10.

Security design og principper for sikkert design

Sikker software, hvorfor?

- Usikker software
- GDPR
  - Etik
  - Ansvarlighed
- ‘Prevention is cheaper than the cure’
- NotPetya omkostninger på $1.2B

Phase	Relative cost to correct
Definition	$1
High-level Design	$2
Low-level Design	$5
Code	$10
Unit test	$15
Integration test	$22
System test	$50
Post-delivery	$100

Hvordan bliver software usikkert?

- Design fejl
  - Privelegier
  - Insecure defaults
  - Defence in depth
- Implementations fejl
  - Input validering
  - Fejlhåndtering
- Maintainence
  - Patching
  - Udfasning
- Højkvalitetssoftware = Bedre sikkerhed

Privacy by Design

- De 7 C’er
  - Comprehension – *hvem opsamler data?*
  - Consciousness – *hvor og hvornår opsamles data?*
  - Choice – *informeret valg*
  - Concent - *samtykke*
  - Context – *kontext styrer præferencer*
  - Confinement – *indsamling begrænset af formål*
  - Consistency – *rimelig forståelse for databrug*

Privacy by Design - Strategier

- Dataorienterede
  - Minimise
  - Hide
  - Separate
  - Aggregate
- Procesorienterede
  - Inform
  - Control
  - Enforce
  - Demonstrate

Security by Design

- Risikobaseret analyse
- CIA
  - Confidentiality
  - Integrity
  - Availability
- Open Web Application Security Project (OWASP)

Security principles

- *Minimize attack surface*
- *Establish secure defaults*
- *Principle of least privilege*
- *Principle of defence in depth*
- *Fail securely*
- *Don’t trust services*
- *Seperation of duties*
- *Avoid security by obscurity*
- *Keep security simple*
- *Fix security issues correctly*