4.

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

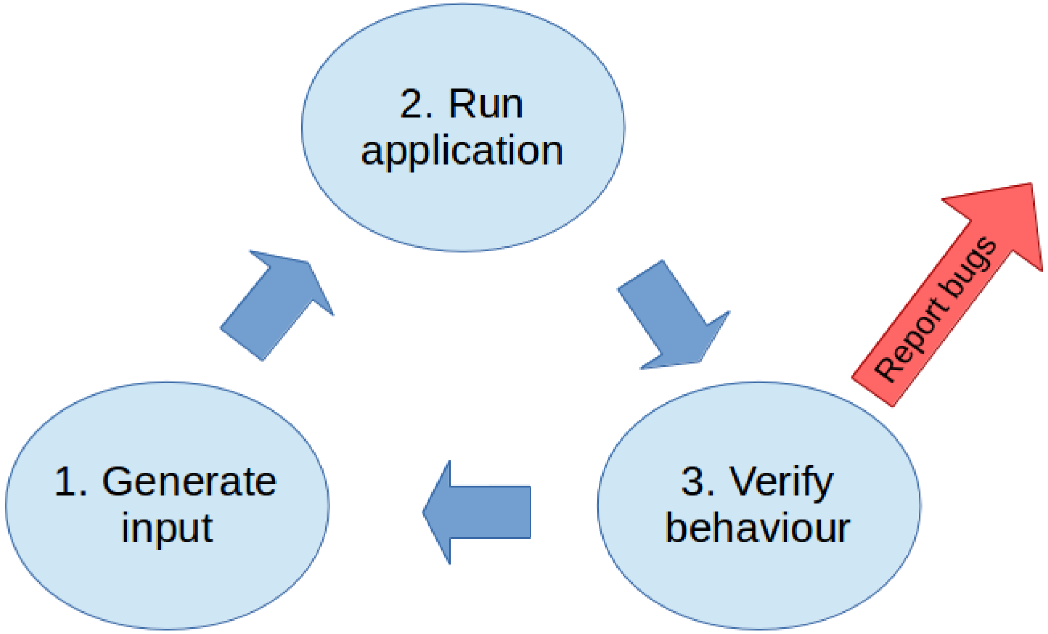
Hvordan bliver software usikkert?

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

Hvad er fuzzing?

“**Fuzzing** or **fuzz testing** is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. The program is then monitored for exceptions such as crashes, failing built-in code assertions, or potential memory leaks.” - *Wikipedia*

- Simplificeret
- Fejl kan lede til kompromitering



Simpelt eksempel

```
$ perl print ‘A’x100
```

- Buffer overflows

Hvad kan man fuzze?

- … Mange ting!
- Inputs
  - Webforms
  - Logins
  - Programmer

Fuzzing tools

- Scapy
- American Fuzzy LOP
- Sulley