Language-based software security

Course summary

Xavier Leroy
2022-04-21

Collège de France, chaire de sciences du logiciel
xavier.leroy@college-de-france.fr
Four attitudes towards computer security

- Pessimistic
- Optimistic
- General
- Specific
Four attitudes towards computer security

- **Pessimistic**: Major threats everywhere! Collapse imminent!
- **Optimistic**: Specific
- **General**: Specific
- **Specific**: Optimistic
Four attitudes towards computer security

- **Pessimistic**: Major threats everywhere! Collapse imminent!
- **Optimistic**: Costs can be tolerated. No action required.
- **General**
- **Specific**
Four attitudes towards computer security

- **Optimistic**
  - Costs can be tolerated.
  - No action required.

- **Pessimistic**
  - Major threats everywhere!
  - Collapse imminent!

- **General**
  - Long list of attacks and vulnerabilities

- **Specific**
Four attitudes towards computer security

- Pessimistic:
  - Major threats everywhere!
  - Collapse imminent!
- Optimistic:
  - Costs can be tolerated.
  - No action required.
- General:
  - Long list of attacks and vulnerabilities
- Specific:
  - Short list of protections and counter-measures
The components of computer security

Users

Regulations  Organizations  Economy

Networks  Web  Cloud

System software  Application software

Hardware
The components of computer security

Users

Regulations   Organizations   Economy

Networks   Web   Cloud

System software   Application software

Hardware
The components of computer security

Users
Regulations
Organizations
Economy
Networks
Web
Cloud
System software
Application software
Hardware
What role for programming languages and tools?

An essential role:

• Run-time safety, to guarantee the integrity of data structures and control flow.

Some specific contributions to security, such as

• Controlling information flow (lecture 2)
• Software fault isolation (lecture 3)
• “Constant-time” programming (lecture 4)
• Mobile code verification; proof-carrying code (lecture 5)
• Protections against microarchitectural attacks (lecture 6)
Security: a challenge for programming

Poor programming practices that make it harder to write secure software.

- Performance does not trump security.
- Testing does not suffice to avoid vulnerabilities.

*Programming Satan’s computer*
(Ross Anderson)

It is difficult to formally reason beyond functional correctness:

- confidentiality, privacy;
- availability, resilience;
- hardware faults and information leaks.
FIN