3.

Sikkerhed i web applikationer

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
  - Unpatched software
  - Legacy systemer
- Højkvalitetssoftware = Bedre sikkerhed

OWASP top 10

- *Broken Access Control*
- *Cryptographic Failures*
- *Injection*
- *Insecure Design*
- *Security Misconfiguration*
- *Vulnerable and outdated components*
- *Identification and authentication failures*
- *Software and Data integrity Failures*
- *Security Logging and Monitoring Failures*
- *Server-Side Request forgery*

Frameworks

- Python XML
- Django
- Representational State API (REST API)
  - Uniform Interface
  - Statelessness
  - Cacheability
  - Layered architecture

Scan & test

- SSLScan
  - Misconfiguration
- Nikto
  - Farlige filer og programmer
  - Outdated versions
  - Misconfiguration
  - Vulnerabilities