# Uge 15

I denne uge blev jeg klar over hvad mine endelige emner skal være. Da jeg indså at IT-sikkerhed og Data Lake er nogle meget bredte emner. Hvilket var et problem, som jeg havde inden påsken. Hvor jeg valgte at holde ferie i påsken, for at blive klar over hvad jeg ville arbejde videre med.

# Uge 16

Mandag:

I dag læste jeg op omkring Threat modeling i forhold til STRIDE, da jeg tænker at det er meget spændende at lære om hvordan man i praksis finder ”weak points” i ens systemer og hvordan man så fixer dem. Desuden skrev jeg til min Onkel omkring min realisering, som jeg fik i uge 15.

Tirsdag:

Begyndte jeg at læse op omkring hvordan man laver et dataflow diagram, da jeg indså at for at benytte stride eller andre threatmodeling teknikker, så skal man vide hvor der er mulighed for at blive angrebet igennem. Der er et data flow diagram, et overblik over de forskellige processer i ens program. Hvor imellem hver process så er der en mulighed for en udefrakommende kan angribe kommunikation mellem de 2 processer.

Onsdag:

Desuden snakkede jeg med Martin om Data Lake vs Data Warehouse…

Lavede jeg et usecase – misusecase diagram, for at fremvise hvilke angrebsvinkler en udefrakommende kan have på systemet.

Uge 17:

Mandag:

Tirsdag:

Onsdag:

Torsdag:

Fredag.

Uge 18:

Mandag:

Tirsdag:

Onsdag

Torsdag:

Fredag:

Data Lake:

**Viden**

* Hvad er big data?
  + De 5 v’er
* De 3 zoner
* Den overordnet struktur
* Tierbased storage

**Færdigheder**

* Opsættelse af en data lake
* Implementering af en event driven arkitektur
* Dataflow diagram

**Kompetence:**

* Finde ud af hvornår man bruger batch processing vs stream processing.
* Diskutere Eventdriven vs statedriven arkitektur.
* Data refinement

# IT- Sikkerhed

**Viden:**

* Digital Signatures
* SSL / TSL
* Typer af angreb

**Færdigheder**

* Digital Signatures
* Threat Modeling
* Misuse cases

**Kompetancer**

* Udvikle et system med et IT-sikkerhed i fokus.
* Håndtere situationsbestemt tilpasning af sikkerhedsmetoder og processer til et konkret praksisnært projekt
* Varetage valg af sikkerhed i forbindelse med udvikling af microservices.

Threat agent: Is an induvial or group that is capable of carrying out a particular threat. It is fundamental to identify who would want to exploit the assets of a company, how they might use them against the company, and if they would be capable of doing so.

Impact: A measure of the potential damage caused by a particular threat. Impact and damage can take a variety of forms. A threat may result in damage to physical assets, or may result in obvious financial loss. Indirect loss may also result from an attack, and needs to be considered as part of the impact.

Likelihood is a measure of the possibility of a threat being carried out. A variety of factors can impact the likelihood of a threat being carried out, including how difficult the implementation of the threat is, and how rewarding it would be to the attacker.

Controls are safeguards or countermeasures that you put in place in order to avoid, detect, counteract, or minimize potential threats against your information, systems, or other assets.

Preventions are controls that may completely prevent a particular attack from being possible.

Mitigations are controls that are put in place to reduce either the likelihood or the impact of a threat, while not necessarily completely preventing it.

A data flow diagram is a depiction of how information flows through your system. It shows each place that data is input into or output from each process or subsystem. It includes anywhere that data is stored in the system, either temporarily or long-term.

A trust boundary (in the context of threat modeling) is a location on the data flow diagram where data changes its level of trust. Any place where data is passed between two processes is typically a trust boundary. If your application reads a file from disk, there's a trust boundary between the application and the file because outside processes and users can modify the data in the file. If your application makes a call to a remote process, or a remote process makes calls to your application, that's a trust boundary. If you read data from a database, there's typically a trust boundary because other processes can modify the data in the database. Any place you accept user input in any form is always a trust boundary.