9.

Audit af software

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?

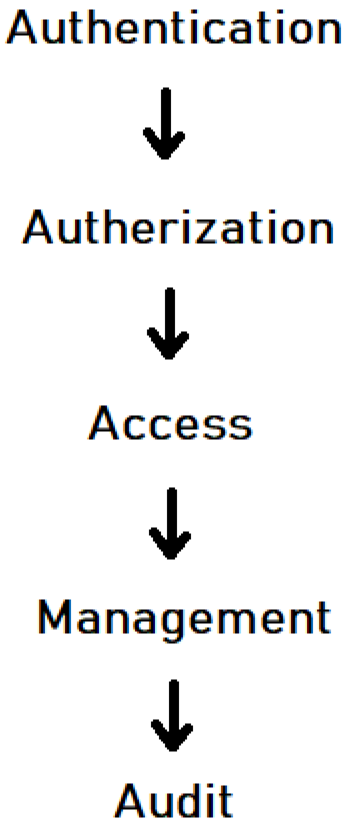
- Design fejl
  - Privelegier
  - Insecure defaults
  - Defence in depth
- Implementations fejl
  - Input validering
  - Fejlhåndtering
- Maintainence
  - Unpatched software
  - Legacy systemer
- Højkvalitetssoftware = Bedre sikkerhed

Patching

- Er det altid korrekt at patche?
- Spectre og Heartbleed
- Feature patches
- Patch modenhed og virksomhed modenhed
- Sikkerhed versus funktionalitet

Access Control

- Hovedelementerne af Access Control
- Discretionary Access Control (DAC)
  - Adgang styres af administrator
- Mandatory Access Control (MAC)
  - Adgang gives baseret på burger eller enheds niveau
- Attribute-based Access Control (ABAC)
  - Adgang gives baseret på attributter
- Role-based Access Control (RBAC)
  - Adgang gives baseret på roller
- Rule-based Access Control
  - Adgang gives baseret på regler (tid og sted)



Authentication

Autherization

Access

Management

Audit

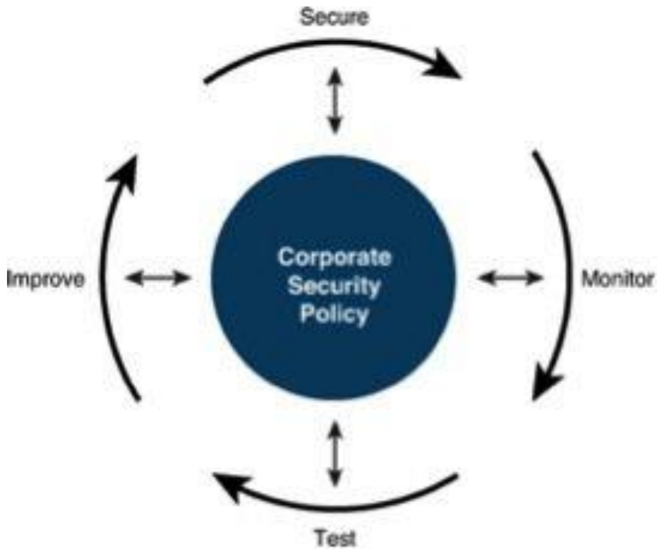
Gennemgang og testing

- Test-Driven Development
- Audit VS Black Box
- Fuzzing VS Code Reading

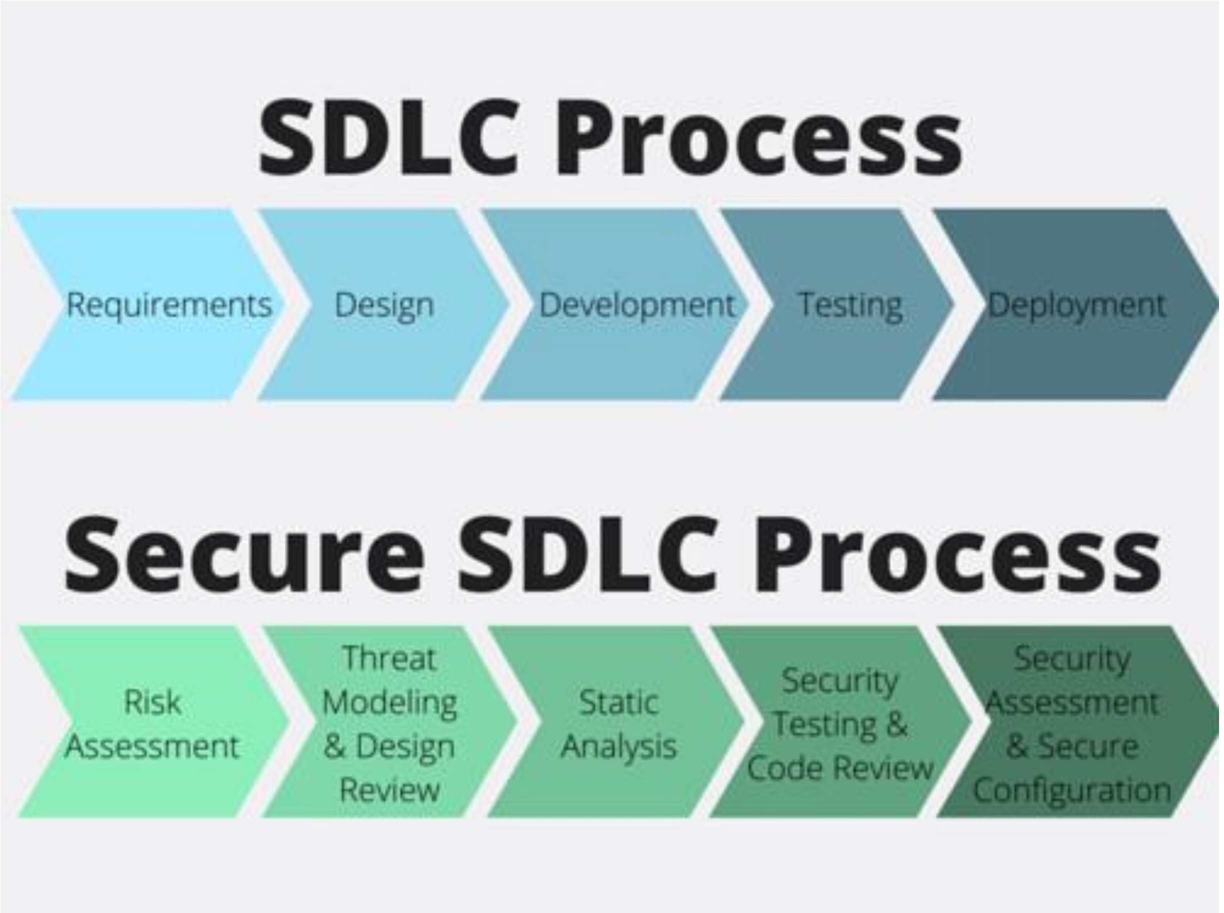
Security is a process

- Bruce Schneier

- … not a product
- Processer og procedurer
  - Vulnerabilities
  - Dokumentation



Secure Software Development Lifecycle



SDLC Process

Requirements → Design → Development → Testing → Deployment

Secure SDLC Process

Risk Assessment → Threat Modeling & Design Review → Static Analysis → Security Testing & Code Review → Security Assessment & Secure Configuration