Security (I)
Lecture 25, cs262a

Ion Stoica & Ali Ghodsi
UC Berkeley
November 30, 2020

Today's Paper

Making Information Flow Explicit in HiStar,

Nickolai Zeldovich, Silas Boyd-Wickizer, Eddie Kohler, and David Mazières

https://people.csail.mit.edu/nickolai/papers/zeldovich-histar.pdf

Using Crash Hoare Logic for Certifying the FSCQ File System,

Haogang Chen, Daniel Ziegler, Tej Chajed, Adam Chlipala, M. Frans Kaashoek, and Nickolai Zeldovich

https://people.csail.mit.edu/nickolai/papers/chen-fscq.pdf

What is the problem?

Programs have bugs

Bugs lead to:

- Security vulnerabilities
  - E.g., buffer overflows, format string issues, SQL injection, JS injection, temp file races, integer overflows
- Lost data
  - E.g., data not flushed to disk from OS buffer before failure

Two papers

How do you avoid information leakage?

- How do you ensure that only users and applications that have the right to see the data see that data an no one else?

How do you make sure that a failure/bug doesn’t cause data loss on a file system

Two solutions

Minimize trust domain: design the system so that you need to trust just a small part of code, and then carefully write it to avoid bugs

- Challenge: how do you know there are no bugs?

Software verification: use formal methods to verify your program is "correct"

- Challenge: can you verify realistic programs?

Software verification

Proof assistant: Prove the implementation meets the specification

- E.g., Coq 

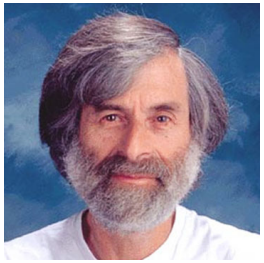
Model checker: Enumerates the state space of the specification and checks whether any of the verification conditions is violated (mostly used for distributed protocols)

- E.g., TLA+ 

TLA+

Developed by Leslie Lamport

- Turing Award 2013



AWS using TLA+ for some production services: https://lamport.azurewebsites.net/tla/formal-methods-amazon.pdf