λ
λ λ

DAT235 - Cheat Sheet

Basic Security (25%) **4**

The Basics of Information Security 4

What is Information Security? 4

CIA Triad 4

Risk Management 4

Incident response 5

Defense in Depth 5

Identification and Authentication 6

Identification 6

Examples 6

Authentication 6

One Factor, Two Factor, Three Factor 6

Mutual Authentication 6

GSM Authentication and Key Agreement 7

Authentication: SSL/TLS 7

Authentication: Passwords 7

Authorization and Access Control 8

Principle of least privilege 8

Access Control 9

Confused Deputy 9

Cross-site request forgery (confused deputy: the browser) 10

Access Control – Bell-LaPadula 10

Access Control – Biba Model 10

Access Control – Clark-Wilson Model 10

Access Control – Brewer and Nash model 11

Access Control Methodologies 11

Cryptography 11

Caesar Cipher (ROT13) 11

Vigenère Cipher 12

λ
λ λ
One Time Pad 12
Modern Cryptography 13
Symmetric vs asymmetric cryptography 13
Laws and Regulations 13
Many important infosec standards 13
GDPR - The EU General Data Protection Regulation 13
Operations Security & Human Element Security 14
Operations security process 14
The weakest link 14
Norwegian cyber security culture 14
Security awareness 16
Social engineering 16
Physical Security 17
Physical Security Controls 17
Protecting Data 17
Network Security 18
Operating System Security 18
Operating System Hardening 18
Application Security 18
Secure design (incl. threat modeling) 18
Software Development - main vulnerability categories 18
Common Vulnerabilities and Weaknesses 19
Microsoft SDL Cryptographic Recommendations 19
MAC/HMAC/keyed hash algorithms 19
Privacy by Design: The 7 Foundational Principles 19
The 7 Foundational Principles 19
Threat Modeling (25%) **21**
STRIDE Threat Modeling cheat sheet 21
What is a model? 21
STRIDE 21

How to «threat model» 22
1. What are you building – Define the model (architecture) 22
2. What can go wrong? 22
3. What are you going to do about it 23
4. Verify and validate 24
MQTT, Project, Lab(s), Assignment(s), etc (25%) 25
MQTT CHEATSHEET 25
MQTT client(=publisher, subscriber) 25
MQTT server(=broker) 25
Topic 25
MQTT Control Packet Structure 25
Basic details and Terminology 25
MQTT Control Packet Structure: Fixed Header 25
Data Representations 26
MQTT Control Packet Structure: Variable Length 26
Quality of Service 26
Topic Names 26
Wildcards 26
MQTT Control Packets 28
CONNECT 28
CONNACK 28
CONNACK: Error Codes 29
PUBLISH 29
Protocol Exchange: QoS 1 29
Protocol Exchange: QoS 2 29
Security 30
MQTT-S (Extension of MQTT for WSNs) 30
LABS 30
TEST EXAM 2017 DAT235 31
The Basics of Information Security (25%) 31

λ
λ λ

Threat Modeling (25%) 33
IoT Security (25%) 34
MQTT, Project, Lab, Assignments.. (25%) 36

Basic Security (25%)

The Basics of Information Security

What is Information Security?

Information security (InfoSec) is a set of strategies for managing the process, tools and policies necessary to prevent, detect, document and counter threats to digital and non-digital information.

CIA Triad

Confidentiality, **I**ntegrity and **A**vailability also known as the **CIA** triad. Is a model designed to guide policies for information security within an organization The model is also sometimes referred to as the **AIC** triad (**A**vailability, **I**ntegrity and **C**onfidentiality) to avoid confusion with the *Central Intelligence Agency*. The elements of the triad are considered the three most crucial components of security.

- **Confidentiality**
  - "Keeping data secret"
  - Restricting read access
- **Integrity**
  - "Avoiding unauthorized modification"
  - Restriction write access
- **Availability** (sometimes Authentication)
  - "Preventing denial-of-service"
  - Protecting possession

Risk Management

There will always be a risk – so better be aware and handle it.

- Identify Assets
  - What is valuable to you?
  - What may be valuable to others?
- Identify Threats
  - What/who may threaten our assets?
  - What types of threats are there (against CIA, ...)?
  - Try to consider if the threats are relevant
- Assess vulnerabilities
  - Could the threats be converted to attacks?
  - Are there any weakness/Flaws etc. that may make an attack a feasible?
- Assess Risk
  - Less likelihood ⟶ Less risk
  - Less impact ⟶ less risk
- Mitigate / Remove risk

λ
λ λ

- If guessing passwords is too easy, introduce better password rules
- Add two-factor authentication (something you know/have/are)

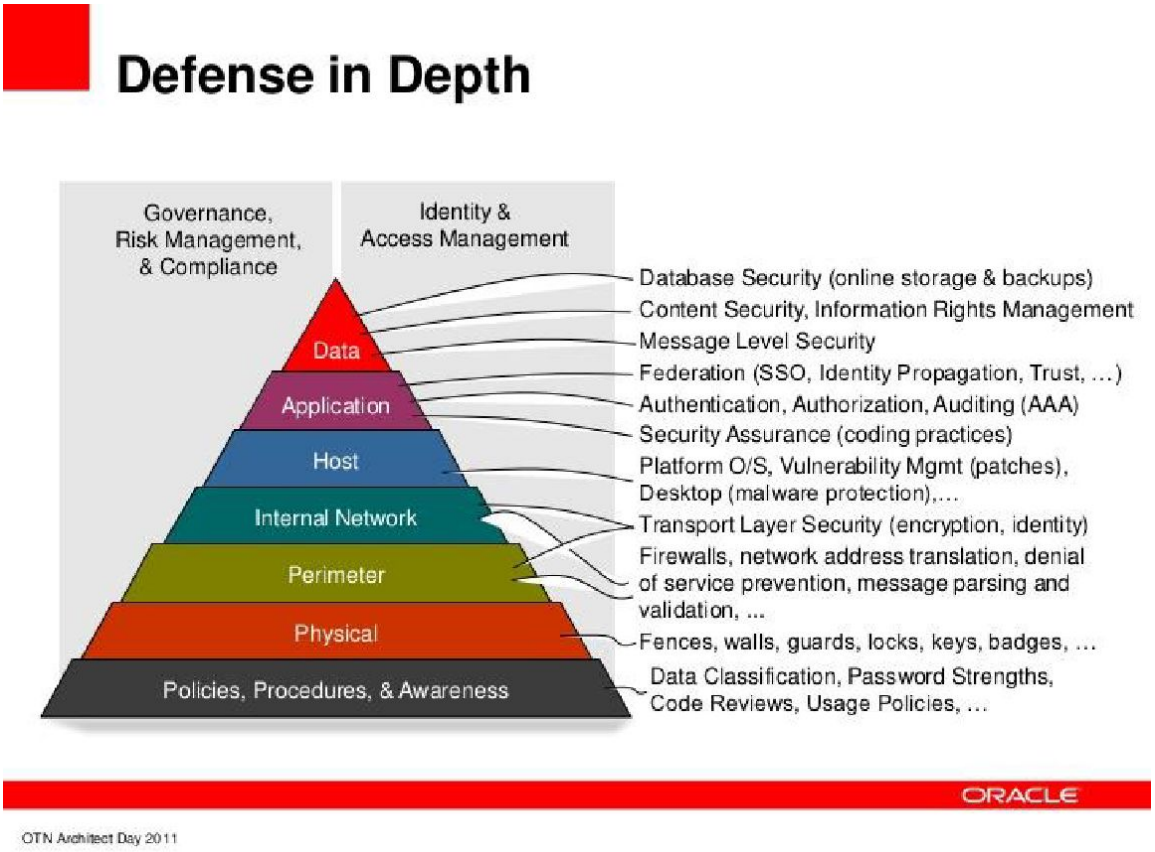
- You may also accept the risk
  - The risk may be sufficiently low that you simply take the risk
  - If the gamble goes wrong, you'll need to handle the consequences...

Incident response

- Containments, eradication and recovery
  - Stop it from getting worse
  - Remove the effects of the attack
- Post incident analysis
  - Why did it go wrong?
    - Technical reasons?
    - Human factors?
    - Bad luck?

Defense in Depth

- Never rely on a single Mechanism → Single point of failure
- Many defense mechanisms – Even overlapping



λ
λ λ

Identification and Authentication

Identification

- Need to identify the entities in the system.
- Entity? - “A thing with distinct and independents existence.”
    - Persons or process to us.
- Identifiers
    - Numbers
    - Names
    - References
    - Addresses
    - Unique, shared, group anonymous, alias

Examples

- IMSI – International Mobile Subscriber Identity
- MSISDN – Mobile Station ISDN Number
- CGI – Cell Global Identification
- Social Security Number
    - Including gender information

Authentication

One Factor, Two Factor, Three Factor

- What you know
    - Passphrase
        - Password
        - Pin code
- What you have
    - Security Dongle
    - ID card
    - Mobile ID
- What you are
    - Biometrics

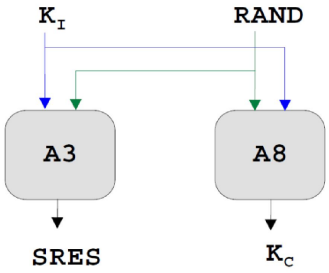
Mutual Authentication

- Security Protocols – Remote Authentication
    - When you cannot *see* the other party
- Alice and Bob
    - If only Alice can verify that she is in contact with Bob
        - One-Way Authentication
    - If Alice and Bob can verify each other
        - Mutual Authentication
    - Eve (or Mallory)
        - The Intruder – Wants to cheat Alice and Bob

λ
λ λ

GSM Authentication and Key Agreement

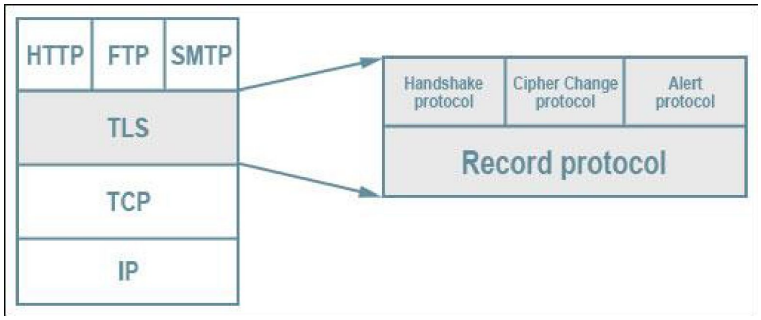
- Why Authenticate an Entity?
  - Usually, because one wants to communicate more so it makes sense to also agree on session keys. These keys are used to protect communications. May also agree on key deriving keys
- GSM Authentication and key Agreements (AKA) Algorithm
  - A3 and A8 (also known as A38)
    - Ki – 128 bit wide
    - RAND – 128 bit wide
    - SRES – 32 bit wide
    - Kc - 64 bit wide



Authentication: SSL/TLS

- **SSL (Secure Sockets Layer)**
  - Is a layer 4 security technology for establishing an encrypted link between a server and a client (web server and a browser)
  - **SSL** has been replaced by **TLS**, but the name **SSL** is still often used.

- **TLS (Transport Layer Security)**
  - Is a layer 4 security technology for establishing an encrypted link between a server and a client (web server and a browser)



Authentication: Passwords

- Passwords should be secrets
  - Do not share them
  - Make sure they're not easy to guess
- How to ensure that you have secure passwords
  - My passwords are long, complex and don't contain dictionary words
  - Never share my passwords
  - Use different passwords for different services
  - Use different passwords for home and school / work
  - Store my passwords securely
  - Use multifactor authentication, like Google's 2-step verification

λ
λ λ

Authorization and Access Control

Authorization Indicates what one is permitted to do.

- Who can read your grades
- Who can set your grades

Access control is the process to ensure that all actions are authorized

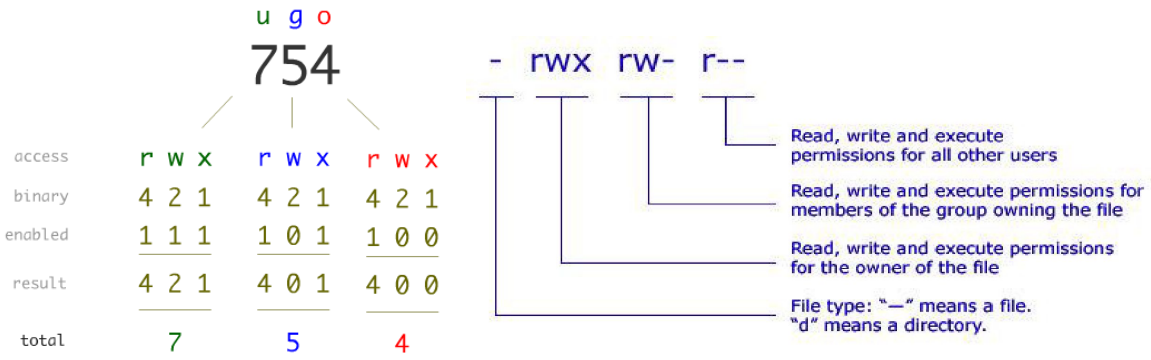
Principle of least privilege

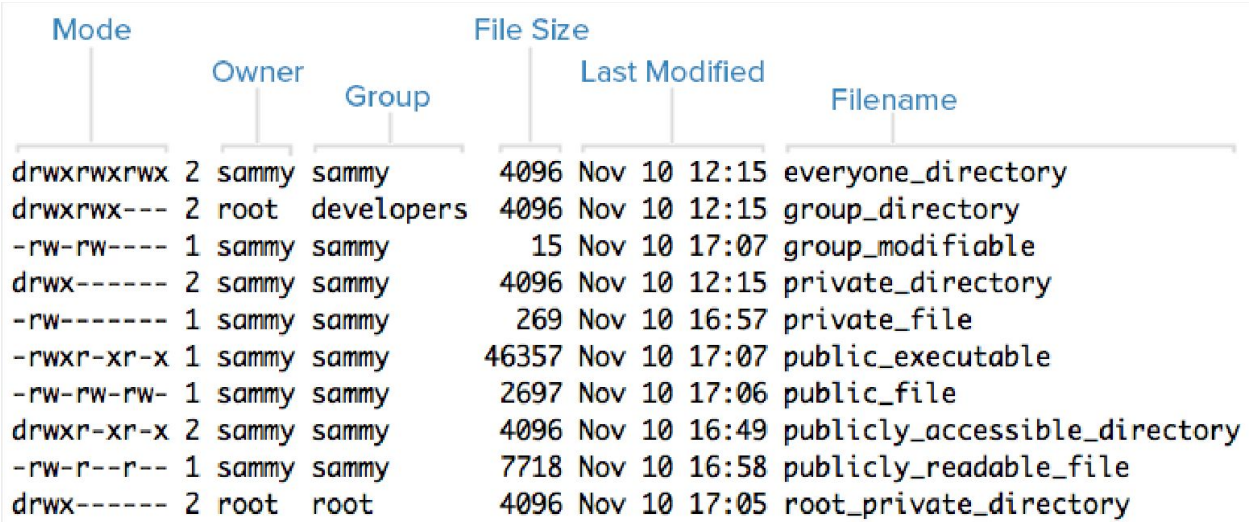
- Strategy
  - Limit the permissions to the minimum
- Benefit
  - It will be hard to abuse the permissions
  - This is true for the authenticated entity (Which may be dishonest!)
  - It is also true if someone gains illicit success to the account.
- Disadvantage
  - Not so many from a security point of view
  - But, usability may suffer (making it harder for the legitimate user)
  - Flexibility also suffers
    - At worst: It limits the availability (!)
- In Real Life
  - Distinguish between administrator and everyday use
  - Administration may also be split
    - Security and rights setup for the users (fix locked accounts, etc).
    - Application/service Administration
  - In large systems, there may be many types of users
    - Lectures that teach
      - Many different courses and different studies
      - Many different lectures
    - Students
      - Many different students and many different courses
    - Administration People
      - But they should not be given universal rights.

λ
λ λ

Access Control

- How to enforce authorization and least privilege?
  - **Access Control List (ACL)**
    - Every object has a list of who may access the object
    - This includes info on the permission they have (read, write , delete,..)
    - Common in file systems



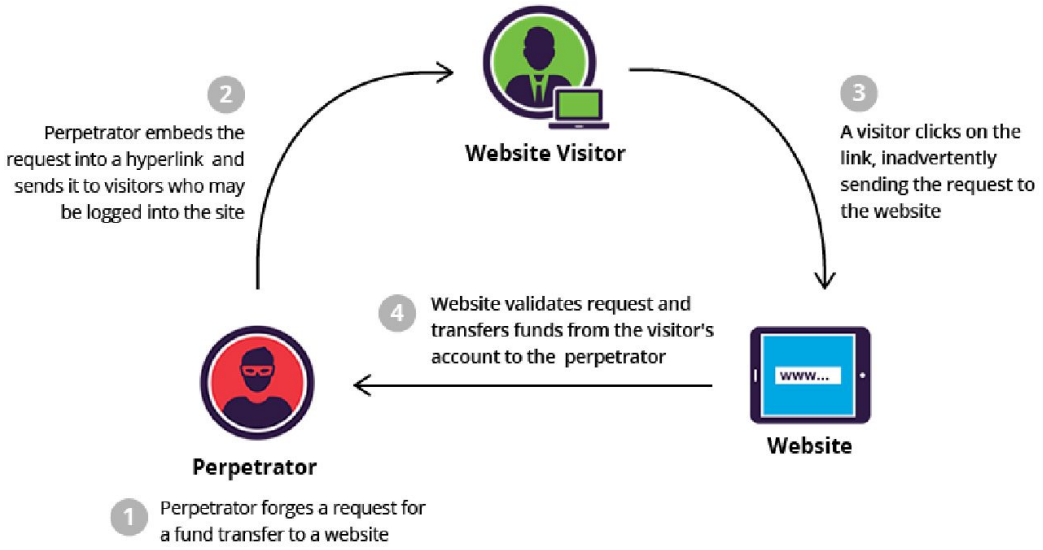


Confused Deputy

- Ingredients
  - A computer program (web-browser, document system, ..=
  - The Program has many access rights
  - You have fewer rights
  - Program should limits its action to what you can do.
- This is a **privilege escalation** problem

λ
λ λ

Cross-site request forgery (confused deputy: the browser)



Access Control – Bell-LaPadula

- It's all about confidentiality
- In short
  - Classification levels (top secret, secret, confidential, restricted, unclassified)
  - People are then authorized to a clearance level
  - For the objects you have access to
    - You can read up to (and including) your own level
    - You can write to all levels:
      - Also higher than your clearance, but then you cannot read it later!
- Bell-LaPadula Model: "read down, write up"

Access Control – Biba Model

- It's all about integrity
- Data Integrity goals:
  - Prevent data modification by unauthorized parties
  - Prevent unauthorized data modification by authorized parties
  - Maintain internal and external consistency (general rules of models)
    - The model must match reality
- Biba Model: "Read up, write down"
  - No integrity harm from reading at all levels
  - To ensure integrity: only be able to **write down** (at permitted levels)

Access Control – Clark-Wilson Model

- Recognizes limitations and problems with Biba and Bell-LaPadula
- It's mostly about integrity

λ
λ λ

Access Control – Brewer and Nash model

- It's an information flow model
- Mitigates conflict of interest
- Also known as the Chinese Wall security model

Access Control Methodologies

- DAC – Discretionary Access Control
  - Owner of a resource decides who gets access and what they are permitted to do
  - Ok when dealing with few objects, but it quickly gets complicated and inconsistent
- MAC – Mandatory Access Control
  - The organization (admin) decides who has access and rights
  - Classified access (top secret, secret, confidential, restricted, ...)
  - Each person must have a security clearance level
  - You get access to your level (not above)
  - Explicit authorization to each resource or resource group
    - Need-to-know (least privilege principle)
- RBAC – Role Based Access Control
  - People have roles – patient, doctor, nurse, accountant, etc.
  - The rights you have are associated with the role you have
  - You may have more than one role (but not at the same time)
- ABAC – Attribute Based Access Control
  - • More fine grained decision logic for access, based on attributes
  - • More context aware than RBAC
  - • Based one policies

Cryptography

Encryption is the process of encoding a message or information in such way that only authorized parties can access it.

Caesar Cipher (ROT13)

The Caesar Cipher, also known as a shift cipher, is one of the simplest forms of encryption. It is a substitution cipher where each letter in the original message(called plaintext) is replaced with a letter corresponding to a certain number or letters up or down in the alphabet.

For example, here's the Caesar Cipher encryption of a message, using a right shift of 3.

```
Plaintext:
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext:
QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
```

As unreadable as the resulting ciphertext my appear, the Caesar Cipher is one of the weakest forms of encryption one can employ.

- The key space is very small. Using a brute force method, one could easily try all (25) possible combinations in order to decrypt the message without initially knowing the key.
- The structure of the original plaintext remains intact. This makes the encryption method vulnerable to frequency analysis – by looking at how often certain characters or sequences of

λ
λ λ

characters appear, one can discover patterns and potentially discover the key without having to perform a full brute force search.

The Caesar Cipher can be expressed in a more mathematical form as follow:

$E_n(x) = (x + n) \bmod 26$

Vigenère Cipher

The Vigenère Cipher was developed by a mathematician Blaise de Vigenere in the 16th century. The Vigenère Cipher was adapted as a twist on the standard Caesar cipher to reduce the effectiveness of performing frequency analysis on the ciphertext. The cipher accomplishes this using uses a text string (for example, a word) as a key, which is then used for doing a number of alphabet shifts on the plaintext. Similar to the Caesar Cipher, but instead of performing a single alphabet shifts across the entire plaintext, Vigenère Cipher uses a key to determine several different shifts amounts across the eternity of the message.

Should the key be shorter than the plaintext, it is repeated until the length matches. In this way, each letter in the plaintext is shifted by the alphabet number of the corresponding letter in the key.

```
Plaintext:  ATTACKATDAWN
Key:        LEMONLEMONLE
Ciphertext: LXFOPVEFRNHR
```

Expressed mathematically, the encryption of the message at letter I, is equal to the alphabetic value of I in the plaintext plus the alphabetic value of the corresponding I in the key.

$E_k(M_i) = (M_i + K_i) \bmod 26$

Decryption is the same process reversed, subtracting the key instead of adding to arrive back at the original, plaintext value.

$D_k(C_i) = (C_i + K_i) \bmod 26$

One Time Pad

Towards the end of the 19th century, it was becoming fairly obvious that simple substitution ciphers that were vulnerable to frequency analysis were no longer secure. How could they transmit a message without its contents being vulnerable to inspection, while still being to be decrypted once they reached their destination? The answer if randomness.

Let's consider a modified version of the Vigenère Cipher as an example. Instead of performing alphabet shifts on each character in the plaintext via a key, we shift every character in the plaintext a random amount – with each of these random shift amounts becoming the resulting key. This results in a completely random key the exact same length as the original plaintext. This means that across the entire message, there can be no structure deduced from the frequency of characters, as the key itself was entirely randomly created. This type is called the one-time pad, and the benefits don't stop there.

λ
λ λ

With each character now having its own individual and random shift amount, the key-space grows exponentially for each character in the message. Let's say we were to encrypt the name "Alice" with one time pad. That's 5 letters – so to brute force it you would have to try a whole lot of possibilities:

(26 x 26 x 26 x 26 x 26) or 26^5 = 11881376

However, this would simply brute force the search space. You still wouldn't know which of the millions of attempts you tried were correct, because (due to randomness of the one-time pad) it's possible that attempting to decrypt the message with an incorrect key, could potentially give a coherent but incorrect result. Because of this, while being a relatively simple encryption technique, the one time pad is considering unbreakable if used properly.

Modern Cryptography

- Kerckhoff's Principle - the system (algorithms) must not be secrets
- Basically means that one **cannot** rely on secrecy of algorithm
- Secrecy **must** be by means of **secret key**

The actual principles (no.2 is the important):

1. The system should be, if not theoretically unbreakable, unbreakable in practice.
2. The design of a system should not require secrecy, and compromise of the system should not inconvenience the correspondents (Kerckhoffs' principle).
3. The key should be memorable without notes and should be easily changeable
4. The cryptograms should be transmittable by telegraph.
5. The apparatus or documents should be portable and operable by a single person
6. The system should be easy, neither requiring knowledge of a long list of rules nor involving mental strain

Symmetric vs asymmetric cryptography

- **Symmetric - same (secret) key for encryption and decryption**
- **Asymmetric - two different keys**
    - A private key for decryption or signing (secret)
    - A public key for encryption or verifying a signature (not secret)

Laws and Regulations

Many important infosec standards

- 3GPP (mobile networks: GSM/EDGE, UMTS, LTE, LTE-A, 5G)
    - Several important security standards amongst them
    - Mostly in the 33.series (but not exclusive there)

GDPR - The EU General Data Protection Regulation

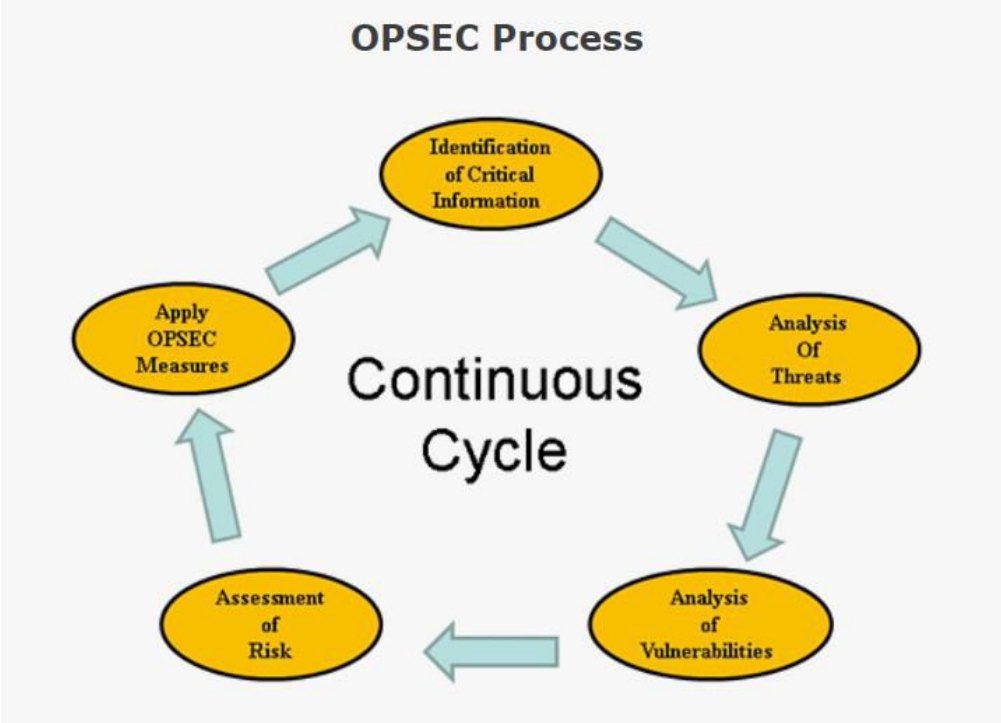
- IoT handles a lot of privacy sensitive data
- You future job can very well include knowledge of GDPR

λ
λ λ

Operations Security & Human Element Security

Operations security process

1. Identification of critical information (assets)
2. Analysis of threats
3. Analysis of vulnerabilities
4. Assessment of risks
5. Application of countermeasures



The weakest link

- kek...HOOMANS

Norwegian cyber security culture

Core issues in cyber security culture

1. Collectivism
    - Cultures are per definition collective.
    - Cultures are developed by individuals, whilst at the same time contribute to shaping the individuals that are part of any given culture.
    - Cultures point to the characteristics of a particular group of people, including such as their social habits, their attitudes, their values and priorities.
    - Cultures necessitate some degree of solidarity amongst the members.
    - The individuals must identify themselves as parpart of the group, contribute to it, and adhere to the explicit and implicit norms of behaviour.

λ
λ λ

2. Governance and Control
    - Governance is a collective term that refers to the questions of how the collective should be regulated and by whom.
    - Governance refers to the users' views on governance and control of information and communications technology (ICT).
    - A critical issue is the question of surveillance: Who are responsible for drawing the red lines of what is acceptable in the use of ICT, where should these lines be drawn and how should citizens abide to these lines?
3. Trust
    - Trust is a cornerstone to any viable democracy.
    - A well-functioning democracy necessitates trust amongst its citizens, amongst citizens and the government, between governmental institutions, between business, between citizens and their employer and so forth.
    - Trust is a prerequisite for economic welfare, stability and growth in a country.
    - As more and more of our national growth is tied to the digitalization of the nation, trust in this area is of great significance.
4. Risk perception
    - Competence, learning and risk are tightly knit together.
    - Studies have shown an increase in "risky behaviour" amongst individuals who have a high level of competence or perceived skill
    - It is likely that people who have skill in the area of cyber security could overestimate their ability to control the threat, and they may therefore take more risks
5. Techno-optimism and digitalization
    - Digitalization is part of how our societies develop.
    - Citizens' attitude towards this societal tendency.
        - Your attitude towards digitalization influences how you relate to technology.
        - A safe e-citizen is fundamental to the success of the national digitalization.
6. Competence
    - Everything from social services and state tax payment to individual communication are happening online
    - Citizens are forced to make use of ICT, regardless of whether they appreciate it or nor.
    - All citizens of Norway must therefore have fundamental digital skills.
7. Interest
    - Do citizens with an interest in ICT have an advantage over those citizens that lack this interest?
    - Interest shapes our attitudes, our skills and our knowledge.
    - Interest influences who we relate to and thereby who we learn from.
    - With interest comes awareness, curiosity and time.
    - These are cornerstone in learning.

λ
λ λ

8. Behaviour
    - Most studies of cyber security culture focuses on behaviour.
    - It is also what we do that most concretely influence our cyber security
    - In terms of cyber security there are certain types of behaviour that are encouraged, whilst others are warned against.
    - Governments, authorities, business leaders and experts provide advice that form a normative standard for how citizens or employees should strive towards behaving

Security awareness

- Knowledge is a key component
- But the right attitude is even more important
- Knowledge about appropriate regulations and laws
- Knowledge about best practices and important standards
- Passwords and other credentials

Social engineering

- Pretexting – a lie that makes the target assume things
    - Affects identification, authentication and authorization
    - Often relies on assumed associations
    - Often relies on your weaknesses (what you want to happened, want to hear)
    - Relies on misplaced trust in institutions, brands, etc.
- Phishing – tricking someone into giving away sensitive information
    - False web pages (banks, etc.)
    - False emails asking for info
- Spear phishing
    - Targeted phishing
- Water holing
    - Where do your targets go to relax (pub, football match, clubs)
    - Gain confidence by familiarity
    - Gain trust by gives and similarity, etc.
    - Also: where on the web do your targets go?
    - Attack that web page specifically
- Tailgating / piggybacking
    - A person tags along with another person who is authorized to gain entry into a restricted area, or pass a checkpoint.

λ
λ λ

Physical Security

- Physical security controls
- Protecting data
- Major categories of physical threats
    - Extreme temperature
    - Gases
    - Liquids
    - Living organisms
    - Projectiles
    - Movement
    - Energy anomalies
    - People
    - Toxins
    - Smoke and fire

Physical Security Controls

- Deterrent
    - To discourage physical attacks
- Detection
    - To discover attacks (burglar alarm, surveillance)
- Preventive
    - Physically hinder an attack

Protecting Data

- Aging and secure destruction
    - Formats come and go
    - Lifecycle (residual data)
        - Data must also be destroyed
        - Old storage is obsoleted
        - Not always easy to destroy data
- Availability
    - Ensuring that we have access to data (by physical means)
    - Capacity / redundancy
- Backups
    - Selected backups
    - Full backups
    - Incremental backups
- Geographical distribution
    - Floods & fires
    - Cloud
- Recovery
    - It must be easy to use the backups when needed
- Redundancy
    - Geo-redundancy, capacity redundancy, etc
    - RAID disk storage is an example (to provide reliability)

λ
λ λ

Network Security

- DETTE BØR DU KUNNE PÅ STRAK ARM...

Operating System Security

Operating System Hardening

1. Remove unnecessary software
    - Reduce the attack surface
2. Remove unneeded services
    - Telnet, smtp, ..osv
3. Alter default accounts
    - system can have many default accounts for different purposes
        - Must assume that these are well-known to attackers
        - Disable and remove those you don't need
4. Apply "Least privilege"
5. Apply updates (user newest versions whenever practical)
6. Implement logging and auditing

Application Security

1. Secure design (incl. threat modeling)

- Security Development Lifecycle
    - Training
    - Requirements
    - Design
    - Implementation
    - Verification
    - Release
    - Response

2. Software Development - main vulnerability categories

- Buffer overflows
    - Memory access problem
- Race conditions
    - Transaction that happen more-or-less simultaneously
    - Asynchronous in nature (missing sync is part of the problem)
- Input validation attacks
    - Checks if the input is meets a set of criteria. Typical checks include checking if string contains unexpected quotation marks, code, special characters, escape characters, etc
    - Never assume anything - check and verify
- Authentication attacks
    - Many ways to carry out impersonation/masquerade
    - Authentication schemes may be fooled!
    - Web
        - Server side
        - Client side - don't trust the client

- Authorization attacks
    - Confused deputy problem
    - lack of policy
- Cryptographic attacks
    - Outdated libaries & API's
    - Wrong library & API usage

3. Common Vulnerabilities and Weaknesses

- CVSS
    - Common Vulnerability Scoring System

Microsoft SDL Cryptographic Recommendations

MAC/HMAC/keyed hash algorithms

A message authentication code (MAC) is a piece of information attached to a message that allows its recipient to verify both the authenticity of the sender and the integrity of the message using a secret key.
The use of either a hash-based MAC (HMAC) or block-cipher-based MAC is recommended as long as all underlying hash or symmetric encryption algorithms are also recommended for use; currently this includes the HMAC-SHA2 functions (HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512).
Truncation of HMACs to less than 128 bits is not recommended.

Privacy by Design: The 7 Foundational Principles

The 7 Foundational Principles

1. Proactive not Reactive; Preventative not Remedial

The Privacy by Design (PbD) approach is characterized by proactive rather than reactive measures. It anticipates and prevents privacy invasive events before they happen. PbD does not wait for privacy risks to materialize, nor does it offer remedies for resolving privacy infractions once they have occurred — it aims to prevent them from occurring. In short, Privacy by Design comes before-the-fact, not after.

2. Privacy as the Default Setting

We can all be certain of one thing — the default rules! Privacy by Design seeks to deliver the maximum degree of privacy by ensuring that personal data are automatically protected in any given IT system or business practice. If an individual does nothing, their privacy still remains intact. No action is required on the part of the individual to protect their privacy — it is built into the system, by default.

3. Privacy Embedded into Design

Privacy by Design is embedded into the design and architecture of IT systems and business practices. It is not bolted on as an add-on, after the fact. The result is that privacy becomes an essential component of the core functionality being delivered. Privacy is integral to the system, without diminishing functionality.

4. Full Functionality — Positive-Sum, not Zero-Sum

Privacy by Design seeks to accommodate all legitimate interests and objectives in a positive-sum "win-win" manner, not through a dated, zero-sum approach, where unnecessary trade-offs are made.

λ
λ λ

Privacy by Design avoids the pretense of false dichotomies, such as privacy vs. security, demonstrating that it is possible to have both.

5. End-to-End Security — Full Lifecycle Protection
Privacy by Design, having been embedded into the system prior to the first element of information being collected, extends securely throughout the entire lifecycle of the data involved — strong security measures are essential to privacy, from start to finish. This ensures that all data are securely retained, and then securely destroyed at the end of the process, in a timely fashion. Thus, Privacy by Design ensures cradle to grave, secure lifecycle management of information, end-to-end.

6. Visibility and Transparency — Keep it Open
Privacy by Design seeks to assure all stakeholders that whatever the business practice or technology involved, it is in fact, operating according to the stated promises and objectives, subject to independent verification. Its component parts and operations remain visible and transparent, to users and providers alike. Remember, trust but verify.

7. Respect for User Privacy — Keep it User-Centric
Above all, Privacy by Design requires architects and operators to keep the interests of the individual uppermost by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options. Keep it user-centric.

λ
λ λ

Threat Modeling (25%)

STRIDE Threat Modeling cheat sheet

What is a model?

Before we begin: What is a model?

- Model is a simplified and abstract version of the modeled item
- Should be kept simple when possible (remove unimportant details)
- But, the essential elements must be there (this depends on the purpose of the model)
- Too simple: then the model cannot capture important aspect of reality (validity problem)

Why model?

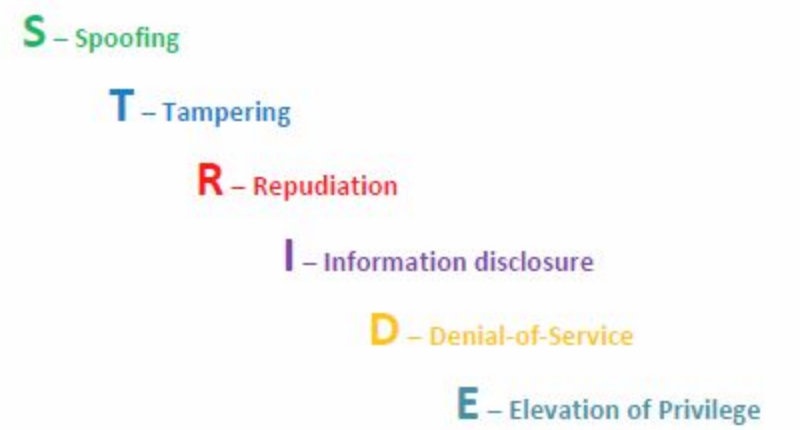
"All models are wrong, but some are useful" – George Box (he almost said this)
We certainly want to find flaws and fix as much as we can. But, fixes are not really the most important part here. To fix a flaw is good, but to improve the design is way better. Perhaps the most important part of a model is to learn more about the target (corresponding) system. This allows a deeper understanding and should lead to enhanced design and improved implementation.

Validity and verification:

- Validity: Does the model correspond to the real world (whatever that is)
- Verifying a model: Checking the model itself

STRIDE

Stride - verb: *Walk with long, decisive steps in a specified direction.*



λ
λ λ

How to «threat model»

1. What are you building?
2. What can go wrong?
3. What are you going to do about it?
4. Check your model and your measures

1. What are you building – Define the model (architecture)

- **The essential parts of the model:**
    a. Interfaces and trust boundaries (trust zones)
    b. Define the actors (processes, people)
    c. Define the data storages and data flows
- **Define the assets**
    a. Something an attacker wants
    b. Something you want to protect
    c. A stepping stone towards a) and/or b)

2. What can go wrong?

- **A lot of things can go wrong**

Brainstorming can be fun, but we need structure. And, we don't want too many "silly" wrongs (totally out-of-scope threats).

- Focus on active (initiator) entities
- Focus on assets
- Focus on:
    a. Interfaces and trust boundaries (trust zones)
    b. actors (processes, people)
    c. data storages and data flows
- Threats according to **STRIDE**
- Breath vs Depth
    a. Usually best to start with breadth first
    b. Depth to follow later (if necessary, time permitting)

λ
λ λ

3. What are you going to do about it

Need to keep track of all the threats. Index them according to threat (STRIDE), entity/actor and interface.

So what do you do with the threats?

- Know Thy Enemy – what kind of intruder should you defend against?
- APT actors are very different from an opportunistic hacker

High-level vs. low-level

- Try to think high-level: Are there any design change that can get rid of a whole class of threats?
- Low-level: Find some specific security control or other means to address each individual threat.

Think long-term when you can:

- What will pay off in the long-term?
- Quick fixes may be costly to maintain (they add operational complexity too)

Some strategies for dealing with threats:

- Remove the threat (remove functionality)
- Mitigate the threat
    - Reducing exposure when possible
    - Technical provision to prevent attacks (security controls etc)
    - May include technical provision to lessen the impact of an attack
    - May include technical provision that makes an attack harder to carry out
- Accept the risk
    - There will always be risks!
    - If you understand the risk, you decide that it is acceptable to take the chance
- Transfer the risk
    - Insurance
    - Some other party

Proactive vs reactive:

- Pro-active: Preventing incidents from happening
- Re-active: “Detection & Response” capabilities

You are going to need both approaches. There will inevitably be incidents, and then you really need “detection & response” capabilities. Of course, to prevent incidents from occurring is good. But, it may cost too much (money or usability, or whatever). And, whatever you do, you cannot prevent everything.

λ
λ λ

4. Verify and validate

Map vs terrain

- Does the model actually correspond to the system?
    - Need to update the model when the system design is updated
    - Also: the system architecture and the system implementation may deviate
    - Ideally: We model the architecture, but we may need to adjust for the implementation
- Is the model accurate enough?
- Is it consistent?

Especially:

- Are the trusty boundary assumptions really accurate?
    - If you have 20.000 employees, then you really don't want to have them all in the same group. Then the trust boundaries needs to be refined (split up).
- Are actor model (users, processes) really accurate?
- What about the assumed intruder and your assets?
    - If you assume a powerful intruder, then your protection should be strong
    - Do you assume that the intruder has a strong motivation?
    - Are the assets valuable to the would-be intruder?
    - How valuable are the assets to you?
        - Note that the value of the assets may be subjective
        - Note that the value may change over time

Have all threats been addressed?

- Are all threats addressed?
- Are the threats addressed in a satisfactory way?
- Are the addressing consistent (same mechanism/level in all parts of the system)

When checking this, it is good to think in terms of security architecture.

E.g., we don't want several different authentication systems for the same type of identity checking.

But, we don't want to rely too much one any single technical (or administrative) solution. Need some redundancy and defense-in-depth. So we want some overlapping mechanisms and a certain amount of redundancy – but it should be by design (and not some ad hoc response).

λ
λ λ

MQTT, Project, Lab(s), Assignment(s), etc (25%)

MQTT CHEATSHEET

MQTT client(=publisher, subscriber)

Clients subscribe to topics to publish and receive messages. Thus subscriber and publisher are special roles of a client.

MQTT server(=broker)

Servers run topics, i.e. receive subscriptions from clients on topics, receive messages from clients and forward these, based on client's subscription, to interested clients.

Topic

Technically, topics are message queues. Topics support the publish/subscribe pattern for clients. Logically, topics allow clients to exchange information with defined semantics.

MQTT Control Packet Structure

The structure is formed by the aggregation of 3 sub-structures: fixed header, variable length header and payload.

Basic details and Terminology

MQTT Control Packet
A packet of information sent across the network

Application Message
The data carried by the protocol

Topic Name
A label attached to an Application Message that can be subscribed by clients

Topic Filter
An expression used to express interest in one or more topics (can use wildcards)

Session
Stateful interaction between a client and a server

***Note**: TCP ports 1883 (default) and 8883 (TLS) are commonly used for MQTT.*

MQTT Control Packet Structure: Fixed Header

Control Packet Type
4 bit representation of the packet type.

Flags
4 bit flags specific to each packet type.

Remaining Length
Number of bytes remaining in the packet.

λ
λ λ

Note #1: Control Packet Type and the Flags are stored in a single byte.
Note #2: The Remaining Length does not include the bytes used to encode itself

Data Representations

Bits
Bits in a byte, from 7 (MSB) to 0 (LSB).

Integers
Big-endian ordered, 16-bits

Strings
UTF-8 strings, prefixed by its length

MQTT Control Packet Structure: Variable Length

Packet Identifier
Used to establish a relationship between different MQTT Control Packets

Payload
A payload associated with the MQTT Control Packet

Note: for the PUBLISH control packet, the payload is the application message.

Quality of Service

- At Most Once (0)
  - Messages are delivered according to the delivery guarantees of the underlying network (TCP/IP)
- At Least Once (1)
  - Messages are guaranteed to arrive, but there may be duplicates.
- Exactly Once (2)
  - This is the highest level that also incurs most overhead in terms of control messages and the need for locally storing the messages.

Topic Names

Must be at least 1 character long
Case sensitive
Can include spaces
Name structured divided by slashes
Example: /news/sports/europe

Wildcards

Multi level #
Used to match any number of levels within a topic tree, including the parent level itself.

Single level +
Used to match a single level within a topic tree.

λ
λ λ

Reserved $
Topics starting with the dollar sign ('$') are reserved for server purposes and should not be used by clients.

Notes: Multi-level wildcard must always be the last symbol on the filter. Either on its own or preceded by the topic level separator.
Single-level: can be used in conjunction with the multi-level one.

Examples:
+/sports/# - valid
sports+ - not valid

MQTT Control Packets

Type	Description	**Type**	Description
CONNECT	Client request to connect	**SUBSCRIBE**	Client subscribe request
CONNACK	Connect ACK	**SUBACK**	Subscribe ACK
PUBLISH	Publish message	**UNSUBSCRIBE**	Unsubscribe request
PUBACK	Publish ACK	**UNSUBACK**	Unsubscribe ACK
PUBREC	Publish received	**PINGREQ**	Ping request
PUBREL	Publish release	**PINGRESP**	Ping response
PUBCOMP	Publish complete	**DISCONNECT**	Client disconnecting

CONNECT
Important elements:

- Connect Flags: to specify the behavior of the connection.
- Keep Alive: maximum time interval, in seconds, that can elapse between client transmission of control packets.

Connect Flags:

- Clean Session: controls the lifetime of the session state (0 to resume state, 1 to discard previous state).
- Will Flag: indicates that a will message is to be sent upon dirty client disconnection.

λ
λ λ

- Will QoS: indicates QoS level for the will message.
- Will Retain: indicates the retain policy for the will message.
- Password: indicates whether (1) or not (0) a password must be present in the payload.
- User Name: indicates whether (1) or not (0) a username must be present in the payload.

Payload:
Length prefixed fields whose presence is determined according to the value of flags in the variable header. The fields are, in order: client identifier, will topic, will message, username, password.

CONNACK
Sent by the server in response to a connection request sent by a client. The important elements on the packet structure are: connect acknowledgement flags and the connect return code.

Note: if a session is already present and the connection request does not have the clean session, the server must set the session present flag to 1.

CONNACK: Error Codes
Connection accepted
0x00
Connection refused (protocol version)
0x01
Connection refused (identifier rejected)
0x02
Connection refused (server unavailable)
0x03
Connection refused (bad user/password)
0x04
Connection refused (Unauthorized)
0x05
Reserved
6-255

PUBLISH
This packet is used to transport application messages for a client to a server or from a server to a client. The important elements on the packet structure are:

- DUP Flag: indicates whether this is the first time the message is being sent (0) or whether it might be a re-delivery attempt (1) of a previous message.
- QoS Level: quality of service for an application message.
- Retain Flag: indicates that the application message and its QoS must be stored in the server and delivered to future subscribers of that topic.

Note: The QoS flag affect how many messages can be stored on the server and sent to the client.

Protocol Exchange: QoS 1
Client/Server protocol interaction:

1. Client ⟶ PUBLISH ⟶ Server

λ
λ λ

2. Client ← PUBACK ← Server

Protocol Exchange: QoS 2

Client/Server protocol interaction:

1. Client → PUBLISH → Server
2. Client ← PUBREC ← Server
3. Client → PUBREL → Server
4. Client ← PUBCOMP ← Server

Security

TLS is the recommended cryptographic protocol to be used with MQTT. Implementations should use port 8883.

MQTT-S (Extension of MQTT for WSNs)

WSN(Wireless Sensor Networks) usually do not have TCP/IP as transport layer. They have their own protocol Stack such as ZigBee on top of IEEE 802.15.4 MAC Layer. Thus, MQTT which is based on TCP/IP cannot be directly run on WSNs. WSNs are connected to traditional TCP/IP networks through gateway devices. MQTT-S is a largely based on MQTT, but implements some important optimizations for wireless networks:

- Topic string replaced by a topic ID (fewer bytes necessary)
- Predefined topic IDs that do not require a registration
- Discovery procedure for clients to find brokers (no need to statically configure broker addresses)
- Persistent will message (in addition to persistent subscriptions)
- Offline keep-alive supporting sleeping clients (will receive buffered messages from the server once they wake up.

LABS

- config.txt
  - dtoverlay=dwc2
- cmdline.txt
  - modules-load=dwc2,g_ether (after rootwait)
- how to enable ssh raspi Zero
  - create empty file ‘ssh’ in root dir
- raspi-config
  - additional settings for raspberry
- change
  - Password aging
- RNDIS
  - **R**emote **N**etwork **D**river **I**nterface **S**pecification

λ
λ λ

TEST EXAM 2017 DAT235

Blå = Nøkkelelementer
Grønn = Viktig kunnskap
Oransje = Tilleggskunnskap
Rød = Dybde/Ekstra-Kunnskap

The Basics of Information Security (25%)

1. 1 **What is meant by "defense in depth"?**

Defence in depth is meant by having multiple layers of security, so that in the event that one of the layers gets breached or compromised so the attack cannot commence further beyond that point. This is to reduce the attack surface, and as a result provide a form of security redundancy.

1.2 **What is multi-factor authentication? (give examples)**

A multi-factor-authentication is having a 2 step authentication or more that usually combines something you have(key-card, phone, ID), with something you know(passphrase), or something you are(bio-metrics).

1.3 **What is meant by the "Principle of least privilege"?**

The principle of least privilege states that you should never have access to more than the minimum needed, as this reduces the attack surface of a compromised user.

1.4 **What is "Kerckhoff's principle"?**

The Kerckhoff's principle states that even if everything in a system is public knowledge, save for the actual key to the system- Then the system should still remain safe. This is in essence the polar opposite to the security through obscurity principle.

1.5 **What is general rule for "breach notification" in the EU General Data Protection Regulation (GDPR)?**

The EU-GDPR for breach notification states that you should notify the appropriate regulator no later than 72 hours after the breach.

- EU General Data Protection Regulation - Article 33
  - §1 In the case of a personal data breach, data controllers shall without undue delay notify the appropriate regulator of the breach.
    - §1.1 This notification should take place no later than 72 hours after the breached party has become aware of the incident.

1.6 **Shortly explain what is meant by phishing and spear phishing.**

Phishing means to "fish" or gather information by making the user put/write in the information you want, usually done by social engineering. One of the common methods is to use emails to make the target write in information that they should not give away.
Spear phishing is specifically targeting the leaders and/or those with high access to either funds or other useful items of interest.

λ
λ λ

1.7 **What is the main difference between a cryptographic hash function and a MAC function?**

The difference between a cryptographic hash and a MAC function is basically one has a key and the other doesn't. They are designed for different purposes, but also has requirements for each. A cryptographic hash has in purpose to be able to verify that data has not been tampered with or that a recipient is who they claim they are. For example that if there is a release of a new software, no matter how minute changes is done to it there should be a clear way of telling that changes has been done. However, if i now would be able to possess the secret key in question used to create that cryptographic hash, then i would be able to "forge" a valid hash for that said software.

This is where MAC functions comes in. A MAC function should only be able to be replicable for the original instance. For example even if i were to acquire the secret key, if i were to try and forge a session, other than one specifically made for me, then the result should differ.

1.8 **What is the main purpose of security hardening? Mention the most common elements of the security hardening process.**

Remove unwanted software and/or modules that are not needed for the daily operation of the system. Example: we don't need a drawing software on a system that's running a server processing numbers..

1.9 **What are the advantages to using a network layer security solution? Are there any disadvantages?**

Link layer security is node to node, while network layer security is end to end.
You also leak all information below the current layer you are operating security on.

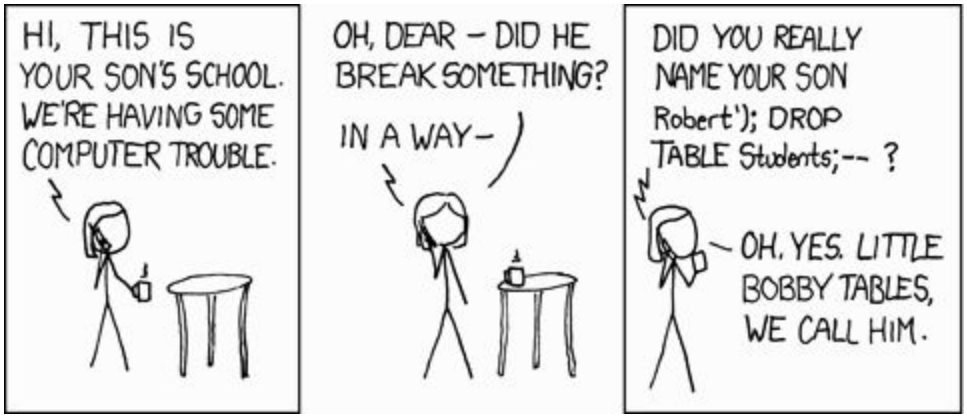
1.10 **Explain what the main difference between input validation and input sanitation are?**

Input Validation is checking that the information typed in corresponds to a valid form of response: If you have a function that adds 2 numbers together, then you want the input to be a integer or a float, and would not want a string.

Input Sanitation is making sure that the whole input only does what it was intended for, within its intended function. An example could be that one uses a semicolon ";" to escape the current function to get higher access, or access to another function which it should not have-

λ
λ λ

theinput is legitimate but i can use it for illegitimate purposes.



Threat Modeling (25%)

2.1.1 **What is a model?**

Abstract simplification of your system, without unnecessary details. A representation of your system.

2.1.2 **In the STRIDE methodology: Explain the "what are you building" step.**

A graphical drawing/representation of your system. With main trust boundaries/zones.

1. What are you building – Define the model (architecture)
   - The essential parts of the model:
     - Interfaces and trust boundaries (trust zones)
     - Define the actors (processes, people)
     - Define the data storages and data flows
   - Define the assets
     - Something an attacker wants
     - Something you want to protect
     - A stepping stone towards a) and/or b)

2.1.3 **What is meant by the "Elevation of Privilege" threat?**

It is meant that there is a possibility that a input or command may be escaped its current privilege level to gain access to privilege it shouldn't have or social engineer an administrator with a higher privilege/ access level.

2.1.4 **What are the advantages/disadvantages of the «breadth first» approach?**

Advantages are that you manage to cover all problem areas, while the disadvantage is that each vulnerability might not be addressed properly in proportion to their risk.

2.1.5 **Do you have to address all threats? (Explain why or why not)**

All threats needs to be addressed but you may choose not to act on the threat until a later stage as long as it is documented.

2.2 **Practical Threat Modeling (12.5%)**

λ
λ λ

IoT Security (25%)

3.1 **Below is the commands for updating Raspbian (apt-get update && apt-get dist-upgrade)**

- **What are the drawback to updating Raspbian this way?**

Using the "apt-get update" commands only updates the program(s) that you have installed on the device, and doesn't take into account the fact that you may have dependencies in other software/code/scripts that gets obsoleted or made null. Not having a backup of your files to do a rollback could lead to the device being rendered useless for the task it was originally set to do. It could also increase the attack surface.

3.2 **What is the main drawback to having security on the link layer?**

One of the main drawbacks is that most information travels more than one node-jump, and link-layer security can only guarantee safety from node-to-node, and not end-to-end.

3.3 **What is meant by "security by default"?**

Security by default means to have the standard setup as strong as possible and only make changes for less security as needed.

3.4 **What are the steps in the Privacy Impact Assessment (PIA) process?**

The PIA process is split into 4 main categories:

- Project Initiation
- Data Flow Analysis
- Privacy Analysis
- Privacy Impact Assessment Report

1. Project Initiation; This step is where you define the scope of the PIA process (which varies by organization), if the project they are running is in early stages and detailed information is unknown the organization may choose to do a Preliminary PIA, and then a full PIA once it gets off the ground.
2. Data Flow Analysis; This step involves mapping out the proposed business process as it regards personal information, identifying clusters of personal information, and creating a diagram of how the personal information flows through the organization as a result of the business activities in question.
3. Privacy Analysis; This step requires all personnel involved with the movement of private information to complete privacy analysis questionnaires, as well as secondary check-ins on the answers to the questionnaires which require more detail, and discussion of the privacy issues and implications brought up as a result of the questionnaires.
4. Privacy Impact Assessment Report; This step requires the organization to create a documented evaluation of the privacy risks and potential implications of said risks brought up by the outcomes of the previous steps, as well as a discussion of possible efforts that could be made in order to mitigate or remedy the risks.

λ
λ λ

3.5 **What is meant by "informed user consent"?**

Informed user consent is the act of making the user aware that if a person were to continue using their program/service then certain conditions will apply. This is usually done by a EULA or a ToS.

3.6 **What are the main advantages of using MQTT instead of HTTP for IoT messaging?**

The MQTT protocol has a lot less "waste data" so that equipment that has a limited amount of power (PLS and similar), doesn't use unnecessary power to transfer a 2kb overhead when it only relays a few bits of information per cycle/read.

3.7 **Can you run MQTT over Bluetooth? (explain)**

Yes, you can run IP and TCP over bluetooth(having a IP stack for standard MQTT). MQTT-SN is a version which allows you run over a serial line communication.

3.8 **What is a vulnerability?**

A vulnerability is a area of your software/system that has liability to be used for an attack.

- **Compare vulnerability with threat, and explain the differences and how they are related.**

A threat is the acknowledgment of a vulnerability, and a classification of how much it will affect you.

3.9 **What is an attack?**

An attack is an attempt to exploit a vulnerability.

3.10 **How can you reduce the attack surface of an operating system?**

You can reduce the attack surface by removing bloatware and functions that would not be needed for the operation of the system.

MQTT, Project, Lab, Assignments.. (25%)

4.1.1 **What does it mean that a client is a publisher?**

That it publishes messages to a subscriber client in the system.

4.1.2 **Which protocols do MQTT normally run over?**

MQTT normally runs over the TCP-IP stack.

4.1.3 **Explain what a topic is in MQTT**

A topic in MQTT is a queue and something you can subscribe to.

4.1.4 **What is the difference between QoS 1 and QoS 2 in MQTT?**

QoS 0 (*At most once* (0))

Send it off, hope everything goes okay ;)

QoS 1 (*At least once* (1))

It has to arrive at least once.

QoS 2 (*Exactly once* (2))

λ
λ λ

exactly one time it's certain that it'll arrive

4.1.5 **What is the "Last will" in MQTT?**

Last will, connect to topic, leave message, that when it disappears.. publish message EXCEPT, if disconnect no message.

4.2.1 **How do you install the paho python client?**

"Pip install paho-mqtt"

"Pip3 install paho-mqtt" for python 3

4.2.2 **Assume that you have a file high-integrity.text stored in your home directory ($HOME). Currently, only you have rights to the file (read, write).**

UGO is an acronym for "User, Group and Other" and stands for each group of permissions in the 3 bit file-permission system in linux.

"Chmod 755"

4.2.3 **Over which port do you connect ssh?**

Standard is port 22, but as long as you adjust for listening could work anywhere between 1-65535.

4.2.4 **Where are the passwords stored on Raspbian?**

All passwords are stored in "/etc/shadow"

4.2.5 **Explain what the following command does (explain the options):**

Mosquitto will with "version" of

- mqtt -V mqttv311
  - protocol-version, default(mqttv31)
- "topic" -t dat235/test
- "message" -m ""Dette er en test"
- "specify the quality of service to use (default is 0) "-q
- "debug" -d function activated.