2017-Injection	Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
A2:2017-Broken Authentication	Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.
A3:2017-Sensitive Data Exposure	Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.
A4:2017-XML External Entities (XXE)	Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.
A5:2017-Broken Access Control	Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.
A6:2017-Security Misconfiguration	Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.
A7:2017-Cross-Site Scripting (XSS)	XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
A8:2017-Insecure Deserialization	Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.
A9:2017-Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.
A10:2017-Insufficient Logging & Monitoring	Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

Security Engineering - Assets

- Security is made to protect our assets
- Sometimes we protect the wrong assets
    - o e.g. if you have a front door well protected but no protection on the glass windows or the back door

Identifying assets – what is that you're trying to protect

- It is easy to identify the wrong thing
    - o Richard car-burglar and AIDS story…
        - § Don't protect your car over yourself…
        - § Not getting aids from a syringe > $5 wallets
    - o Richard window smashing story…
        - § Are you protecting money or window? leave your window open.

Coke

- Pretend the asset is the secret recipe
- But the real asset is the reputation/brand
- People prefer to drink coke even though they prefer the taste of Pepsi

- It's important to identify the right things to protect
- It's easy to protect the wrong thing
- Uni's asset review
    - Machines + Computers were mainly identified but didn't considered...
    - Reputation also needs to be considered
    - Students + Staff
    - Private user data

Strategies for Asset Protection

- Multiple pairs of eyes - ask lots of people
- Standards and protocols tend to miss more than they catch
    - A good starting point - shouldn't be completely ignored
    - Examples of standards are what NIST recommends, Information Security Forum (ISF) came up with some best practice rules. None of these are mandatory per say just a good starting point.
- Don't think you've done everything
    - There is always a blind spot or new category to consider
- Regularly surveying the values of people involved in what you are protecting
- Develop a sensible plan - people suck at knowing what they want. Need to ask the right questions to tease the information
- Constantly re-evaluate current list of assets

Categorising Types of Assets:

- Tangible assets: Those that are easily given a value (e.g. jewellery)
- Intangible assets: These cannot be easily and objectively valued (Customer info, company secrets)
- Monetary + psychological/emotional costs
- Difficult != Don't do

Strategies for assigning values to assets:

- Survey what many people think
  - No single person/group should be solely evaluating the assets
  - Examples of the info. that should be gathered are as follows:
    - "How much money would you lose were this data centre to go down for 24 hours?"
    - "How much will you lose if your company is disconnected from the internet for 3 hours?"
- Examples:
  - In assessing the value of a park
  - Picasso

Diffie Hellman works to establish confidentiality and integrity but how do we get authentication? If we use public and private keys on their own we are susceptible to Man In The Middle attacks. So what can we do?

- We could use Public Key Infrastructure (PKI). It is like a passport in that it links the public key to a domain name which is certified by a 'trusted' group (signer - certificate authority). It needs to know the signers public key which is often installed in your computer automatically.
- This PKI system is not foolproof you can get variations of legitimate sites (Gooogle instead of Google for example) approved because the signers have an incentive to agree to sign things because they get money that way.
- From Threatpost article on weakness of PKI "Researchers said certificate abuse boils down to three types of weaknesses in the code signing PKI: inadequate client-side protections of certificates, publisher-side key mismanagement, and certificate authority-side verification failures." https://threatpost.com/assessing-weaknesses-in-public-key-infrastructure/128793/
- PKI can let malware through (Stuxnet)
- Pretty Good Privacy (PGP)/Web of Trust is another way to deal with the problem of authentication.
- SSL / TLS diagram of it

PKI Weaknesses

- Single point of failure
  - CA (certificate authority) can verify someone who is malicious or is impersonating another entity - e.g. InfoWorld article
  - A Root CA certifies a CA that then issues a fraudulent certificate
  - LetsEncrypt provides free SSL certificates. Lowers the barriers for individual web developers but also for attackers
- Padlock - can provide a false sense of security now that many sites (legitimate and phishing) use HTTPS
- Users can easily proceed if a browser presents an SSL (Secure Sockets Layer) warning for an invalid/expired certificate etc

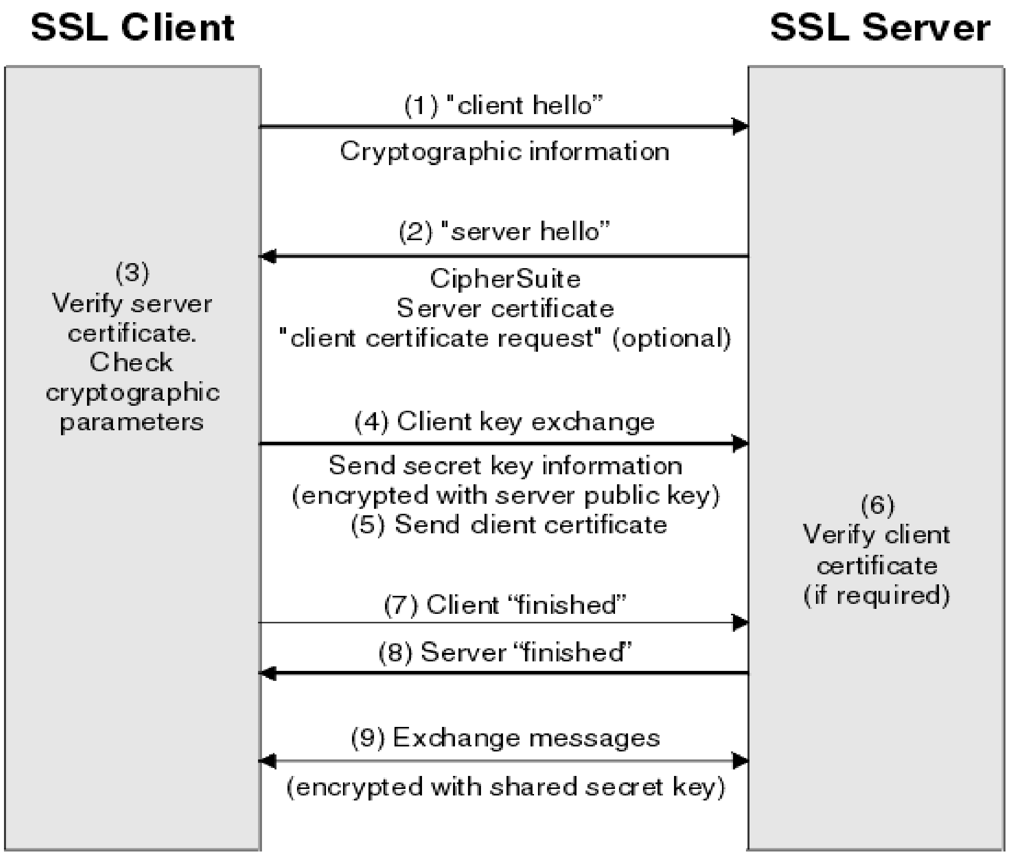
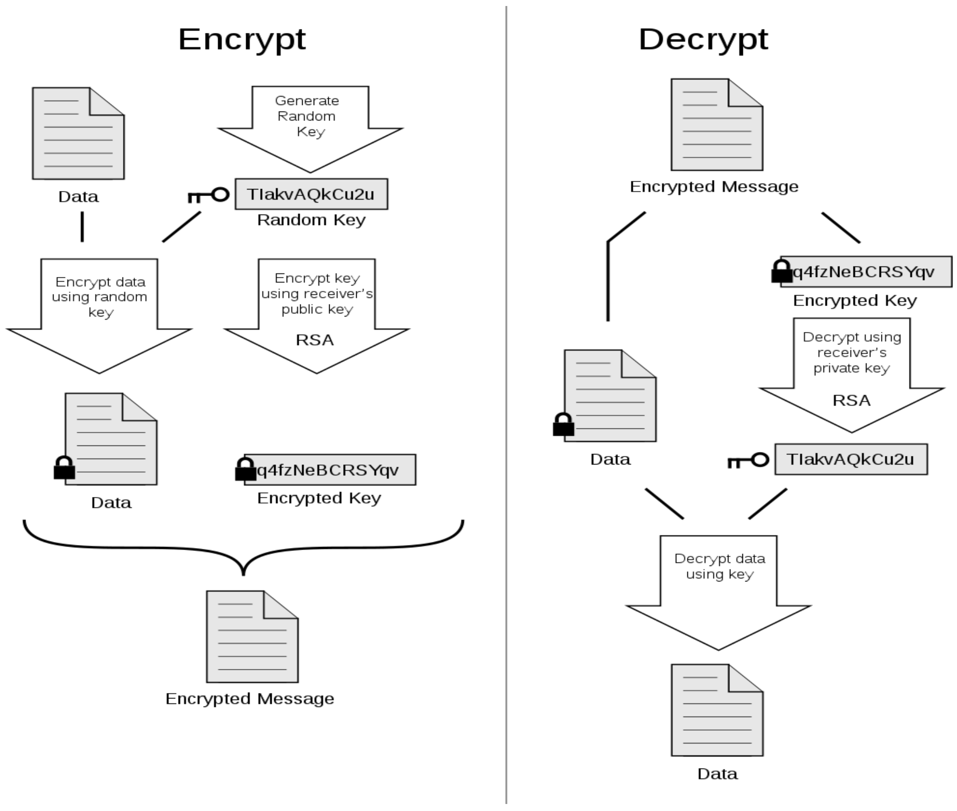

SSL Client
SSL Server
(1) "client hello"
Cryptographic information
(2) "server hello"
CipherSuite
Server certificate
"client certificate request" (optional)
(3) Verify server certificate. Check cryptographic parameters
(4) Client key exchange
Send secret key information (encrypted with server public key)
(5) Send client certificate
(6) Verify client certificate (if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

DIAGRAM OF PGP

Web of trust



A diagram of PGP

TLS (Transport Layer Security) Handbook Example:

1. A client contacts the server
2. The client and the server exchange information about the communications they intend to perform, such as the ciphers to use (SSL handshake)
3. The server transmits its certificate to the client
4. The client checks that it trusts the certification authority that issued the certificate. If it does not recognise the CA and does not get an override, the communication ends
5. The client checks for revocation information on the certificate. If the certificate is revoked or revocation information is unavailable, then the client might attempt to obtain an override. Implementations vary on how they deal with null or unreachable CRL information, but almost all will refuse to communicate with any entity using a revoked certificate
6. Both client and server send each other random data, which they can use to make calculations separately and then derive the same session keys. Three kinds of randomly generated data are sent from one side to another.
    - The "client random": This is a random string of bytes that the client sends to the server
    - The "server random": This is similar to the client random, except that the server sends it to the client
    - The "premaster secret": This is yet another string of data. In some versions of the TLS handshake, the client generates this and sends it to the server encrypted with the public key; in other versions, the client and the server generate the premaster secret on their own, using agreed upon algorithm parameters to arrive at the same result.
7. Both parties generate 4 session keys using this data
8. All communications in the same conversation are encrypted with that set of keys

Bug Bounties

- Crowd-Sourced Bug Bounty Websites
    - Public: Hackerone, bugcrowd
    - Private: Synack
- Often have criteria of whats in/out of scope, as well as what kind of bugs they won't accept. For example websites that they don't want you touch
- Tips
    - Learn web apps
    - Use a wide scope → bigger net = more bugs
        - Stay in the scope
    - Look for software updates, or assets that have recently changed
    - Look for publicly disclosed reports → Can see prior bugs that have been found/exposed. If a bug has occurred once, there's a chance it will occur again
- Process:
    1. Find a program
    2. Review scope
    3. Find target via Recon
    4. Hit and find vulnerability

    5. Write a report
    6. Submit it
- Fuzzing
    - **fuzzing** is the usually automated process of finding hackable software bugs by randomly feeding different permutations of data into a target program until one of those permutations reveals a vulnerability
    - Automate process - a program that continually adds input
    - Some fuzzers are aware of input structure, and some even are aware of program structure
    - Fuzzers aren't precise, but can test a large amount of inputs
    - Ideally you would use both the Fuzzing and human-written tests, but that often doesn’t happen.
    - Fuzzing software - afl (the way to go apparently)
- Mutation strategies - bit flips, byte flips, arithmetic, havoc (combination
- Use fuzzing to test your own software
- **Homework: Do the fuzzing tutorial**

Penetration Testing

- Why is it important?
    - It allows us to discover vulnerabilities in our system before attackers do
    - It can test you security controls(Firewalls, IPS,IDS)
    - Sometimes thinking like an attacker is the best way to expose weaknesses
    - Digital security in today’s age is everything
- Authorized simulated attack on a computer system to evaluate security risks
    - Performed to identify strengths and weaknesses
- Steps of pentesting
    - 1. Recon
        - Intelligence is gathered
        - Pentester finds potential vulnerabilities in the system
    - 2. Planning
        - Plan how you are going to execute exploits
        - That payloads are going to be used
    - 3. Exploitation
        - Put gathered intel to use
        - Exploit vulnerabilities
    - 4. Post exploitation
        - Establishing persistence
- Certification
    - Offensive security
        - OSCP (Certified professional)
        - OSWE (Web expert)
        - OSEE (Exploitation Expert)
    - Kali Linux training

- Pentesting tools
    - Metasploit

        - Provides information about vulnerabilities and exploits of many different systems
        - Your antivirus will go nuts with the installation
    - Burp
        - Scans websites for vulnerabilities
    - Wireshark
        - Network protocol analyzer
        - See what’s happening on the network
    - Kali
        - Linux OS containing many security tools
    - Nmap
        - DANGEROUS
        - Perform security scans
        - Network discovery and security auditing
            - Discover hosts and services on a computer network by sending packets and analyzing the responses
        - Finding open ports on remote machine
    - Gobuster
        - Brute force
            - URLs in websites
                - Directories and files
            - DNS subdomains
- CTF Websites
    - Pwnable.kr
    - Hackthebox.eu
    - Root-me.org
    - Overthewire.org

Week 8

Leave this sippy cat in to give determination in the final exam

When something goes wrong - What is the root cause?
- All cyber issues are due to human error (not necessarily 1 person, could be many)

Root Cause Analysis
1. Blame everything on the last person to touch it
   - Old airline industry blamed the last engineer to sign off
2. Culture - the culture of a company or organisation is to blame
   - Difficult to change
   - Doesn't really tell you what to change
   - An easy thing to throw money at so that it looks like progress is made: Saves reputation rather than fixes root cause
3. The whole system; it was a "normal accident" -- the system was too complex/badly designed and something going wrong was inevitable. That things went wrong in this particular way was just bad luck.

Human Weaknesses

Honesty
- Commander in Cheat - book about how Trump cheats at golf with his tiny hands
- 6
- Humans lie
- Universities get students to sign honor codes·
  - They did a study to check effectiveness

        - Results: people were more honest when they signed the honour code before rather than after filling out information
    - Richard didn't jaywalk for the first 10 years of his kids lives because jaywalking while telling them not to jaywalk would be dishonest
    - Richard didn't lie about Santa 'worked around by omission' but ended up telling them the truth and it spoiled his kids' fun :(
    - People lie to themselves about their own honesty
    - When asked who contributed to these lecture notes, people will honestly say they helped, but *will people lie about contribution? ooft*

Misdirection and limited focus

- 'Chekhov's Gun' is a concept that describes how every element of a story should contribute to the whole. It comes from Anton Chekhov's famous book writing advice: 'If in the first act you have hung a pistol on the wall, then in the following one it should be fired.
- people are bad at picking the right features for focus
- We often look for a few factors, in a very large space (needle in a haystack)
    - Attackers try to get us to focus somewhere else
- logically important vs psychologically salient
    - People SHOULD focus on what is logically important
    - People USUALLY focus on what is evident

similarity matching

- is when people look for similar situations that happened to them in the past that they are familiar with and applying it to the current situation.

Frequency gambling

- If many patterns match, you pick the one which you have the most experience with
    - A natural reaction - if gravity works 99 times it becomes plain it will work the 100th time
    - Note: this is the current proof of general relativity
- What worked in the past will hopefully work in the future
- Not always the best solution. Especially for new problems, which happens often in Security

availability heuristic - Kahneman 3

- The availability heuristic is a mental shortcut that relies on immediate examples that come to a given person's mind when evaluating a specific topic, concept, method or decision

Satisficing and bounded rationality

- Satisficing is only doing good enough, rather than doing perfect
- Bounded rationality refers to how we have a limited amount of information, meaning our ability to make decisions is bounded by that information

Tendency to verify generalisations rather than falsify

People prefer positive statements

- we ignore what we don’t like, people convince themselves that they are right when the evidence don’t exactly line up.

Cognitive strain

- *Making multiple mental calculations*

Group-think syndrome

- How people think when they’re in groups rather than as individuals
- We prefer to keep the peace in a group rather than fight against collective ideas
- The result is groups become homogeneous
- For example: leader makes a joke in a group and everyone laughs even if it isn’t funny
- Good for analysing systems

Confirmation bias

- We prefer the evidence that confirms what we believe
- Is an example of cognitive bias and describes that people gather and recall information selectively/interpret in a selective manner
- Richard is very smart and very good at orienteering

Accidents vs Attacks

- Intent is the main difference.
- In an accident, ‘holes’ don’t really line up
- In an attack, the attacker will make those holes line up

Error

- Human error is inevitable - how do we fight it?
- 3-mile island - an example
    - Shaving the buffalo I think
    - A situation where an infinite number of things go wrong causing some undesirable situation
    - To fix the situation requires fixing about 100 different things
    - For example, late to work because of traffic because of a bus strike because you left late because you had to make coffee because a meteor struck your coffee pot because people are doing in-atmosphere asteroid mining because people like shiny things etc
    - In the real world 3-Mile Island situation, multiple parties blamed multiple other parties, mostly the operators
    - In reality, it was *everyone* and *no-one*’s fault - sometimes everything just goes wrong
- The point of the 3-Mile Island example is that fixing the problem isn’t about finding a single point of failure, maybe it’s just fixing everything you can, a tiny bit

- We have to design things so that when they go wrong, which they will, the impact is limited - assume you're going to be breached and set it up so that it's not a disaster
- **Simplification:** people will simplify situations to have only one cause, when the truth is probably a lot of causes
-

Error proofing systems:
- Luck doesn't cut it when it comes to building systems.
- Always assume that the person using the system is going to do whatever they can to attack it.
- Always assume that the environment of a system will be the worst case scenario

Common Mode Failure:
- related to redundant systems where one cause can lead to the failure of otherwise redundant elements leading to system failure.
- Elements which should fail independently are under some circumstances dependent.
- E.g: Indicator Failure (Too Many Alarms), Change Over System, Repeated Errors, Common Paths

Just Culture
- Don't just punish the person who made an error, or the last person who touched it
- It's about learning and stopping these situations in the future

Just Code
- Complexity bad
    - No one component should be too complex or too empty
- Coupling bad
    - You can change one component without having to rewrite everything
- Cohesion good
    - Components that are close to one another make use of each other
    - Components that are far away from one another do not make heavy use of one another
- These together create defence in depth
    - A good system can fail at one point or another, but if it doesn't fail at EVERY level, the system itself isn't faulted (yet)

Systems that follows these rules are easier to maintain, and more resilient to attacks

Cassandra Syndrome:
- knowing the truth but no-one believes you
- Occurs when valid warnings or concerns are dismissed.

Chekhov’s Gun: anything on-screen in a movie is there for a purpose

- A gun is on a wall because it will be used
- A person coughs because they have Everything Disease
- Useless facts aren’t included
- Believe event only has one significant cause
- Plan for fewer contingencies than occur
- Ability to control outcomes - illusion of control
- Hindsight bias
    - Knew it all along phenomenon
    - perceive events that have already occurred as being more predictable than they actually were before the events took place
    - illusion of control
    - plan for fewer contingencies than occur
    - Knowledge of outcome of previous event increases perceived likelihood of that outcome
    - Complexity, coherence, coupling, visibility

Distinguishing between latent errors and active errors

- A latent error is an error that is present but not detected; consequences will occur later
- An active error is an error that is present but detected
- <u>Latent errors undermine defense in depth</u>

Homework:
Learn about

- **Chernobyl**
  Power test + unstable (graphite tipped boron rods) + too vigorous 1% power boil + no automation + steam pressure + rod stuck + increase power + sad lyf → cancer + death + air spread
- **Bhopal**
  Pipe clogging + water + MIC tank + boom + money (obv) → union carbide + no supervisor
- **Challenger**

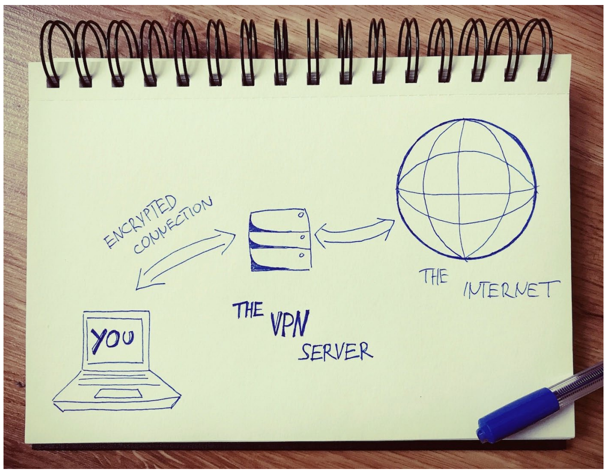
Focus on why things went wrong, where the focus was and wasn’t
There will be a generic question about an accident we learned about in the exam

Privacy

- We sacrifice privacy for convenience and fun
- When private data is compromised it is dangerous because attackers can use it against us
- Signal Blocking on Phones

- Police have been known to access metadata on phones without warrant, so signal blocking your phone may be essential to journalists (who might not want the police seeing their phone data)
- Wrapping a phone in foil blocks bluetooth signal, but who knows if it blocks anything else
- Faraday cage bags work on the same principle - dispersing incident electromagnetic radiation, making the phone inaccessible

- Online Privacy Practices
    - Incognito mode
        - Deletes cookies
        - Not very effective
    - Use privacy focused browsers
        - Eg DuckDuckGo, an alternative to Google - doesn't track personal data and requests like Google does
        - This info brought to you by Google
    - Take care of your accounts
        - Log out when you can
        - Don't link accounts or services unnecessarily
        - Lie
    - VPN
        - Virtual Private Network
        - Acts as intermediary between external servers and you
        - Encrypts all traffic between you and the VPN servers
        - Meaning external entities see your VPN, not you
        - Examples include nordvpn, and some other bad ones speaker doesn't advise using



    - Onion routing
        - Encrypt your data and obfuscate its origin by forwarding through multiple nodes
        - Outgoing packet is N-times encrypted, packet visits N nodes on its journey, each node it visits in the network decrypts a layer, like peeling the skins off of an onion
        - Desired location then gets the unencrypted packet
        - Weaknesses
            - Logging into accounts like fb identifies you, gets rid of anonymity
            - Timing attacks
                - Cross reference when nodes receive and send packets to identify which sent packets correspond to packets which are received

- "*Arguing that you don't care about the right to privacy because you have nothing to hide is no different than saying you don't care about free speech because you have nothing to say*" - Edward Snowden
- Why should I be concerned?
    - "I have nothing to hide" just means "I have nothing I can think of that I want to hide"
    - The problem is an imbalance of power between citizens and the government
    - The issue of privacy is not one of a large flood of your data, it's the slow trickle of your information over time
    - "It's not about individual data it's about a pattern of data"

Digital Forensics

- See: COMP6445, there's still a few places left
- What is Digital Forensics?
    - Branch of forensics science
    - Recovery/Investigation of material found on digital devices
- Stages of an Investigation
    1. Acquisition/Imaging
        - Capture an "image" (duplicate) of the drive
        - Chain of custody
        - Need to preserve original data, so use write-proofed connector to data storage module to prevent data corruption
    2. Analysis
        - Keyword Searches
        - Look for data
        - Recover Deleted Files
        - Specialist Tools Used
        - Browser history, documents, pictures
    3. Reporting
        - Evidence used to construct events/actions
        - Compile data learned
        - Layman report written
- Types of forensics
    - Computer
        - Memory (Ram)
        - Data
    - Mobile
        - Contacts
        - Messages
    - Network
        - Routing table
        - Switches
        - Packet Capture
    - Database
    - Video/Audio
    - Steganography

- Tooling
    - Encase - used to duplicate files or analyse it
    - Autopsy/The Sleuth Kit - analyse different file systems
    - File and Strings - UNIX commands that give file type and ASCII strings
    - xxd - Hex dump
    - foremost -recover files by finding headers, footers and internal data structures of malformed files
    - Binwalk - finding embedded files
- Drives and partitioning
    - Allocates memory in clusters
    - The FAT (file allocation table) record stores status of each cluster (bad region, allocated, unallocated)
    - When a file is deleted, it is not fully wiped out. The name of the file is changed to 0xE5[file_name].
    - Forensic analysis tools will check for this prefix in the file allocation table, and if it is there, they can extract the contents because nothing else has been removed.
- When data is deleted, the electrons aren't destroyed, the pointer to that location is just forgotten
    - Meaning we can still access it if we have a very good microscope
    - For more on that try COMP3231, very interesting course

Chain of Custody: protecting forensic evidence from being changed while it's being moved from the crime scene to the place of analysis

Secure Systems: 3-Mile Island

- An island
- 3-miles long
- 0 miles wide making it a 2-dimensional shape
- Nuclear reactor - simple nuclear reaction to heat water to spin a turbine
- The red part is the nuclear reactor chamber
    - It's super hot and submerged in water (or molten salt, whatever floats your salt boat)
    - It's meant to be abstracted away from the rest of the system
- The yellow and blue systems interface with the red one, are heated by the hot liquid in the red, and are used in turbines to turn them
- After the failure of the 3-mile island (unit 2), a lot of focus turned to it
    - Runners of the plant blamed the builders
    - Builders blamed the runners (management) I think
- Transfer of heat from primary to secondary to prevent core overheating
    - From red to yellow/blue, but someone deleted the diagram *wasnt me i swear* do you know you're a purple badger
- https://www.antipodesmap.com/
- Attacking people with nuclear weapons is against the course policies
    - Remember: do not be a dick
- A cupful of water leaked out of the cooling non radioactive water system into the pneumatic system
    - The pneumatic system drove the instruments which were used for monitoring
        - Water got into the pumps for the feedwater system and made the instrumentation give crazy readings
        - Feedwater pumps were auto shutdown to prevent damage
        - Turbine was shutdown
        - Core couldnt be cooled because turbine was shutdown
        - Secondary backup pumps were started to cool the core
        - The backup pumps weren't working because the valves to the pipes into the cooling system had been left closed after routine maintenance 2 days before
        - 2 indicators on the control panel showed the status of the valves
            - No one looked because they didn't expect it
            - 1 of the lights was obscured by a repair tag on the switch above
    - SCRAM protocol: do your best then run like heck
        - Drop the graphite rods and SCRAM
    - Relief valve should open to allow gas to exit to prevent explosions
        - It failed to close after it opened
        - The indicator for the valve failed and the operators were misled into believing the valve worked
    - So they flooded the reactor which is bad because it stresses the system
    - Then an automatic high pressure injection system came online
        - The operators turned it off because they were being told everything was ok by the indicators

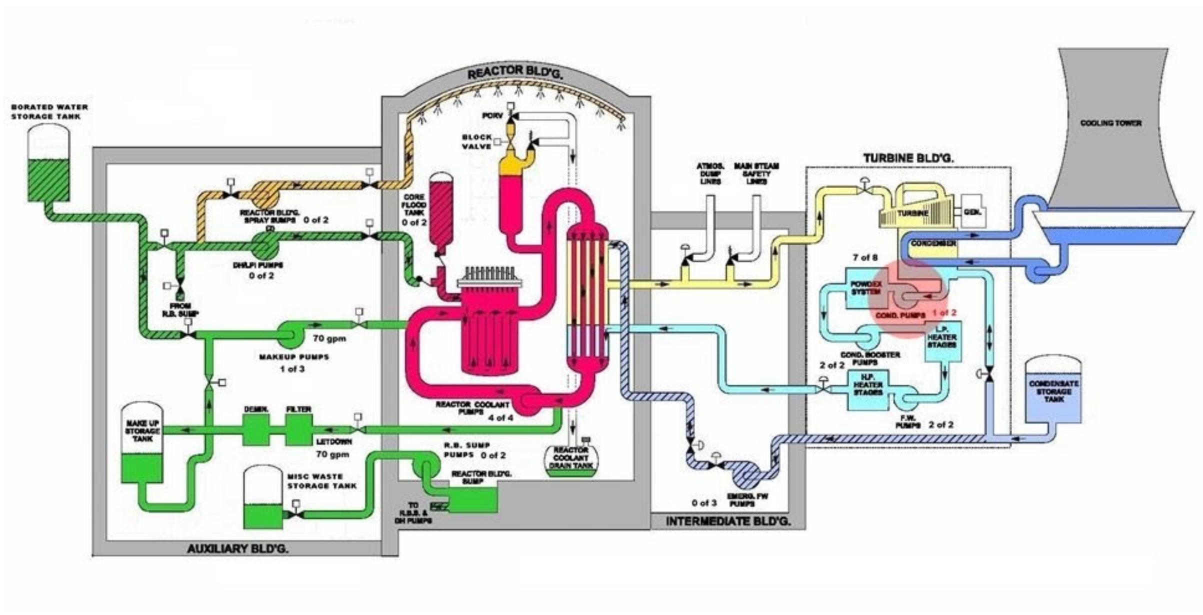
        - The reactor core was becoming uncovered which is VERY INCREDIBLY BAD
        - **The water in the system started turning to oxygen and hydrogen which is very bad because that's the recipe for a bomb**
    - China Syndrome: when a nuclear reactor is so hot it melts through the crust and “goes to China”
        - Realistically, this happening would irradiate the magma in the earth under the melting site, causing nuclear volcanoes lol *magma is already radioactive naturally, wouldn't make a difference* I have to see sources on that m8 https://www.google.com/search?q=how+radioactive+is+magma&oq=how+radioactive+is+magma&aqs=chrome..69i57.3750j0j1&sourceid=chrome&ie=UTF-8
        - *Theres alot of magma and not much nuclear material*
        - “Yes, and so are you, and maybe your kitchen counters as well. Traces of radioactive elements are found in many places. Carbon is a key element of all living things, and a small percentage of all carbon is the radioactive isotope carbon 14. Living things also contain potassium. Some of the potassium in your body is also radioactive. Your kitchen counters might be radioactive because granite contains traces of radioactive uranium and thorium. Lava is about as radioactive as granite rocks.” - Quora answer, https://www.quora.com/Are-magma-and-lava-radioactive
        - Tl;DR magma is radioactive, but not as radioactive as bananas
- This bit is very dry, unlike the nuclear power plant, which is wet with radioactive water
- Nuclear reactors, like technological systems, are complex, and if we look at how they’re ‘failproofed’, it can shed light on how to design a secure system
    - Specifically methodology, eg how failures are discovered and buffered against
    - The system is tightly coupled but the problems were not directly related which made it hard for operators to identify the issues
    - Richard starts talking a bit quickly here, but the point is this:
        - In a complicated system you have high coupling
        - Therefore changing something in one place can do random dumb stuff in an entirely unrelated part of the system
        -

SYSTEMS THAT ARE TIGHTLY COUPLED AND OPAQUE AND COMPLEX ARE ALMOST GUARANTEED TO HAVE PROBLEMS

THINGS YOU CAN DO
- Identify the most important things to protect and focus all your energy on those
- Assume you’re going to be breached and work to make sure that that won’t be a crisis
    - Discard all the valuable data, make the system unattractive to attack

As more water flows in, it’s possible for reaction to break water down into hydrogen and oxygen.

hydrogen gas is highly combustible especially when mixed with oxygen => this is why water beats explosions, because it's made of explosion
Reactor was producing oxygen/hydrogen.
At some point oxygen gas combusted.
Tank began to fill up with hydrogen (one spark can cause a large explosion)



Week 9

COURSE 👏 REVIEW 👏

Regarding the course textbook provided in the exam

- Please don't have a specific answer to questions
    - Richard will have to make the exam harder if you have too much information in the textbook
- Also avoid putting TOO MUCH detail
    - This will make it hard for students during the exam to read and understand what you've written under pressure
- Make sure to proof-read and check for errors

The movie, 'The China Syndrome' will be in the Exam

Just Culture

When something goes wrong, instead of acknowledging the root cause, you simply:

- Instead of blaming the individual who was the approximate cause of the problem, you look at the whole system, and figure out what needs to be changed
- When Qantas brought **'Just Culture'** in, it reduced their accidents by 50%!
- Blaming the individual: self gratification
    - The problem with that is that you'll simply encourage the cycle of firing people if the system is broken
- 'Don't shoot the messenger'
    - No one wants to be the deliverer of bad news (since they usually get the blame for it; i.e. Darth Vader)
- By analysing the problem, you're more able to discuss clearly and figure out the root cause of the problem

Look up: how the NHS used it in England

Security Engineering

Recall back to the first question Richard asked:

What is it to be secure in an Engineering sense?

> What are the properties that we can learn from Engineering and apply to Security?

1. Learning from the past

- This has allowed us to fail and learn from them
- Reflecting on past experiences

2. Methodologies to follow

- For example Civil Engineering and building a bridge:
    - Australian Standards**
    - What materials to use
    - Reports to write///

Engineering has this notion called *"Best Practice"*, where whenever someone has a good idea, we add to this. This is like a **Checklist**, which is useful because it **stops you from overlooking things**.
Google: the Tacoma bridge

People don't notice things
- If you missed something once, it's very hard to see it the following times
- Fresh pair of eyes is more successful
- Research says: if you miss something you'll find it difficult to spot it the next time

That's why pilots have checklists
Checklists are just a baseline, there are possibilities of things happening
- So engineers need to have processes to be creative
- That's why we don't just use robots who are very good at checklists

3. Redundancy and Testing

- Defense in depth
- Dual control: multiple things have to vote on the same thing (parallel)

Carl Popper: a scientist is always someone who is trying to prove themselves wrong
- It's not what is blindly following the procedure that is science, it's trying to **falsify yourself** (you're following it to prove yourself wrong)
- That's why we like theories that can be proved wrong

The same thing occurs in Engineering
- We test things to make sure things don't go wrong
- Such that we keep the logs and adapt principle from the results
- So that we break things in the lab

4. Engineers take pride in what they do

- They wish to do a good job
- They revel in craftsmanship

Even if your Product Manager is constantly saying *“SHIPSHIPSHIPHISHIP IT”* but you feel like the product isn’t ready yet, an engineer with pride in their position will stand their ground and say no. By rushing to send something out, you usually drop the most time-consuming thing, which is **Security**.

5. Focus on process

- Understanding the process rather than what you’re building

6. Review

- Review your process
- Review your tests
- Having others review your work (independent, peer review)
    - Removes bias and blindness to own work

7. Professionalism

- You’re doing this job because your duty to the profession
- This is different to your duty to your position (i.e. your company asks you to do something)

Conflicts of interest

- Plan for it and address them
- Have systems in place to deal with this

Quantify things

- Come up with metrics and such
- Have specific meaningful numbers so you know
- They should be tested and understood

“Closing the Loop”

- So many systems are simply just ‘feed forward’ (like high school)
    - There’s no feedback coming back
- Closing the loop is simply the idea of **checking that you are right**
- The Art of being an Engineer is the Art of Closing the Loop
- We don’t want wishful thinking, we want **WISE thinking**
- This means that you are often listening for feedback

Aerospace Safety Conference
Story: Accident on the oil rig

- 350 separate alarms went off
- Oil spilled, started fires
- Of those 350 alarms that went off, he surmised that only about **8 of them were important**
- So if you imagine that you're in a control room, and 350 alarms are going off, it's very hard to figure out what's going on
    - What are the key ones?
    - This is what Security Engineering is about

System Properties

- It's easy to be reactive like the barbarian who boxes
- What you need to do is understand and think about systems
    - What goes wrong
- E.g. pilot leaves a wrench in a socket when the airplane leaves
    - Yes that is a mistake
    - However the fault comes from the system which allows people to make mistakes
        - This is like the barbarian
    - A problem will arise next week and someone else will have to go down for it

Russia sub in Cuba

- Needs to surface for air
- Radio signals blocked out
- They're allowed to launch nuclear missiles because they're the last line of defence if things go to poop
- When they went down the last they knew was
- Needed a unanimous vote
- One commanding officer was onboard who kept voting against launching
    - Credited to saving the world

What is Coherence?
Coherence: Everything in the unit has to be doing the right thing

- Everything that is related comes together (Object-Oriented principles)
    - Coherence is a system property we want
- Don't want it to be complex
- Want dont want the system to be tightly coupled
    - Then changes in one affects changes in many others

Youtube: Most unexpected gold medal in history

- Example of a tightly coupled system
- System had a common node of failure

Wargames clip

- Why are humans in the loop?
- Can we replace humans with ASD

        - Automatic decision makers
- ASDs don’t always work (failed in
    - It’s difficult to cut the head
- What if there’s a different problem that happens in which the designer didn’t design the system for?
    - Attackers attack where the system isnt designed for
        - They put the system in a state where the preconditions are not right
    - This is what security engineers are interested in
    - Systems have flaws and vulnerabilities
    - HUMANS design these systems

We need humans in the loop
- All these films are the consequences of such
- So we need to understand **SYSTEMS** as well as **HUMANS**
- So we need psychology, anthropology

Society itself is a system
- The system can be hacked
    - Russians in election
    - Brexit interference

Security is **END TO END**
- In software things are secure in the middle
- Just like the fence (fence post abused, or missing at the end)
- You need it secure all the way around
- E.g. typing password into system
    - Typing the password in is END
    - Attack vector is keyloggers

Work to undermine limits

RESEARCH: How the Romans and the Greeks transferred power
- Anyone in power works to increase their power and subvert the system
- As soon as someone is in power, they work to undo the systems that restrict their power
- Key idea is ensuring that checks and balances exist

This is just like the wereWolf (Buffy da Vampire Slayer)
- Every night they know they’ll be turning into a werewolf
- So they lock themselves in before to prevent them from attacking anyone
- werewolf spends the time trying to get out and undo the restrictions (cage)
- So we’re hoping that the cage (system) is strong enough to stop people in power from undoing/overcome these systems

Multilateral security designs

- Like the Bell-Lapadula System
    - Person on top controlling the rest
- If the person on top goes crazy you’re in trouble (Single point of failure)

Assange and Free Speech

- People fighting for free speech
- But trying to lock Assange up
- You want free speech, except that is against you (Conflict of interest)

Cars and Trolley Cars

- When Google/Uber went to parliament and talked about Self-driving cars
- They did some misdirection
- Someone asked the question “What if it the system gets hacked?”
    - Yes there are dangers
    - But we have top people working on it
    - We have men on the moon and planes
    - Not an impossible problem
    - “Just trust us”
    - Distraction 1: probably sometimes cars will kill people
        - But if you look at the road, people die all the time

        - Random statistics about car deaths as a result of human error
    - Distraction 2: the really interesting thing is the decisions that the cars are going to make in an ethical decision
        - Ethical dilemmas are so hard we really don't know the answer to that question (the trolley cart problem)
- Let's not think about the hacker problem (MISDIRECTED)

How can cars be safe, if one of the biggest software companies in the world can't fully secure their phones, email systems, etc.

- If it gets hacked, they have control over the system
- There are systemic flaws
- How do the cars know what the speed limit is?
    - Centralised data-base from GPS can be hacked and changed
    - What's the end-to-end secure solution?

How to ace the exam:
good question! this reminds me of a really interesting problem to do with the number 7

No trusted third parties

- There are bad people everywhere
- If you have an M&M security you're at risk to insiders and such
- Security is all built on trust (Things work and operate on trust)
- If you can't trust anyone you need walls everywhere
- This breach of trust is the real problem
- So we set things up so when things do fail, they fail small (minimal impact)

Ideas for your future from Richard

1. Community
    a. Help each other
    b. Listen and learn
2. Self-directed learning
    a. Learn for yourself
    b. Think about what you need to do (be critical)
3. Exercise
    a. Practice, practice!
    b. If you don't exercise you get fat
    c. Don't just do what you're assigned to do
4. Professionalism
    a. Do what is right for Cyber Security
    b. Grow and become professionals

Reversing & Cracking Seminar

WannaCry malware

- Encrypted your data, demanded payment in Bitcoin
- Marcus Hutchins reversed it and found the killswitch

What is reverse engineering?

- Taking an executable program and analysing its inside
    - E.g. constructing a recipe of an already baked cake

How to Reverse Engineer

- **Static**
    - Look through code (Assembly)
    - Ghidra
    - Binary Ninja
    - Radare2
    - GDB
    - Ida
    - Hopper
- **Dynamic**
    - Walkthrough the program during runtime
        - GDB is useful for this
    - Useful for obfuscated programs

Typically used together to maximise effectiveness

How2Reversing

1. Run the program first (don't do this with malware!!!)
2. Put the program in a disassembler (e.g. BinaryNinja)
3. Look at what the program is doing
    a. MOVs before function call is setting up arguments
4. Find exploit
5. Profit???

Tips

- Look at the big picture
    - Don't get caught in the nitty gritty
- Many tools let you modify and patch the code to test ideas
- Run beforehand where possible
- Work your way up
- Focus on areas of interest only
    - E.g. reversing a password, look for where the authentication is done, and reverse your way back up the call tree

Practice

- Crackmes/CTFs
    - PicoCTF
    - ioli Crackmes

    - Jazz’s crackmes
- Reversing programs on your computer like hello world

Youtube
- OA Labs
- Malware analysis for hedgehogs

Cracking/Patching

- A step after reversing (ILLEGAL - without permission)
- Patching is when you change the binary to make the software do what you want
    - E.g. removing the authentication step of the password

Further Knowledge

The Stack
- Grows upwards (except in Australia)
- Push, pop, EBP (base pointer for current stack frame), ESP (current stack pointer)
- Has entry and exit procedures (prologue, epilogue)
- Stack frames are like a contained stack for the function scope
    - This way stacks can be called and exited cleanly without manipulating the main stack

Preventing Reversing

1. Don’t Release the Debug Build!
- Remove symbol table from the binary
    - Gcc -s prog.c
- Dump symbols
- Disable asserts
    - Leaks information about the asserts
    - Gcc -DNDEBUG prog.c
- Watch videos of Tomb Raider on PS1

2. Trick the Diassembler
- Make the program as convoluted as possible
- Dummy instructions
- Excess jumps to make spaghet (**SOMEBODY STOLE MINE**)
- Overlapping instructions
- Self modifying code (runtime encryption)
- Jump halfway into add instruction (to hide the return call) <- embedded instructions

Main Lecture

“Canary in a coal mine”
- People used to take canaries in the coal mine in them
- The canary would die very quickly to indicate where the dangerous fumes are

- So it served as an indicator for danger

Opal is tracking your taps even when using credit cards
- Credit cards can now be used with DISCOUNTS!
- This means that they're tracking our identities

Companies collect data and sell them to brokers for a lot
- "They're de-identified of course"
- Our phones are constantly monitoring our data and sending it out
- 'NORMAL' PEOPLE CAN BUY THIS
- People pay for these surveillance systems (phones) and even more money to replace them

"Data needs to be free"
- They won't use their data
- They're talking about the data about you and me (the public)

Each individual bit isn't dangerous
- The aggregation of the data is
- These pools of data are extremely valuable

Surveillance is making it harder for CIA agents to operate
- Just pervasive tracking and the ability to data-match helps to identify and expose them
- Fake identities need digital trails because a person without one is suspicious
- Facial tracking is pervasive, especially in airports
    - Airports now have facial recognition for checking in and passport control

Week 10

Morning lecture

6841 ONLY:

- there will be exploitation in the finals
- answer k out of n questions, where one of them is an exploitation question
- use `python -c` in terminal if you wish
- tooling: stub command environment, gdb

Privacy lecture by Adam

Privacy in the modern age: we're stuffed

- best one sentence description
- a lot easier to execute security argument than privacy argument
    - What does this mean?^

this lecture is held under Chatham House Rule:

noun

a rule or principle according to which information disclosed during a meeting may be reported by those present, but the source of that information may not be explicitly or implicitly identified.

Story:

Adam and his sister both studied at UNSW, engineering and psychology respectively. One day Adam was listening to a podcast on the Stanford Prison Experiment for an ethics course (not his favourite). In 2009, Adam believes that it was more important that engineers and computer scientists study ethics than doctors and scientists.

"**Data is the new oil**" - coined 5-10 years ago

Data is refined and carefully protected just like oil, however people forget that oil is a huge source of danger. eg. storing too much, oil spills, long term unknowns. This leads to massive reputation damage.

Issues around data collection:

- **data linking**: network effect of linking data sources - 63% of the population can be uniquely identified by the combination of their gender, date of birth, and zip code
- **de-anonymisation**: all anonymised data runs the risk of being reversed. it is often difficult to draw a sharp distinction between personal information and de-identified data. Possible Bayes rule reverse inference.
- **massive scale**: data is being collected at a scale and pace that could not be imagined in other times
- **massive data-breach**: data is increasingly being disclosed in massive data breaches with serious impacts
- **phishing expeditions**: large and persistent datasets provide the temptation for phishing expeditions in the future

“There's just so much bloody data" - Adam
eg. location data - mac addresses, pinging towers
When we walk down the road, we are spewing information.

Roomba maps your house and uploads it to servers, Bose track your music history

bodies that collect Adam’s data:

- university
- government eg. tax department, welfare
- tech companies eg. bose, fitbit, spotify
- large tech eg. google, facebook

personal data:
- political leanings
- friend groups
- news sources
- health data
- communication history
- purchase history
- browsing history
- criminal records
- credit scores

bad if:
- government found out political leanings
- tax depot got personal docs
- insurance companies got health records

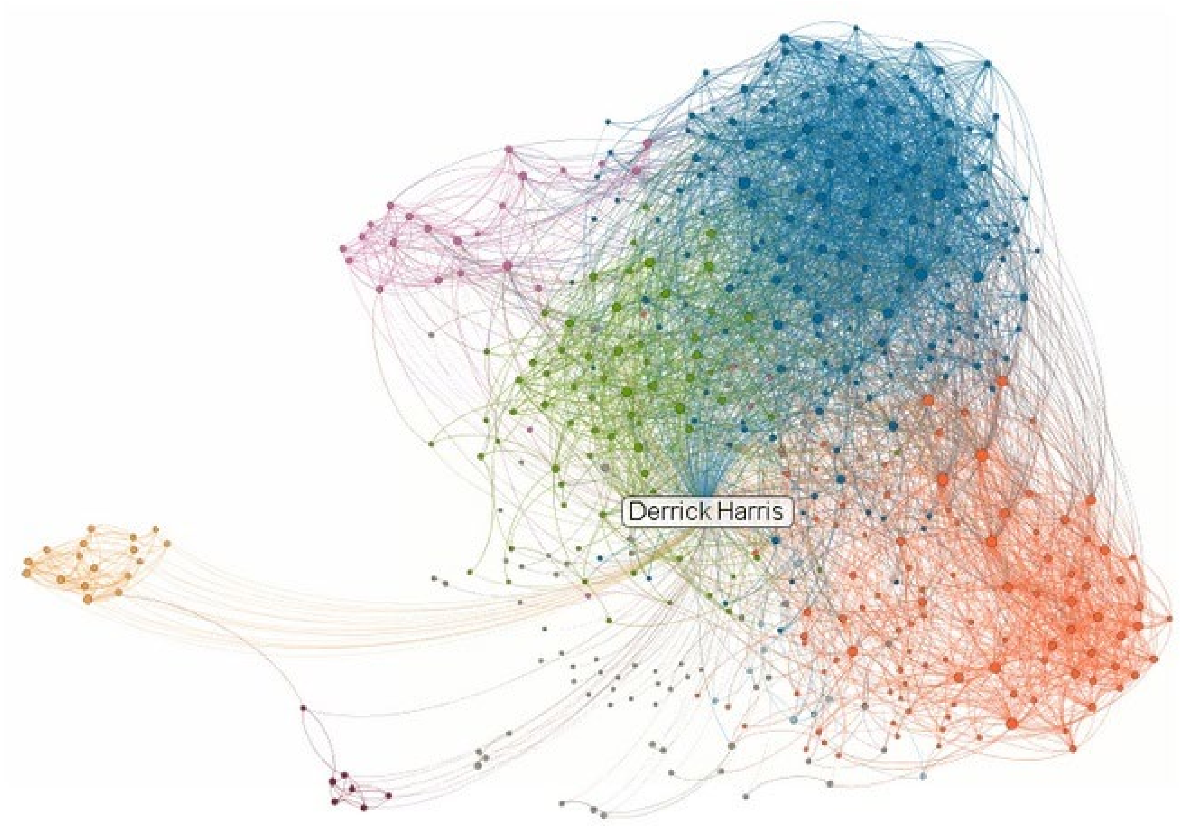
big tech companies have a lot of our data

Google Takeout
Warning: think before you download. Do you expose yourself to more risk downloading it? These tools weren't built for users, they were actually built for the government for law enforcement so that they could self select the data that they needed.

Adam went to a tech con at Bent's Basin, in the middle of nowhere. There was no reception (data or wifi) but Google was still able to track his location as shown in Google Takeout. He also found 5 years of search history.

iPhone data
Apple has a backup software which dumps all of your phone’s content onto your computer. It contains contacts, photos, notes, call history and text messages. Adam found over 87,000 texts since 2008.

Meta-data is incredibly powerful. The government probably creates a giant graph to link people together like below:



App data
eg. MS word - Adam found an old case study from COMP3441 (now COMP6441)
Tinder had an exposed SQLite database in backup which contained caches of conversations and unique identifiers.

Facebook
some of the things you can download:
- messages
- timeline info
- email addresses added to your account (including those removed)
- addresses
- check-in history
- ads you clicked on
- ad topics
- ip addresses
- active sessions

Facebook has a lot more data that doesn't appear in the list above. It is their intellectual property, but it is also information about us.

How can we use this data?
A day in the life of Adam - 13th July 2017
Using information from his iPhone, Facebook, Google, Opal card, Dropbox, and Tinder

web history - how to make a strong password, trello board
opal - eastwood to town hall, town hall to eastwood
photos + GPS data - 363 george street, qvb t2 store

what we know:
visited city after 1pm
attended a 'cyber' event
evidence of presentation
visited qvb stores
home at 7pm

The meta-data revealed a lot of information, but not all the information

---

Q&A
Incognito mode doesn't store browser history and cookies

Companies are using browser fingerprinting instead of cookies. When you visit a website your screen resolution, plugins installed and other information is sent to the site. Incognito mode protects you from shoulder surfing but not traffic inference, ISP, security (eg. phishing), tracking and shadow fingerprinting.

Under the Australian legislation, companies like Facebook have to hand over their information about you. They don't have to disclose the inferences made about you eg. race, job, marital status, sexual preference, political leanings

Data can be used against you
Bank data contains ID data eg. drivers licence or passport; and spending data

Privacy concern: risks around bank data being made publicly available
This data is retained for 7 years for tax purposes.
If you don't want your spending to be tracked, use cash

Adam suspects cash will be illegal soon. There was a legislation that stated any cash transactions over $10k will be made illegal. Unsure if legislation has been passed...

Companies collaborate closely with the government (banks especially)
Financial data is a huge source of information. Intelligence agencies collaborate closely with a bunch of companies to detect spies.

Having a VPN essentially means not trusting your ISP. It creates an encrypted tunnel to another computer and protects from snooping companies, man in the middle attacks. You can still get phished. It is not secure after the VPN terminates.

It's a BIG problem. Categorised into three sections:
- Private industry
- Government

- Intelligence community

Adam could readily find GPS coordinates of photos on Flickr!

Private companies:
data is the new oil, all going crazy about it
"data lakes" - streams coming together into the lake
run analytic reporting, make inferences

Aim: hoover as much data as possible and put it in one place

collect all the data, get it into one spot, oh no too much access restrictions, oh no it's encrypted

wrong attitude to take: "of course it's encrypted, of course there's access control"
walk back from this thinking
examine the possibility that nothing is perfect and done on time
normally people want results so non-functional requirements often scrapped (eg. privacy)

hundreds of companies struggling to keep their computers up to date
less than 50% of businesses have patching within a month
false to assume business will be up to date

what inferences can you make from a data set?
mental illness? divorce? pregnancy?

Quantium
data analytics, owned by Woolworths
massive data mining based on shopping habits
Every time you scan your Everyday Rewards card, it is pulled into a data set and the data ends up on your Facebook to target you
Facebook states that the company received "de-identified" purchase data

Coles
Flyby vouchers etc.
Adam's theory is that they only care about the first time you sign up because it links your credit card to a name, email, phone number

Qantas
Frequent Flyer Program - one of the biggest businesses inside Qantas
Qantas loyalty ($369m) > Qantas international ($327m)
The loyalty program has an estimated worth of $4bil and Qantas itself worth ~$9bil

Facebook:
not the best reputation in privacy space
eg. tampering of federal election in the states

politicians x tech companies

Workplace surveillance:
everything done on work laptop can be monitored
apps on mobile phones being used to track people
skype, hoover, slack
many instances where company found something to fire undesirable employee

unsw was tracking location throughout campus
access point - triangulate signal strengths
DNS requests are in the plain, unsw has access to this
whether they are storing it is another question

Issues around data collection:
often hear don't worry you can't de-anonymise it, it's not linked etc.
it's secure, it only does x under y condition
it's end to end encrypted
it's algorithms not people
it's locked down, only x can access it
we have thorough auditing in place

all this means nothing ^

eg. imessages - end to end encrypted, but icloud is not

There isn't enough public data sets to train AI on stuff eg. detection of nude photographs

Anonymising strategies:
- redaction
- encryption/hashing
- pseudonyms (unique identifier)
- statistical noise/'binning'
- aggregation

every one of these strategies is breakable
depends how determined the attacker is

4 data points in mobile cell towers enough to identify 95% of the population

Takeaways:
- anonymisations is not a certainty
- most data breaches had some sort of identifier between the records
- the pattern becomes the fingerprint, not the data itself
- to adequately anonymise data, might render the data as useless

The government:
- centrelink

- my health record - no one was asking for this; incredibly valuable data, probably selling the data back to drug companies and researchers
- opal: location history
- mygov: starting to link everything together
- drivers licence

Truism about government data collection:

1. Collection laws will emerge and re-emerge in different forms until passed
2. Collection is always going to increase in scope; it never diminishes
3. Terms of usage always start narrow, but quickly broaden
4. The least bad thing is no change in the status quo; most bad is robo cop scenario
5. Access is automated

Snowden discoveries:

- PRISM
- XKeyscore
- Tempora
- ECHELON

Mandatory metadata retention
- incoming/outcoming telephone caller id
- date, time, duration of call
- location of the device which call was made
- unique identifier for each phone
- email address

"It's only metadata"

Access and Assistance bill (2018)
https://www.homeaffairs.gov.au/about-us/our-portfolios/national-security/lawful-access-telecommunications/myths-assistance-access-act
---

Is it justified?
How do you fight it?
What about meta information?

Adam's opinion:

- **Start with engineers**: Understand the ethics around it. Take a stand if necessary. Broaden our perspectives
- **Advocacy**: Educate other people. Complain to governments and organisations, it's our duty as technical people in the room
- **Design**: It's not all or nothing. Engineer a genuine choice. Engineer safer ways eg. on device. Minimise data along the way
- **Story Telling**: Allow people to fully understand the impact eg. Blackmirror, 1984

Rootkits seminar

root: root, or admin; highest possible level of access privilege
kit: software that grants root-level access to the system

a rootkit can:
- conceal itself
- execute any process
- make changes to the system
- track usage of the system

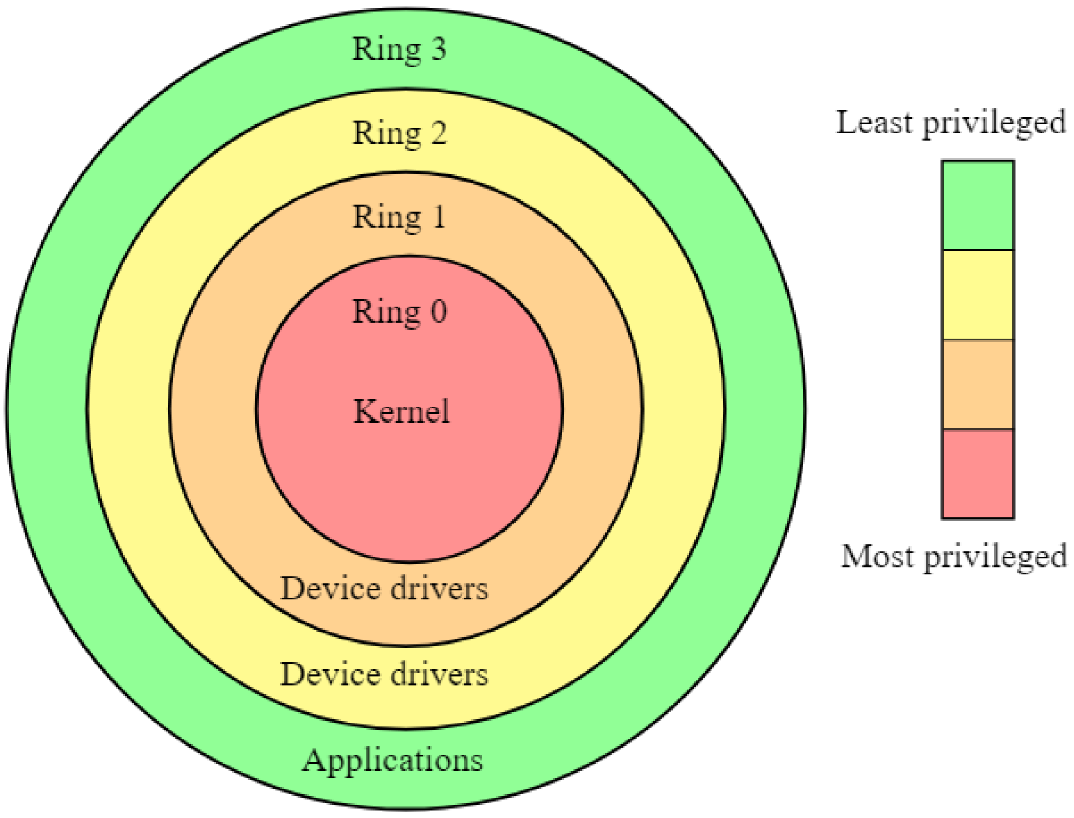
not malicious by itself
can enable malware
often bundled with malware
Zeus aka ZeuS aka Zbot - trojan
uses rootkit to hide keylogger

installation:
- phishing attacks
- social engineering
- inserting usb into system
- someone willingly granting access to system

history:
- earliest rootkit - admin tools that replaced legitimate tools on UNIX
pretty easy to detect
- 1980 - Ken Thompson
- 1986 - 'brain virus', not really a rootkit uses cloaking (stealth virus)
- 1990 - first real rootkit
- 1999 - first malicious rootkit for windows OS - NTRootkit
- 2005 - sony BMG modifies operating system to tamper with disc copying
- 2009 - first rootkit for Mac OS X
- 2009 - Zeus infects 3.6 million devices in the US

concept of rootkits is actually quite old
gotten a lot more sophisticated as time went on

Privilege ring (not completely accurate, more inside kernel)

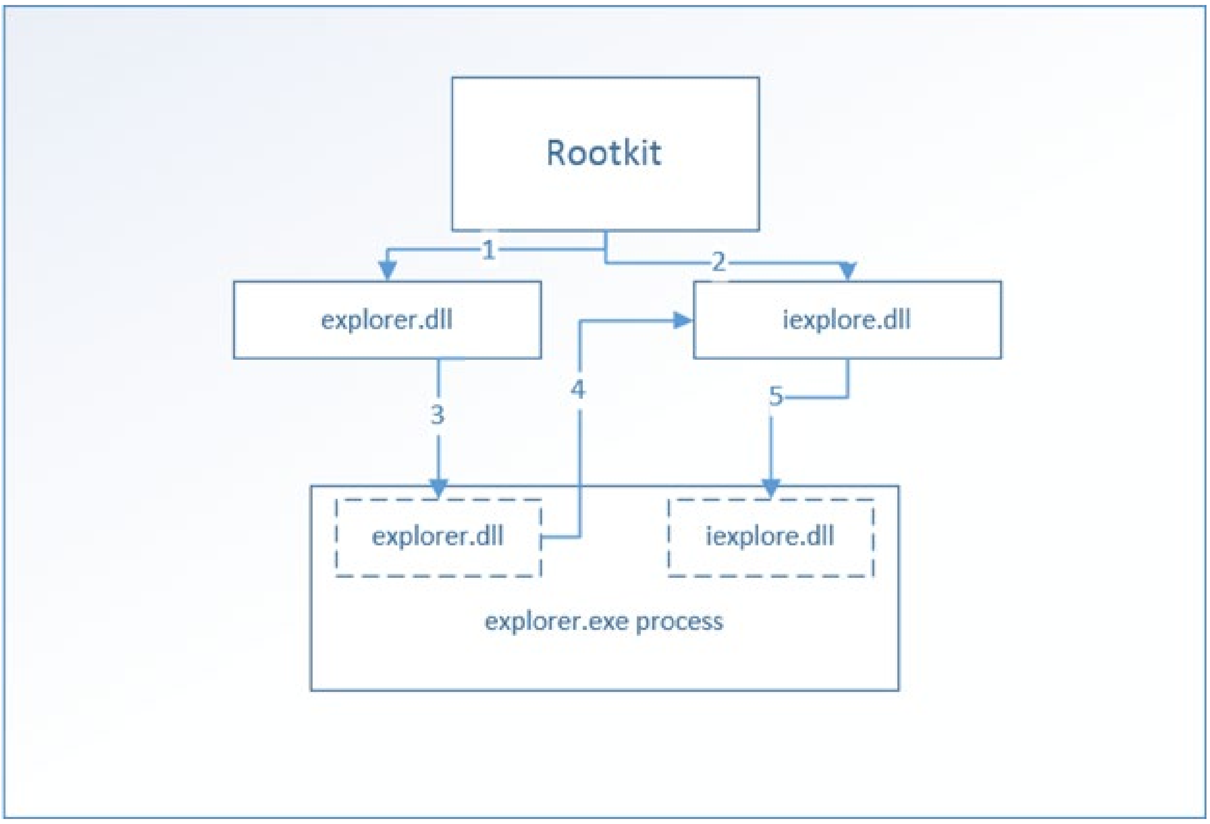


Types of rootkits:
- usermode

modifies user-level apps or shared libraries
remote access: backdoor sshd
privilege escalation: backdoor su
hiding:

  - process - replace ps, pidof, top
  - network - replace netstat, ifconfig
  - file hiding - replace ls, find

DLL injection
injecting into explorer.exe process



- kernel
violates trust of users and processes
alters objects stored in kernel memory
alter results of system calls eg. list files in dir
pro - hard to detect by traditional anti-malware software
con - harder to write

- system call hooking (BSD):
one of the most popular techniques
sysent - system call table of syscalls
have index point to your hook
syscall j now points to your hook
so it'll run your program

- memory based:
never write to disk
    - no physical presence
    - hard to detect
no persistence, sometimes that's ok
good for one-off, specific, targeted attacks
SucKit (2001)
    - pattern searches /dev/kmem and located syscall table
    - patches /dev/kmem by overwriting entries in the syscall table
    - undetectable by LKM detection methods

- bootkits:
replaces the legitimate boot loader with a boot loader under the attacker's control
usually performs the transition to 'protected mode' thus intercepting encryption and passwords
harder to detect
removing a Bootkit may 'brick' your system
can attack full disk encryption systems

- hypervisor level rootkits
exploit hardware virtualisation features such as Intel VT and AMD-V
runs at ring -1 (before the kernel) and hosts the OS as a virtual machine
intercepts hardware calls made by the oS
does not have to modify kernel, thus harder to detect than regular Bootkits
only detectable through extremely low level monitoring such as measuring hardware latencies

detection:
- boot into a different OS and check the contents of the drive
- RKHunter - hashes files and compares them with known good hash
- catch the OS lying
eg. checking netstat, nmap
- cat and mouse game, defence is always playing catch up

a lot of these tools (RKHunter, Tripwire, antivirus) can be evaded though

further reading material:
Designing BSD Rootkits
Rootkits & Bootkits
Rootkits Arsenal
http://www.phrack.org/
Reflections on trusting trust

modify your own kernel (in a virtual machine)!

“That’s some hickity hackity stuff, damn” - Lachlan

“I’m really proud of you and your spirit and the work you have put into the course and how you have all worked together. India is lovely but I miss you all and wish I was there to share the final week with you. Please send me a huge class selfie if you can take one Lachlan! I’ll be running a revision session in the week before the exam - will sort out a room and time when I get back and post it on open learning.” - Richard