Cyber Security 101

Erik Poll

Digital Security group / ICIS / FNWI

Radboud University Nijmegen

Hacking

vs

Security

vs

Privacy

vs

other problems with digitalisation

Should I, as a computer security researcher, be in the iHUB?

To answer this, I'll discuss two topics

I. Everything you need to know about cyber security

- the relation between hacking and security
- security requirement engineering aka threat modelling
- attacker model

II. How security relates to privacy and other societal problems with digitalisation

Part I.

Cyber Security

aka

Computer Security

aka

Information Security (InfoSec)

Central research questions in my research

1. *Why* do computer systems have so many security issues?
2. *How* can we improve this?

Main ways to create security problems:

1. ‘hack’ the computer
   - eg exploit a zero-day

```
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;

    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;

}
```

2. ‘hack’ the user
   - eg phishing

Pointing the finger at the user is nearly always victim blaming and the badly designed interface is the real cause

Why can computer systems be 'hacked'?

Because they contain SOFTWARE

if **something contains software, it can typically be hacked**

Why?

- **Software is the most flexible, powerful & complex artefact ever produced by humankind.**

  DNA is also software, our minds probably too, so most complex artefacts in nature are also software.

- **Power & flexibility is great ☺ → we can do anything in software**

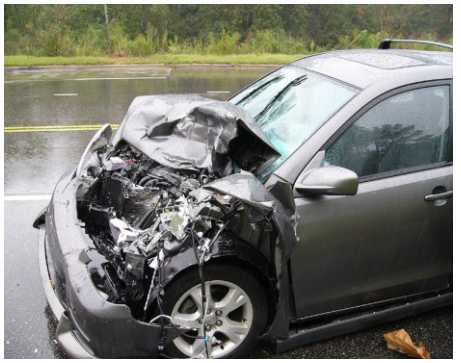
  Computers are *programmable* machines, unlike earlier machines

- **COMPLEXITY is bad ☹ → bugs**

  - **Bugs + (power & flexibility) ☹ ☹ → lots of power to abuse**

    Worst case: attacker can re-program the computer (eg. to encrypt all data to then hold it at ransom)

Software fails differently



- If your *analogue, mechanical* brakes work at 100 km/h
  they also work at 30 or 50 km/h
- If your `digital, computer-controlled` brakes work at 100 km/h
  they might fail in a totally weird way at 31.128 km/h

Software engineering is not rocket science,
but is WAY more interesting & complicated that

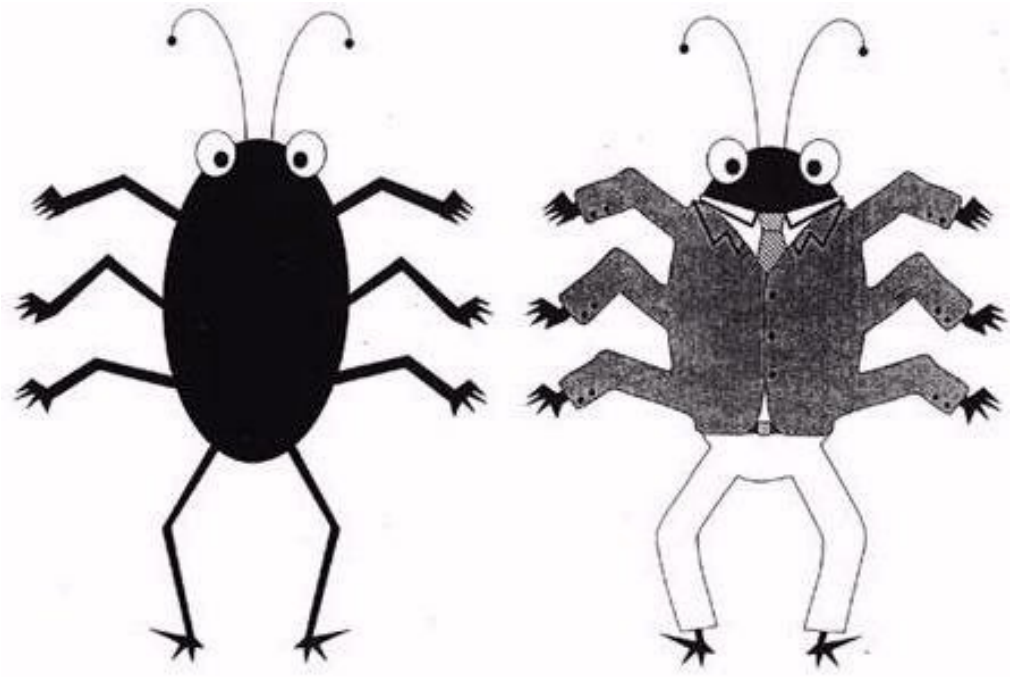
Not all security problems involve hacking or bugs!

- First big computer security problem, back in the 1990s
- Spam does not exploit any bugs or involve any 'hacking'
  It (ab)uses the very features of email:
  - the quick, easy & cheap sending of messages
    in a way that can be automated

Other reasons why spam is a very interesting example:

- Like many security problems, it was originally totally overlooked
- Like many security problems, the solution was not prevention but detection
- As is often the case, this detection then introduces a privacy risk
- Many of the problems the iHUB looks at are a lot like spam,
  in that we did not see them coming…

How software becomes insecure



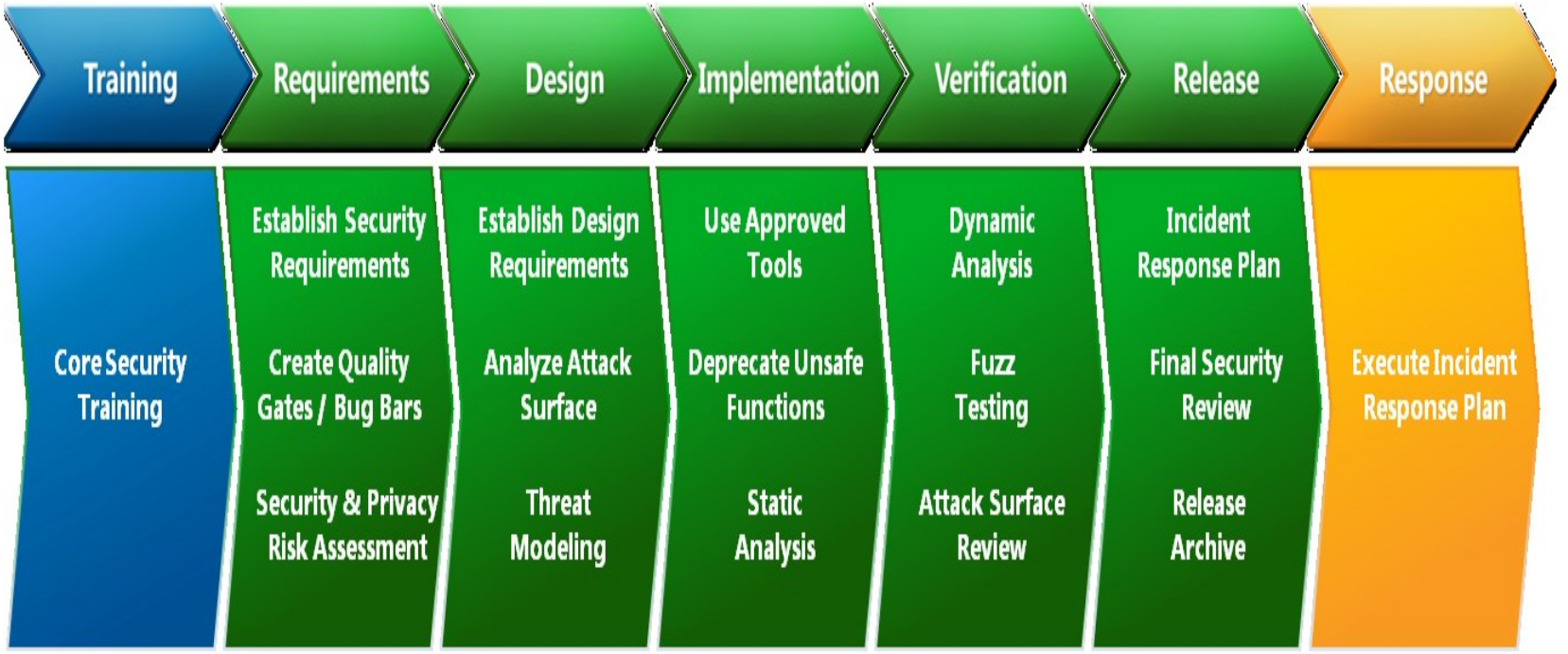
bugs
that can be
exploited

features
that can be
abused

We know how to improve security

By range of measures throughout the software development lifecycle

eg Microsoft SDL or Gary McGraw's BSIMM



Microsoft SDL (Security Development Lifecycle)

No silver bullets

We can add security features to systems

- eg firewalls, anti-virus, intrusion detection, network monitoring, multi-factor authentication, encryption, TLS, VPN, …
- this is more software, namely security software

but that does not make the system secure:

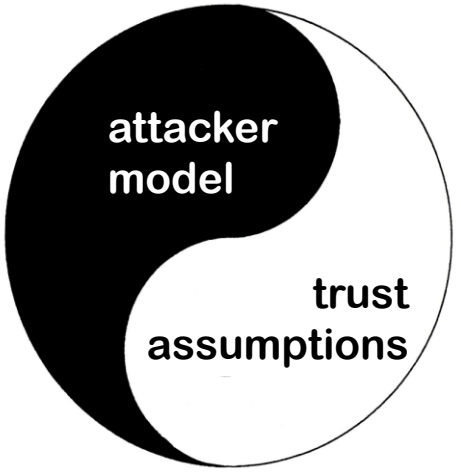
- *all* the software in the systems needs to be secure, not just the security software

First step: What does it mean for system to be secure?

Two sub-questions:

1. *What are the security requirements?*
   - Or*: what are the threats that we worry about?*

2. *What is the attacker model?*
   - i.e*. what are the attacker's capabilities & resources?*
     - Possibly also: attacker's motivation & goals
       The attacker's goals overlap with threats
   - Also: *what are our trust assumptions?*



One security requirement to rule them all

One, generic, baseline security requirements for any system:

the system cannot be hacked

For some systems, this is the *only* security requirement.

This 'negative' property is not very actionable…

Or, a bit less ambitiously,

the system cannot be hacked *without us noticing*

‘Threat modelling’/ ‘Risk Analysis’/ ‘Security Design’

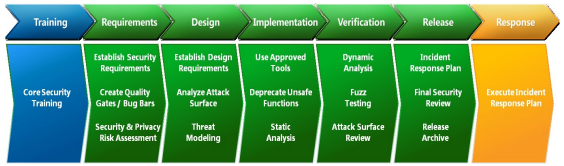
- As part of the design process, threat modelling should go hand in hand with risk assessment to guide/be guided by (security) design decisions
  - Messy, iterative process!
- Outcome: security functionality or security controls, esp. for
  - access control: authentication & authorisation
  - monitoring: detection & reaction if things go wrong

  aka AAAA (Authentication, Authorisation, Audit, Action)

But never forget

all functionality needs to be secure, not just the security functionality

We know how to improve security



BUT

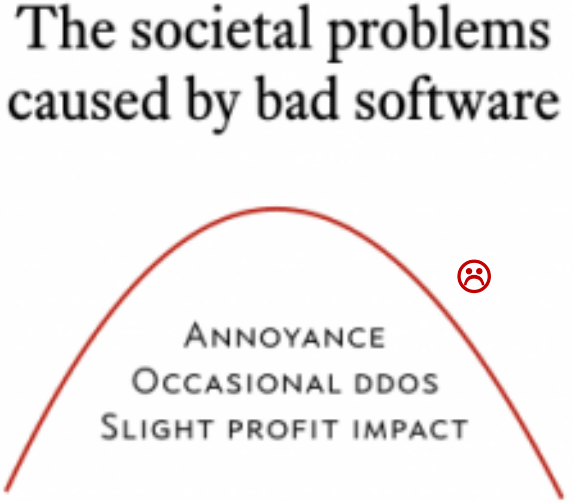
1. There will always be bugs and unforeseen abuse of features
   given the complexity of software ☹

2. Better security costs time & money ☹
   and measuring security & security benefits is hard

Why software remains insecure

Software's win/loss ledger

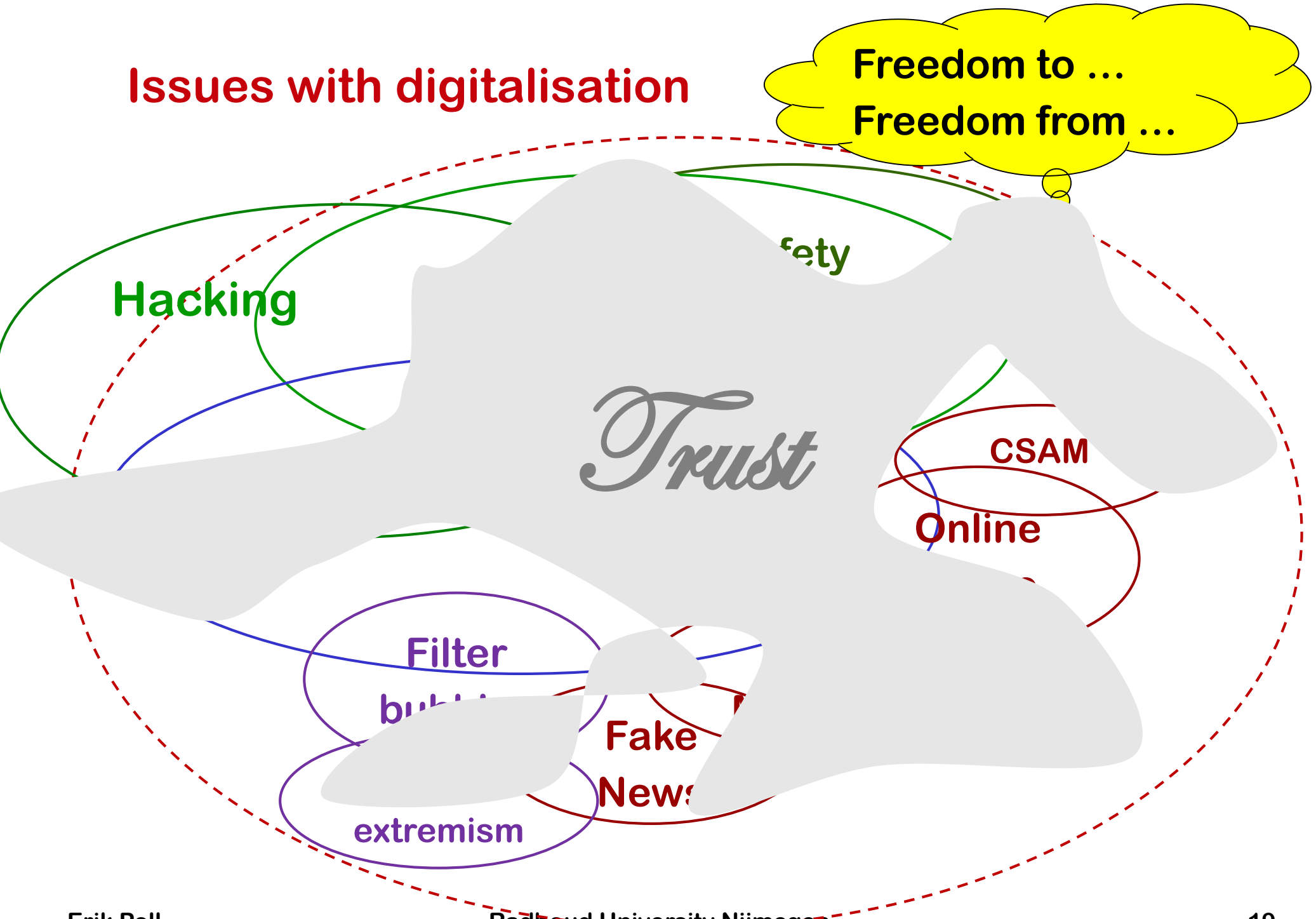
Benefit to humanity	Unfathomable
People killed by bad software	Basically zero
Times the internet crashed	Basically never
Chance of living without it	zero
Number of people helped	billions



[Daniel Miessler, https://danielmiessler.com/blog/the-reason-software-remains-insecure/, 2018]

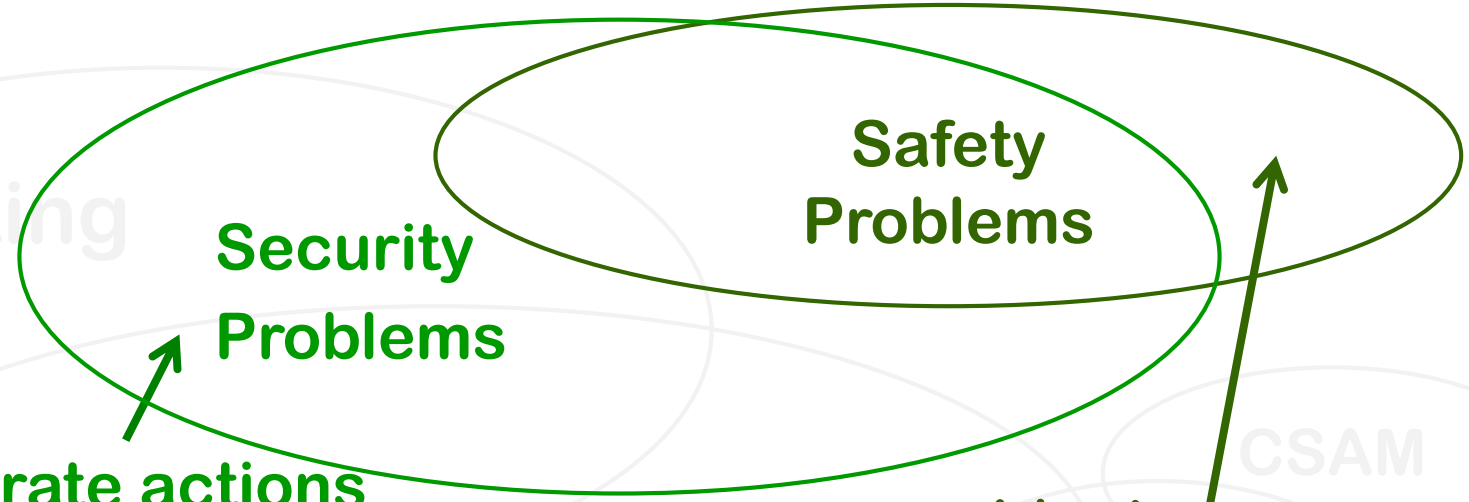
Part II.
What about other problems, besides security?

Issues with digitalisation



Security vs Safety

Security vs Safety



deliberate actions
by attacker

eg. phone hacking

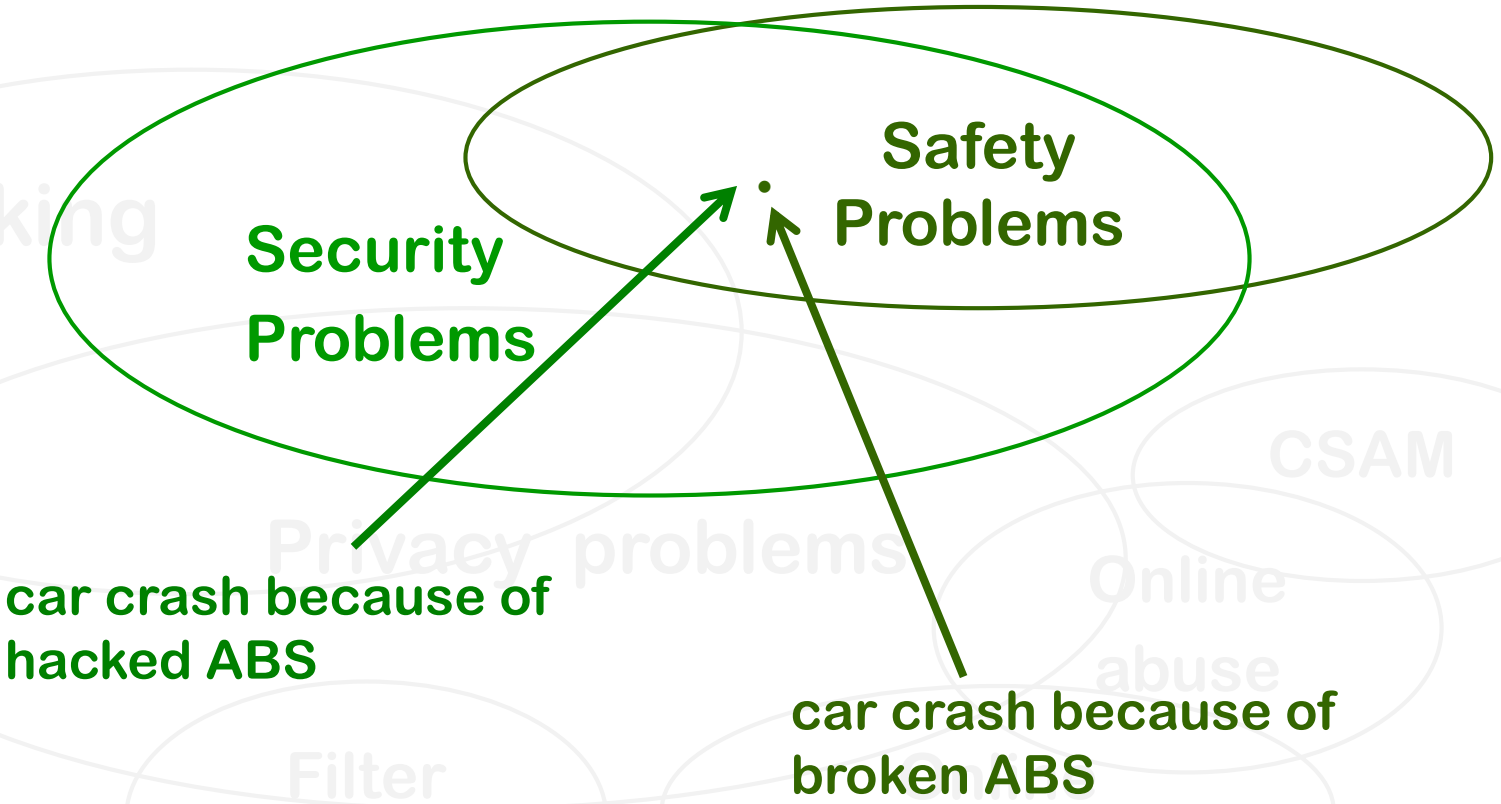
accidents

eg. phone battery catching fire

Beware: this distinction is easily lost in translation!

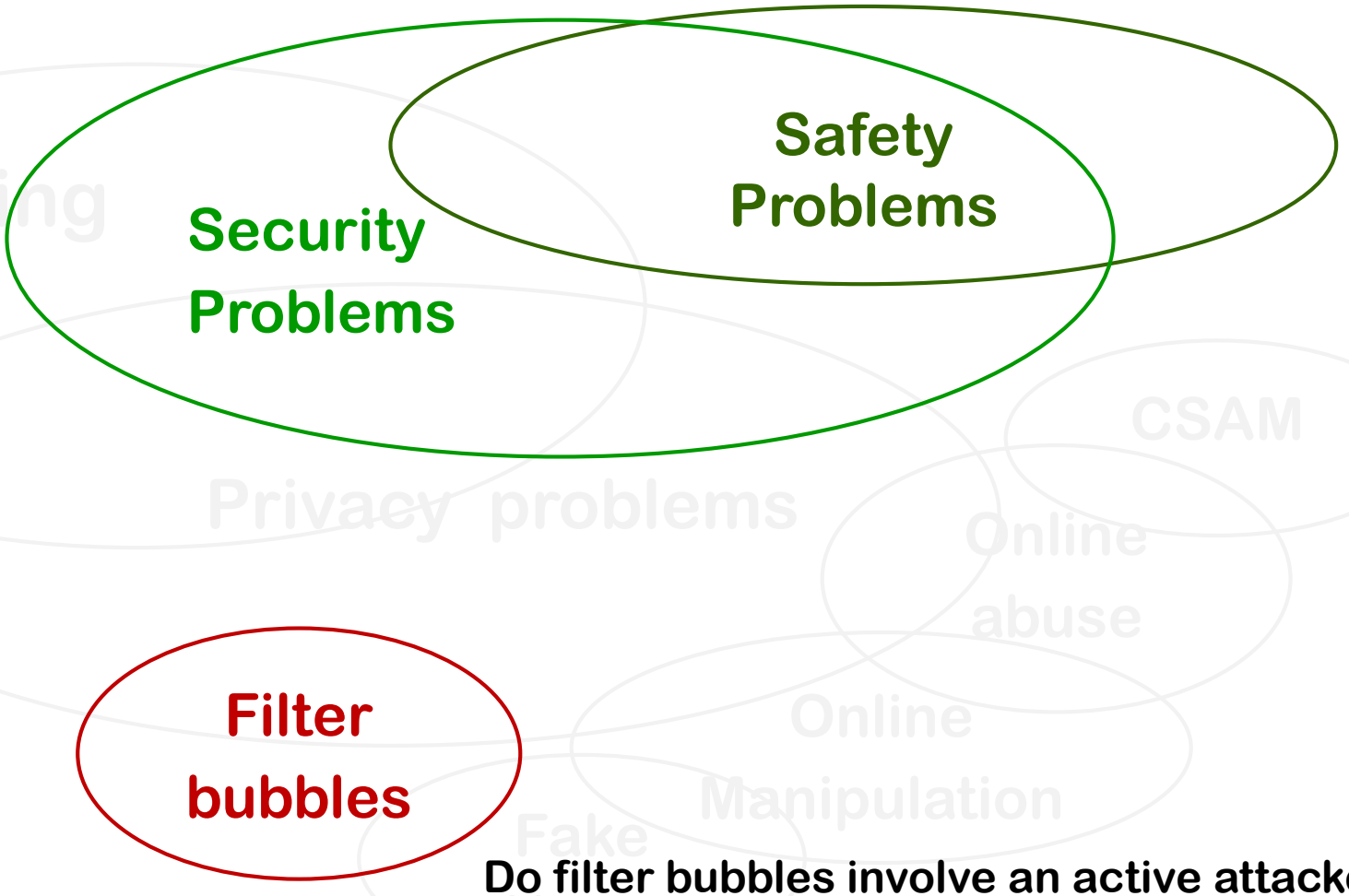
Eg. veiligheid, IT sicherheit, sécurité

Huge overlap, in problems & in countermeasures



But: security harder to get right because
acts of god evolve more slowly than acts of attackers

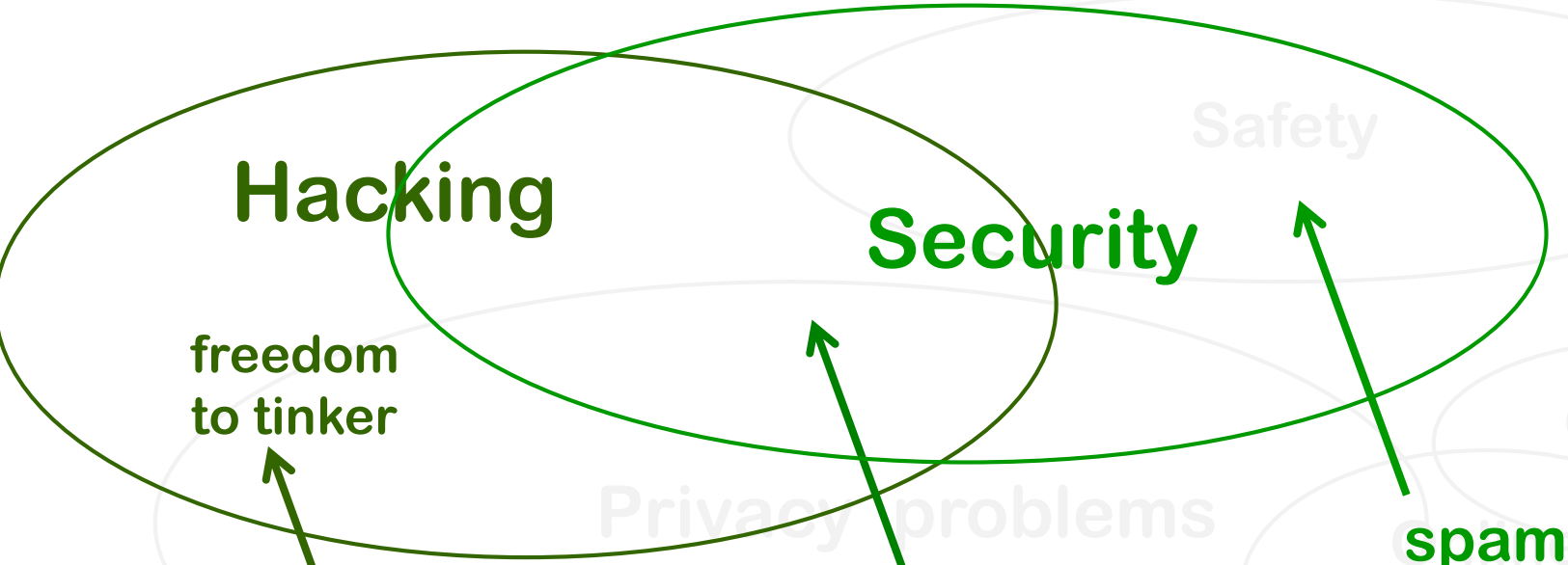
Presence of active attacker?



Do filter bubbles involve an active attacker or are they naturally occurring phenomena?

Security vs Hacking

Security vs Hacking



- Hacking your smart TV to let it run the Netflix app
- Hacking your phone to replace Chrome with Firefox as default browser
- Repairing your smart dishwashing machine

Hacking

using something in a way it was not designed
or intended to be used

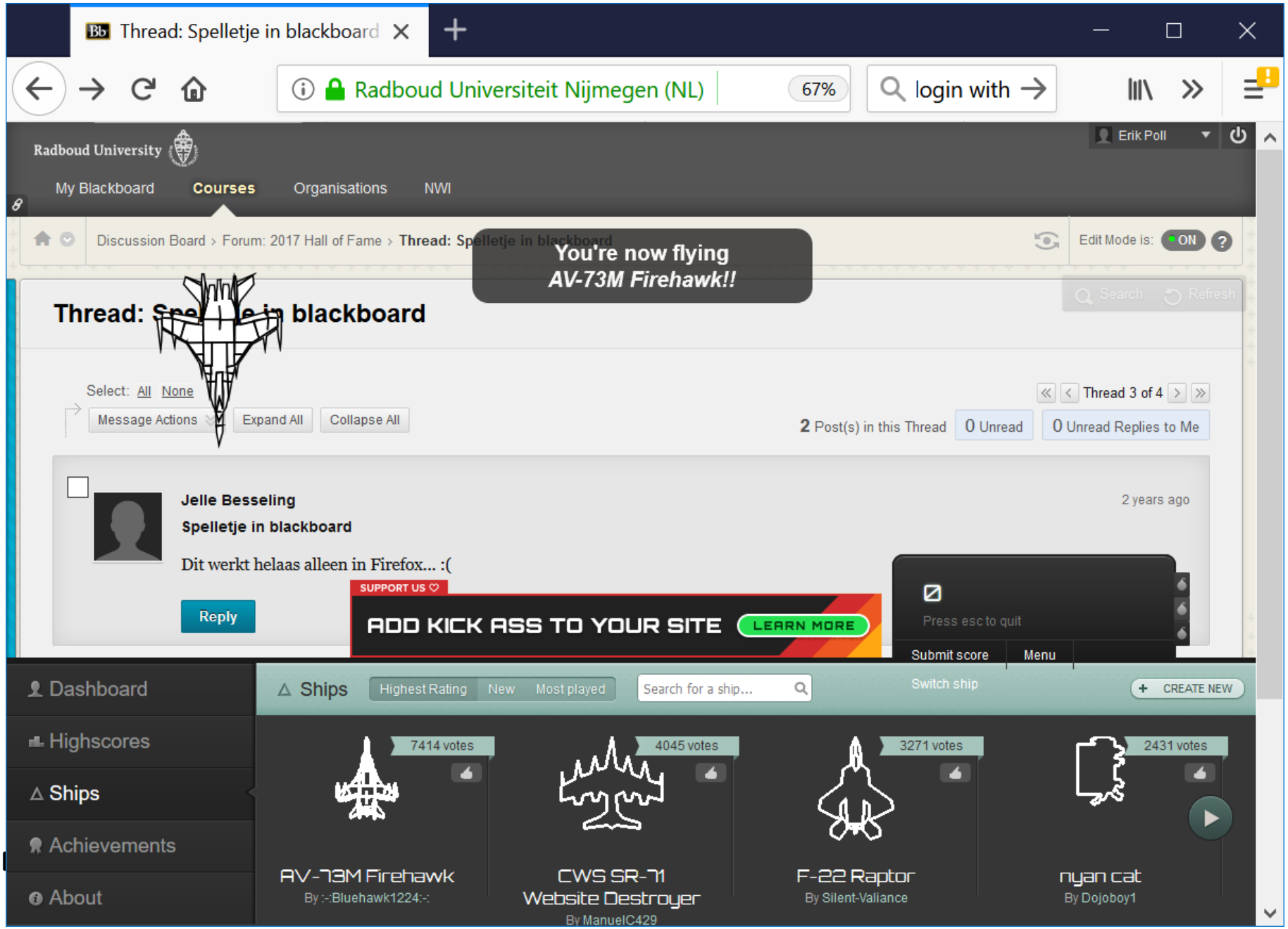
for *good*, or for *bad*

Useful hacking ?

using charge pole to cook waffles
… for free!

[Matthias Dalheimer, CCC'2018, https://evsim.gonium.net]

Harmless hacking? game inside Blackboard



Erik

Non-IT hacks

Hacking does not have to involve IT or software

But: the flexibility of software means that hacking IT systems provides *many* more possibilities

Bad hacking - early example

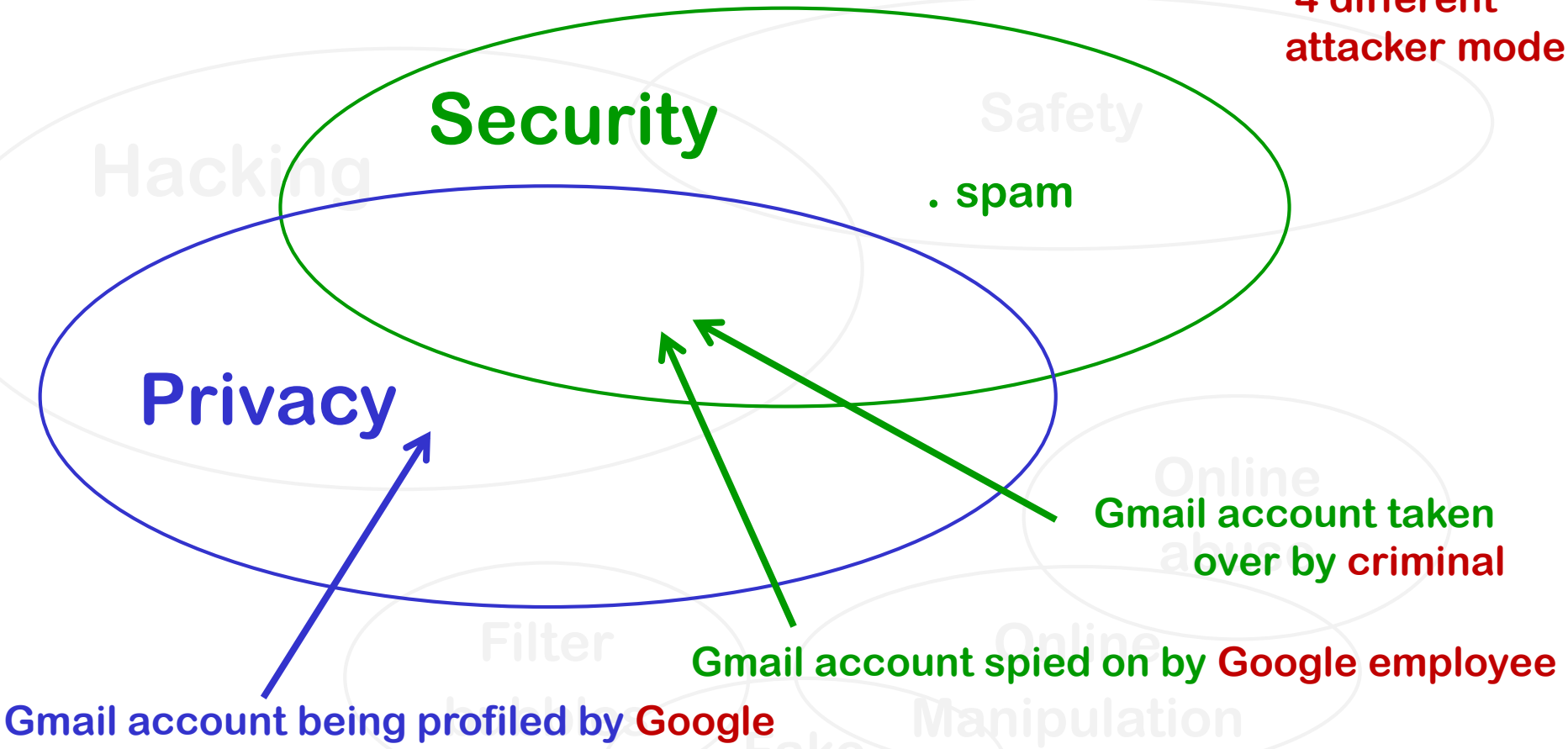
Good hacking - early example

Olduvai chopping stone
1.8 million BCE

Security vs Privacy

Security vs Privacy

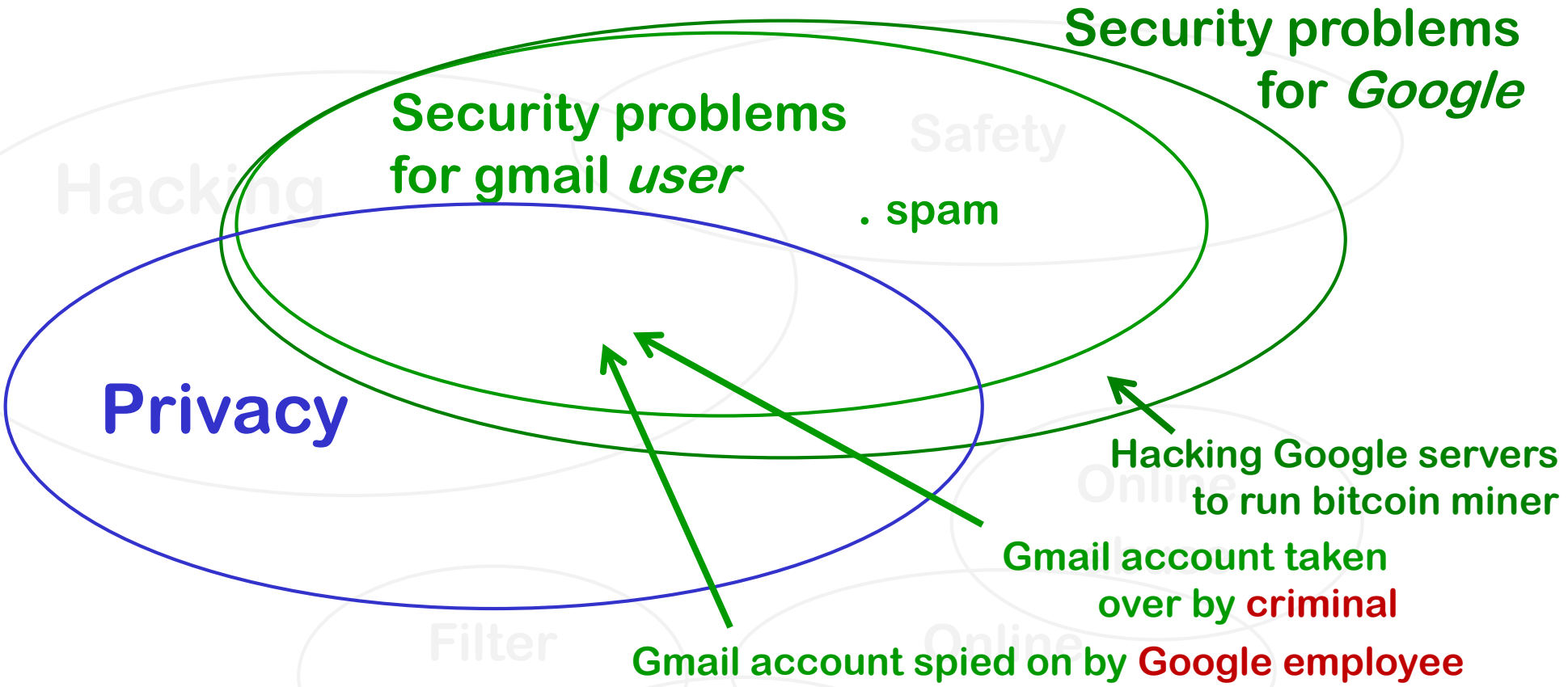
Note the
4 different
attacker models!



The crucial difference between security & privacy is in the attacker model.

For specific threat, not always clear or important if it is security or privacy threat.

Security for *user* vs security for *Google*



Most (all?) security concerns of gmail users are also security concerns of Google

Security example: access control to crisps

Security is about controlling access to resources

We can do this with eg clothes pegs or with duct tape

Which of these is more secure ?

Btw, this security control is a hack!

A

B

A is probably less secure

As often, here security relies on detection, not prevention

Attacker model

- What you mean by 'secure' depends on your attacker model!
  - Does the attacker have scissors & opposable thumbs?
  - Can the attacker reach the top shelf in the kitchen?
  - Does the attacker have sharp claws or teeth?
  - Does the attacker worry about detection?
  - ....

Which of these is more *privacy-friendly*?

A

B

A, as allows access without detection

So as usual, security is bad for privacy! ☹

Which of these is better *to fight obesity*?

A

B

So privacy is bad for tackling societal problems ☹

Privacy, revisited: Which solution is more privacy-friendly?

A

B

Meta-data (crisp flavour) is hidden!

So security can be good for privacy after all! ☺

Security, revisited: Which solution is more secure?

A

B

Less secure if we're worried about easy availability of crisps

So now privacy is bad for security after all?

There are two opposing security concerns:
1) securing unauthorised access vs 2) allowing easy availability

Moral of the story

This one-dimensional view of the world is not correct

less secure **more secure**

Eg.

- **Making backups is good for availability but bad for confidentiality**
- **End-2-end encryption is good for confidentiality but bad for fighting spam and phishing**

Corollary

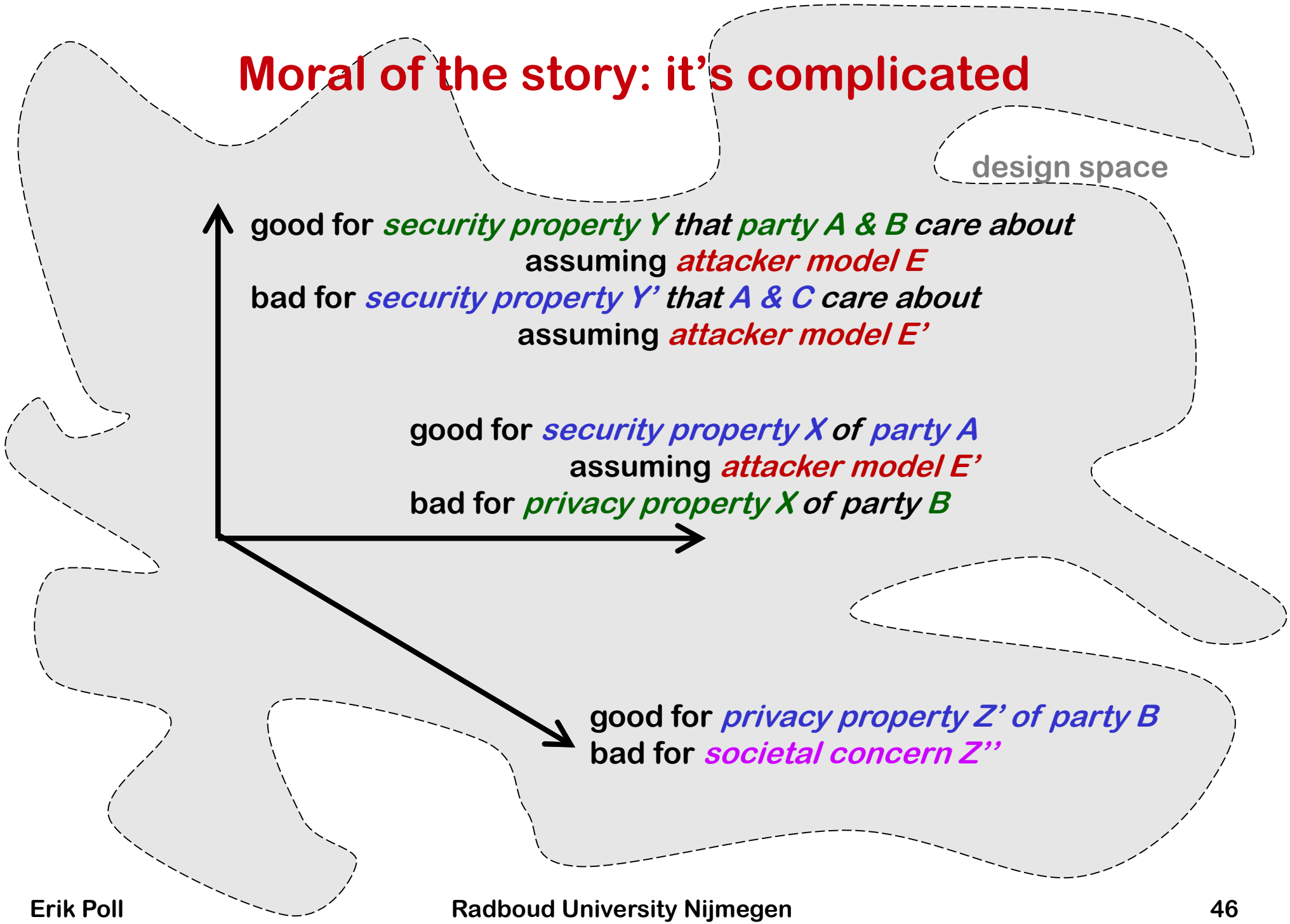
This one-dimensional view of the world is also not correct

more privacy-friendly ⟷ **more secure**

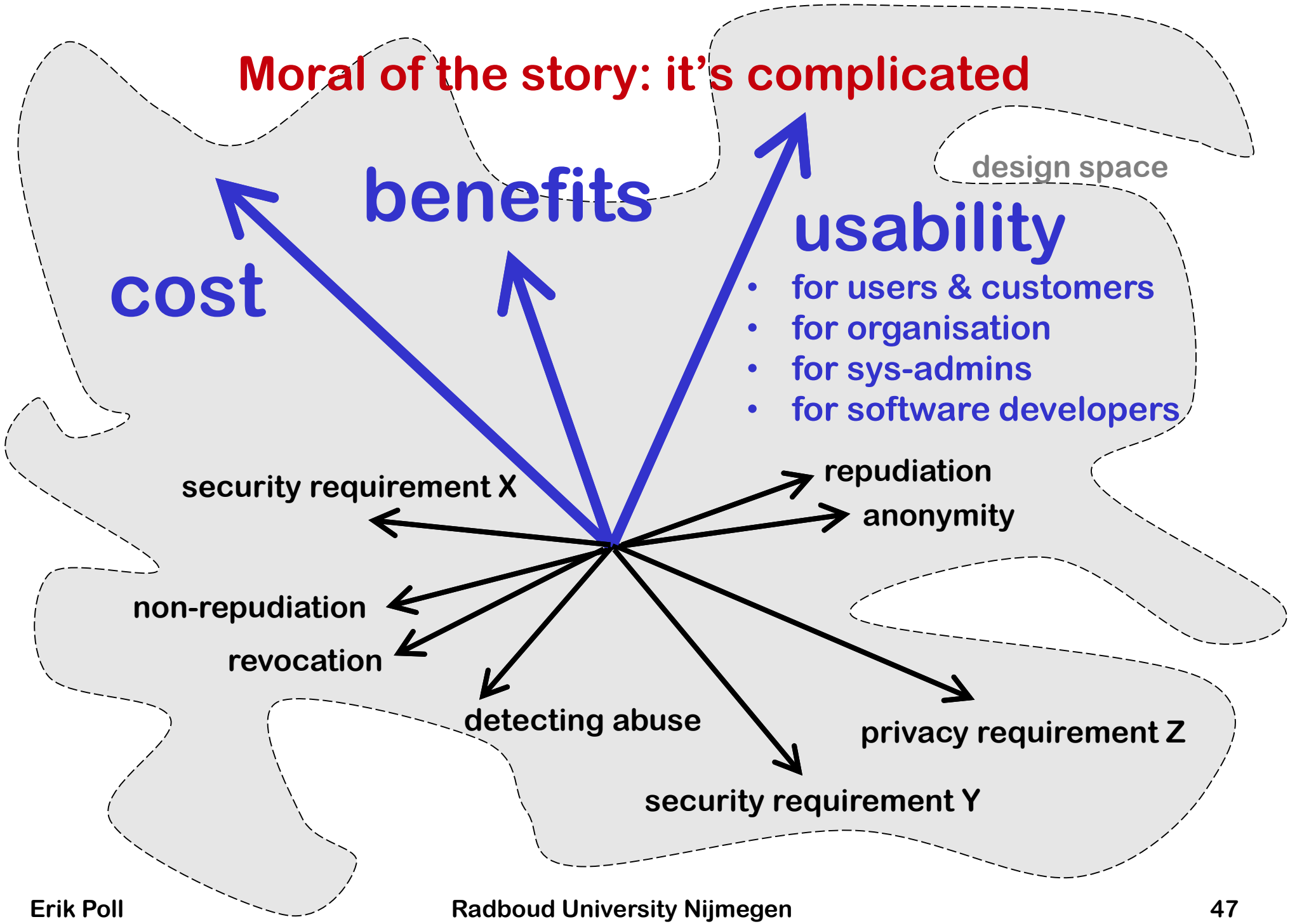
though there can be trade-offs between privacy requirements and specific security requirements (or specific security measures)

esp. when it comes to detection & reaction

Moral of the story: it's complicated



Moral of the story: it's complicated



Conclusions about security, privacy, and hacking

- Hacking is not the *only* source of security problems
  but is the *main* source of security problems
- Hacking *can be* a source of privacy problems
  - e.g. some external attacker hacking Facebook to steal data

  but *is not* the main source of privacy problems
  - Facebook and Facebook's customers pose the bigger risk
- Security is easier than privacy
  because security concerns of various parties - say the platform, its subjects, and society as a whole - tend to be aligned.
  - the platform itself *is* in our attacker model for privacy
    but is not in our attacker model for security

  Still, security externalities can arise & then make security harder

Why am I not in the iHUB?

- As a system designer / system analist
  I do not care if some system requirement is a security requirement or a privacy requirement
  - The distinction may be relevant for my risk assessment, because privacy requirements come with the risk of GDPR fines

- As a software security researcher,
  my goal is to prevent / detect security problems that are *accidentally* introduced
  - Ways to help with this are of no use against people *deliberately* building features into systems that are privacy concerns

Bored tonight?

Talk by former Facebook CISO Alex Stamos

The Platform Challenge: Balancing Safety, Privacy and Freedom

https://www.youtube.com/watch?v=ATmQj787Jcc