(Nog steeds geen) secure software development

Erik Poll

Digital Security group

Radboud University Nijmegen

CIP meeting, March 2017

About me

- research into security since 1999, with interests in
  - smartcards
  - static analyis (esp. program verification)
  - dynamic analysis (esp. automated security testing)
  - more applied security analysis in case studies




- teaching of several courses in
  - our Cyber Security Master (since 2006)
  - our Cyber Security Bachelor (since 2013)

About this talk

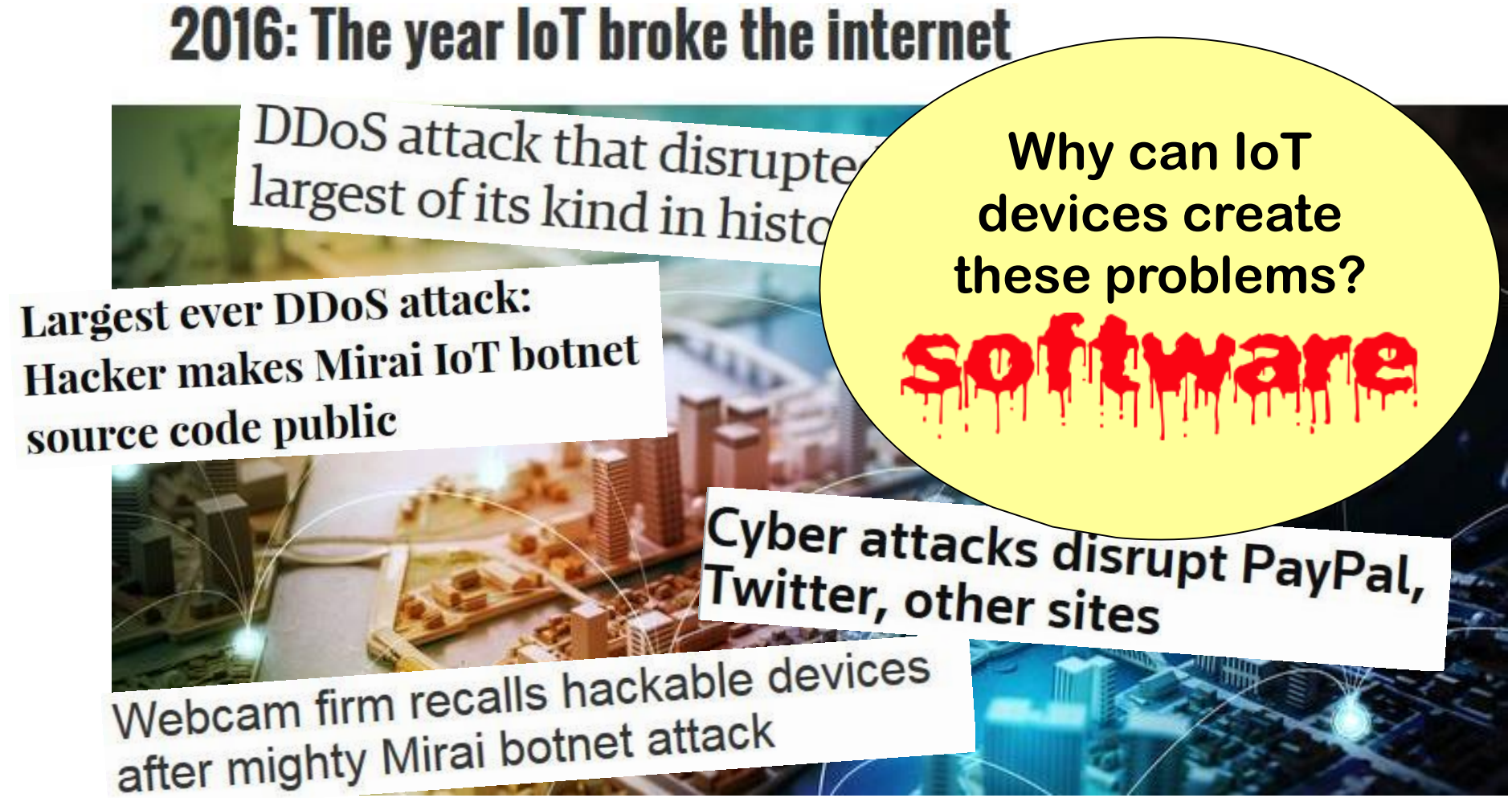
- Some observations about
  - security,
  - software security,
  - security in the software development lifecycle

  in the past two decades

- Some anecdotes, successes, and failures

- Probably (hopefully?) nothing new…

Depressing security news, as usual...



Fact 1:
security is always a secondary concern

- The primary goal of ICT is to provide functionality
- Security risks this introduces are a secondary concern
  - and are ignored or considered too late

Hence:

- Functionality always trumps security
- Without some pressure, security will be overlooked
- Security problems are often externalities
  - ie. *de vervuiler betaalt niet*
  - aka tragedy of the commons

Fact 2:

is no. 1 root cause of trouble

Devices be hacked because they contain 

When it comes to cyber security
software is not our Achilles' *heel*
but our Achilles' *body*

The history of software security

The dark ages of software security

Before 2001, nobody paid much attention to software when it came to security

All textbooks about computer security only talked about

- **cryptography**
- **hashing & salting passwords**
- **Mandatory Access Control (MAC) & Multi-Level Security (MLS)**

A new dawn: Microsoft's security push [2002]

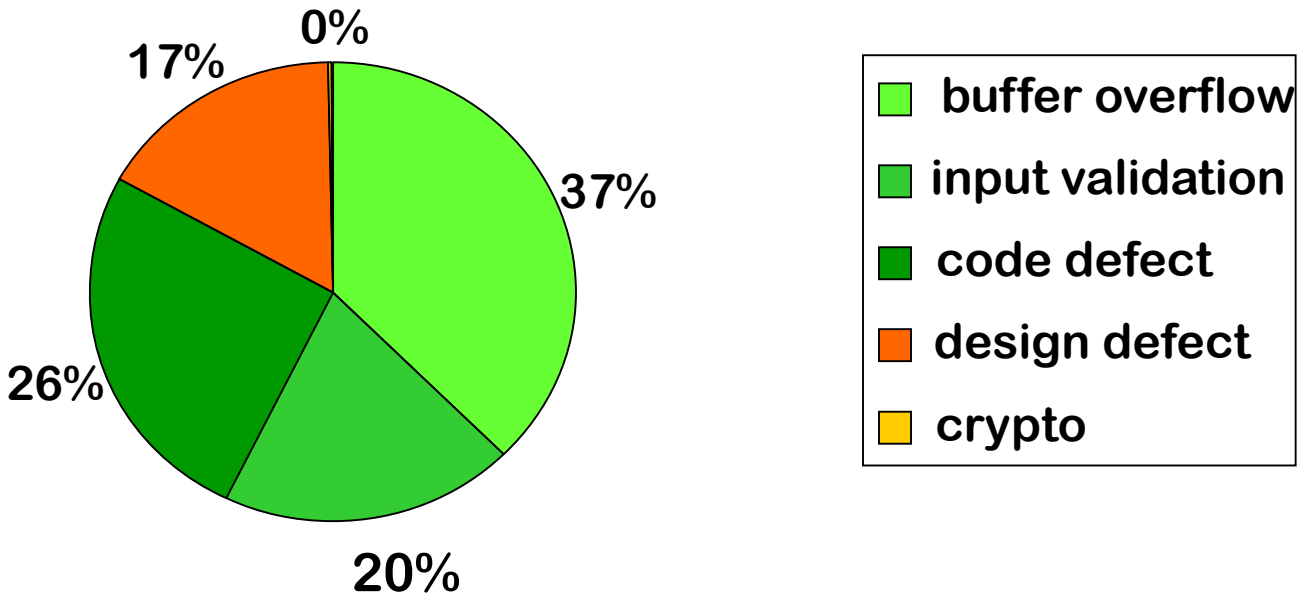
NEWS

MS: "No new code for a month"

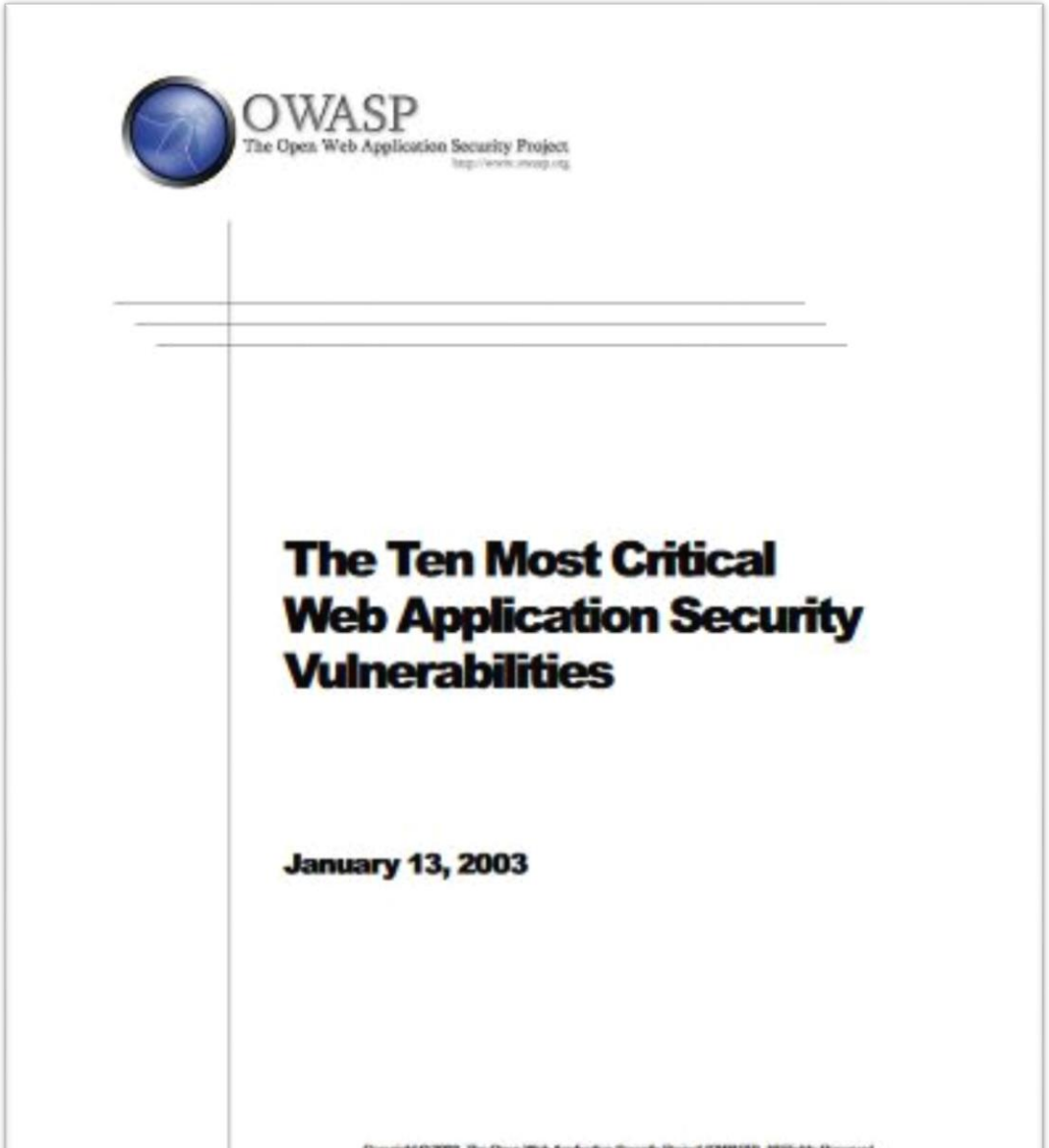
BY

02.06.2002 :: 8:01AM EDT

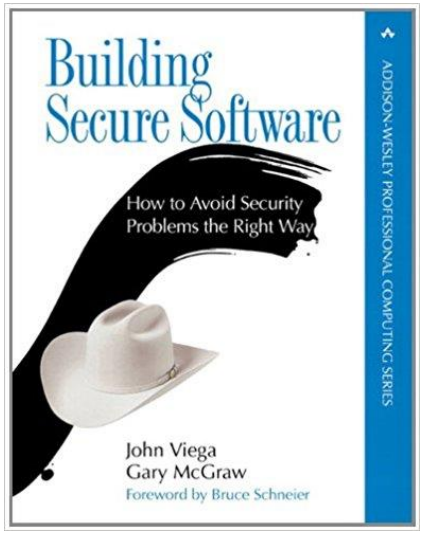
Security bugs found in first security bug fix month



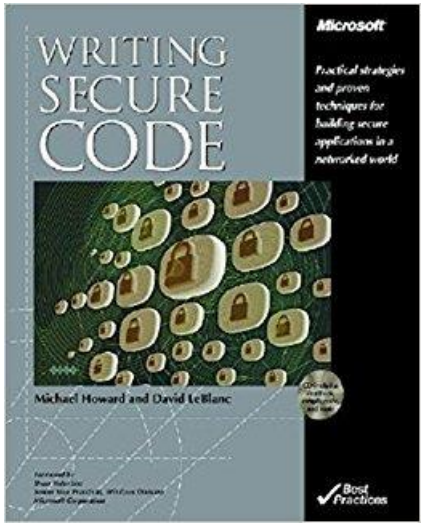
OWASP Top 10 [2003]



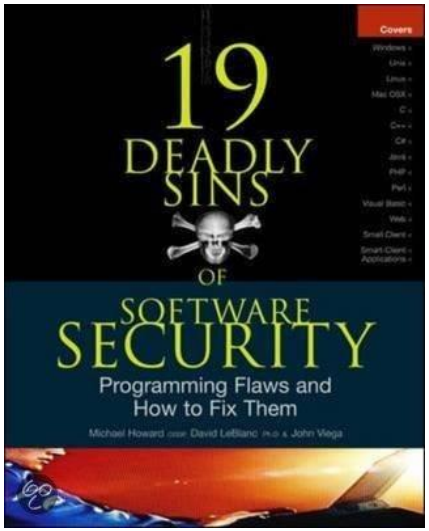
Books on writing secure code



[2001]



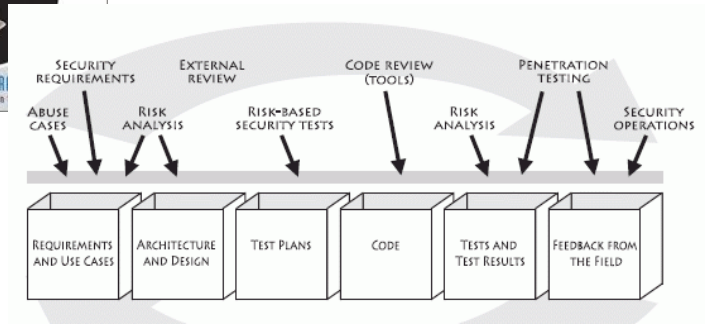
[2001]



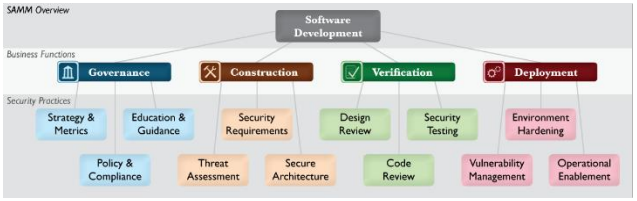
[2006]

Secure development methodologies

Gary McGraw's
Touchpoints
[2006]



Microsoft
SDL
[2006]

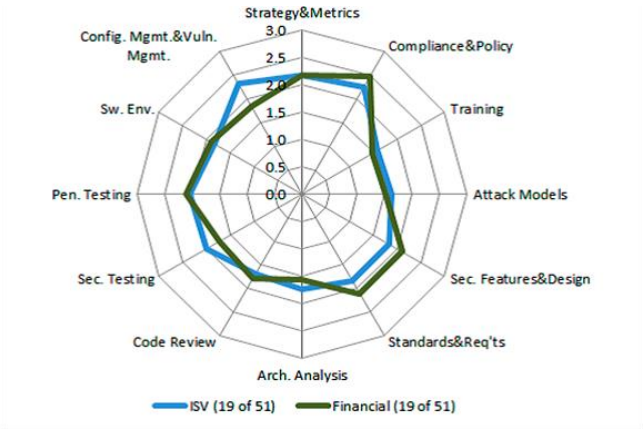


[2008]

Secure development maturity models

Building Security in Matury Model

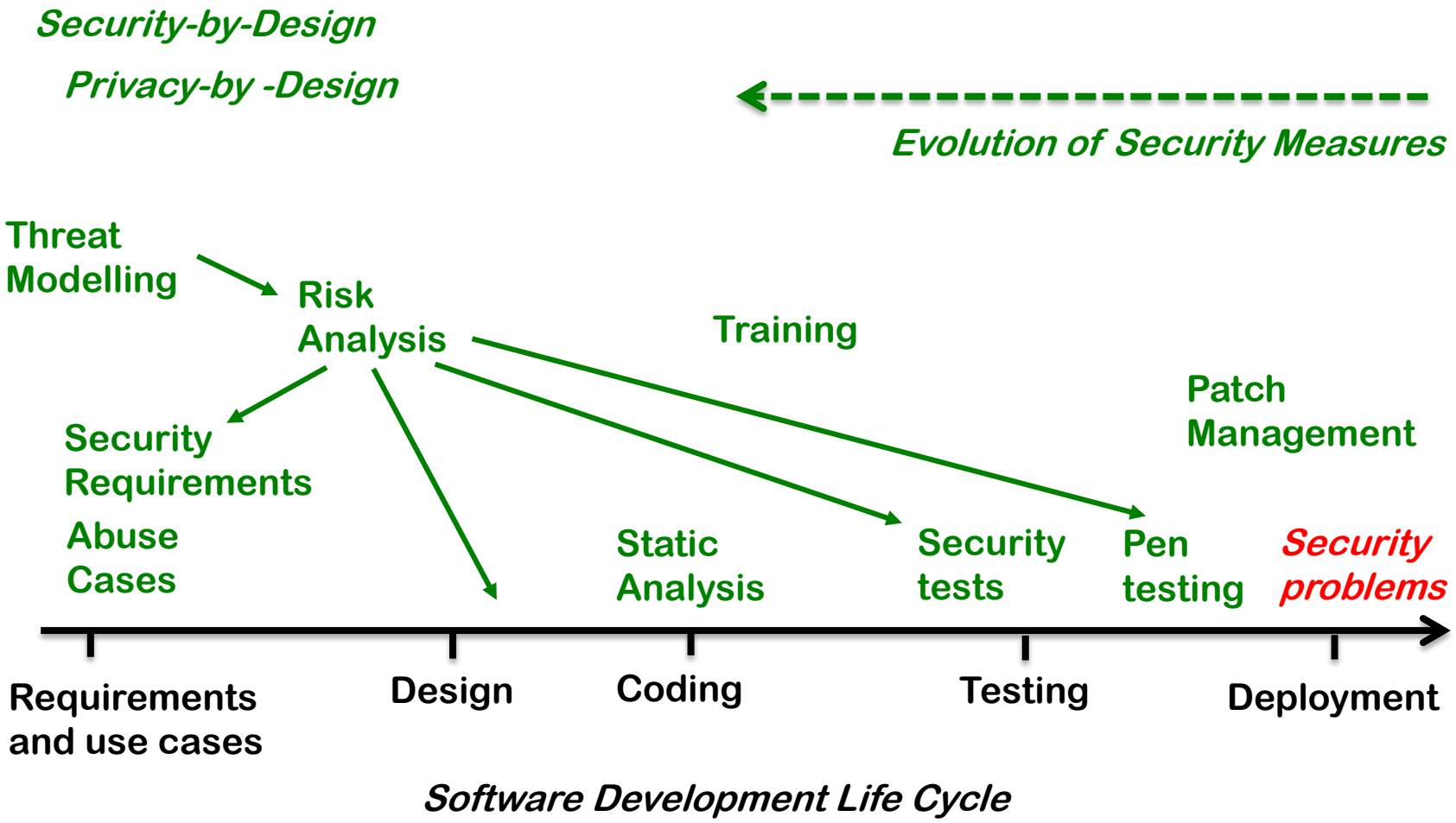
[2006]



Governance	Intelligence	SSDL Touchpoints	Deployment
Strategy and Metrics	Attack Models	Architecture Analysis	Penetration Testing
Compliance and Policy	Security Features and Design	Code Review	Software Environment
Training	Standards and Requirements	Security Testing	Configuration Management and Vulnerability Manage-ment

[2006]

Security in Software Development Lifecycle



So are things getting better?

SSD failure

The root cause of Mirai botnet?

default passwords

USER:	PASS:	USER:	PASS:
root	xc3511	admin1	password
root	vizxv	administrator	1234
root	admin	666666	666666
admin	admin	888888	888888
root	888888	ubnt	ubnt
root	xmhdipc	root	klv1234
root	default	root	Zte521
root	juantech	root	hi3518
root	123456	root	jvbzd
root	54321	root	anko
support	support	root	zlxx.
root	(none)	root	7ujMko0vizxv
admin	password	root	7ujMko0admin
root	root	root	system
root	12345	root	ikwb
user	user	root	dreambox
admin	(none)	root	user
root	pass	root	realtek
admin	admin1234	root	00000000
root	1111	admin	1111111
admin	smcadmin	admin	1234
admin	1111	admin	12345
root	666666	admin	54321
root	password	admin	123456
root	1234	admin	7ujMko0admin
root	klv123	admin	1234
Administrator	admin	admin	pass
service	service	admin	meinsm
supervisor	supervisor	tech	tech
guest	guest	mother	fucker
guest	12345		
guest	12345		

The dark ages continue: software security in education

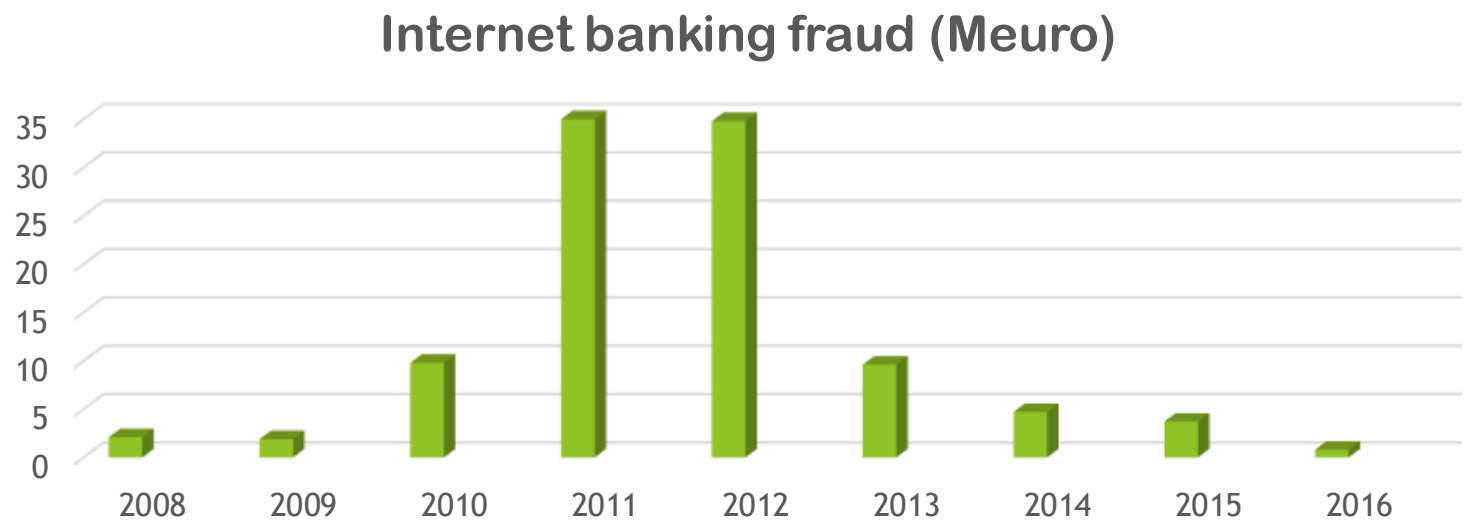
Students are taught programming,
but are they taught to program securely?

- Teaching students C(++) without teaching them about buffer overflows is a case of *criminal negligence*

- The same goes for teaching how to use databases or build websites, without teaching about SQL injection, XSS, CSRF, ...

In Nijmegen first year bachelor students are taught about buffer overflows in Q3 and web security in Q4.

Security-by-Design success?



- Most security risks are not so easy to measure
- Main improvements not due to better software
  - but eg. better detection & response

Security-by-Design failure!

Obvious security flaw in USB-connected internet banking device

- malware on the PC can press OK

How could this flaw be missed?

Who is *really* checking the security?

- Is everyone just trusting someone else?
  Banks, suppliers, Visa, Mastercard,...

Or is Cover-Your-Ass security all that matters, as usual?

SSD failure?

- Software company X hires external pen tester to check products & report back to development team

  But… pen test reports are *not* shared with other development teams within the company

- ***Reason?***

  Worries that pen test reports might leak and cause bad publicity…

Security-by-Design success!

Dutch smart meters will *not* have a remote off-switch

Software security in the past decade

Nothing new in the past decade?

- we are still fighting buffer overflows, SQL injections, and XSS,...
- SSD techniques have been known since 2006

except

- risks are increasing
  - as our reliance on software increases
- attackers are getting cleverer
  - eg ransomware

Are we finally reaching the tipping point to use SSD?

Uitdagingen aan jullie!

- How to get and keep commitment for security in your organisation?
  - How do you check & show (cost)-effectiveness?
  - How do you decide how much resources to spend?
  - Can comparing your security maturity with your peers help?

- What are effective ways to build & share knowledge?
  - within your organisation
  - between organisations
    - incl. in (higher) education

How many of you have attended OWASP NL, BeNeLux, or EU AppSec meetings?