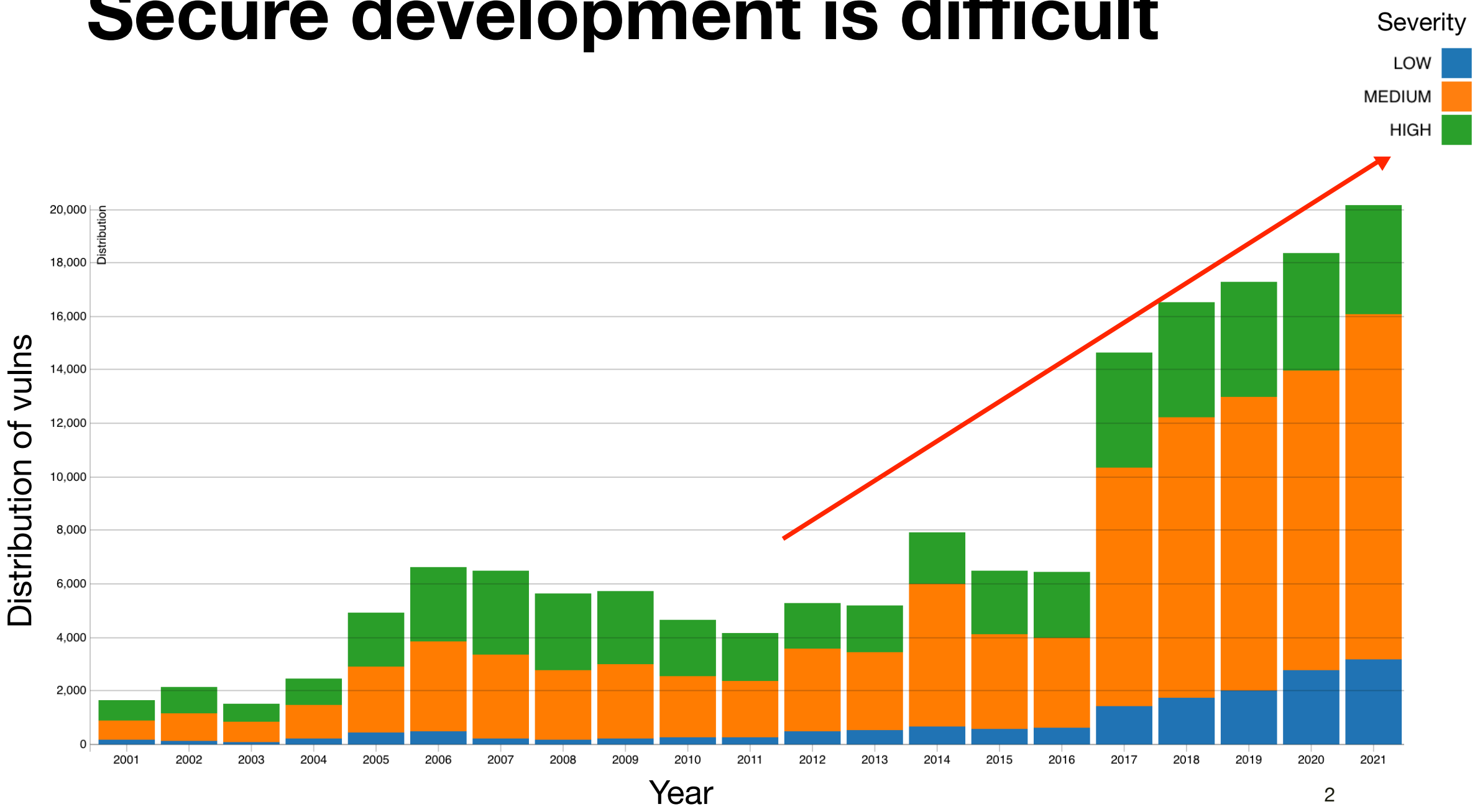
UNDERSTANDING THE HOW AND THE WHY: EXPLORING SECURE DEVELOPMENT PRACTICES THROUGH A COURSE COMPETITION

Kelsey Fulton, Daniel Votipka, Desiree Abrokwa,
Michelle Mazurek, Michael Hicks, and James Parker

MARYLAND CYBERSECURITY CENTER | MC2

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

Secure development is difficult



Many solutions exist!

- Secure tools
- Secure training/ education
- Better integration of security

How do we help companies prioritize solutions?

We need to understand why and how different types of vulnerabilities get introduced, found, and fixed?

We need to understand why and how different types of vulnerabilities get introduced, found, and fixed?

We need to understand why and how different types of vulnerabilities get introduced, found, and fixed?

Prior work

- Analyzed code submitted to BIBIFI

Build it, Break it, Fix it (BIBIFI)

Secure programming competition:

- **Build-it**
  - Build to a secure spec with open choices
  - Earn points for performance and functionality
- **Break-it**
  - Other teams' code made available
  - Submit exploits against other teams
- **Fix-it**
  - Update their code to fix submitted vulnerabilities

[Ruef et al., 2016]

Prior work

- Analyzed code submitted to BIBIFI
  - 94 projects
- Building in-depth taxonomy of vulnerabilities
- Uncovering characteristics of vulnerabilities
- Unable to uncover

Prior work

- Analyzed code submitted to BIBIFI
- Building in-depth taxonomy of vulnerabilities
- Uncovering characteristics of vulnerabilities
- Unable to uncover **how**

Prior work

- Analyzed code submitted to BIBIFI
- Building in-depth taxonomy of vulnerabilities
- Uncovering characteristics of vulnerabilities
- Unable to uncover **how** and **why**

Research questions

- What type of vulnerabilities do developers introduce and why?
- What types of vulnerabilities are found during review and why?
- Why and how do developers fix different types of vulnerabilities?

Methods

- Used BIBIFI in 3 week long course
  - Spent approx. 1 week in each phase
- 14 teams composed of undergrads
  - Juniors/seniors
  - Participants had minimal security/development experience
- Participants were not expected to have prior security exp
  - Took core systems course
  - Had short lectures on security and threat modeling

Methods

- Collected fine-grained data
  - Design documents (multiple times)
  - Snapshots of code as they developed
  - Submitted exploits and fixes
  - Commit messages throughout build, break, fix

Methods

- Collected fine-grained data
  - Design documents (multiple times)
  - Snapshots of code as they developed
  - Submitted exploits and fixes
  - Commit messages throughout build, break, fix

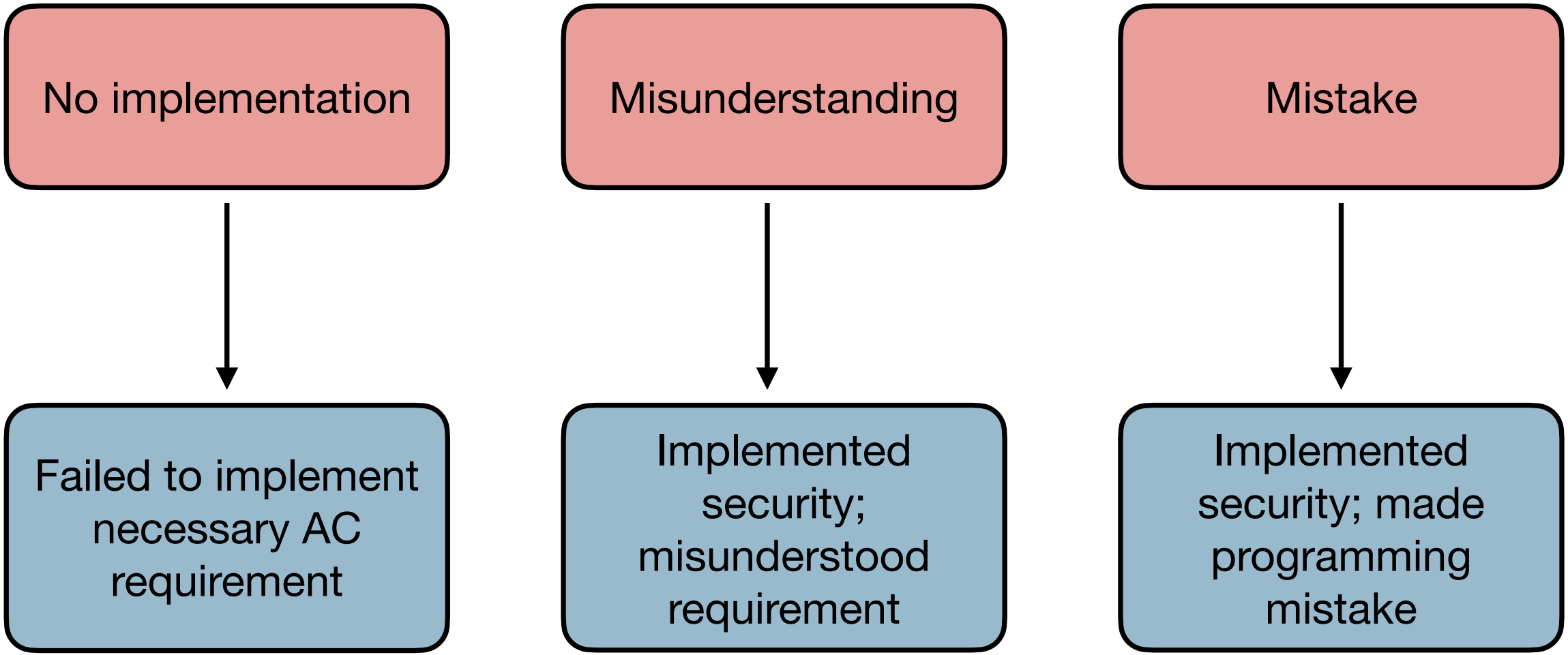
Methods

- Collected fine-grained data
  - Design documents (multiple times)
  - Snapshots of code as they developed
  - Submitted exploits and fixes
  - Commit messages throughout build, break, fix
- Analyzed data using:
  - Manual code review for vulnerabilities
  - Qualitative coding

IoT smart home

- Runs user scripts
- All data protected by RBAC
  - Customizable by special users and data owner
  - Other users receive permissions

Overview of vulnerabilities



Research questions

- What type of vulnerabilities do developers introduce and why?
- What types of vulnerabilities are found during review and why?
- Why and how do developers fix different types of vulnerabilities?

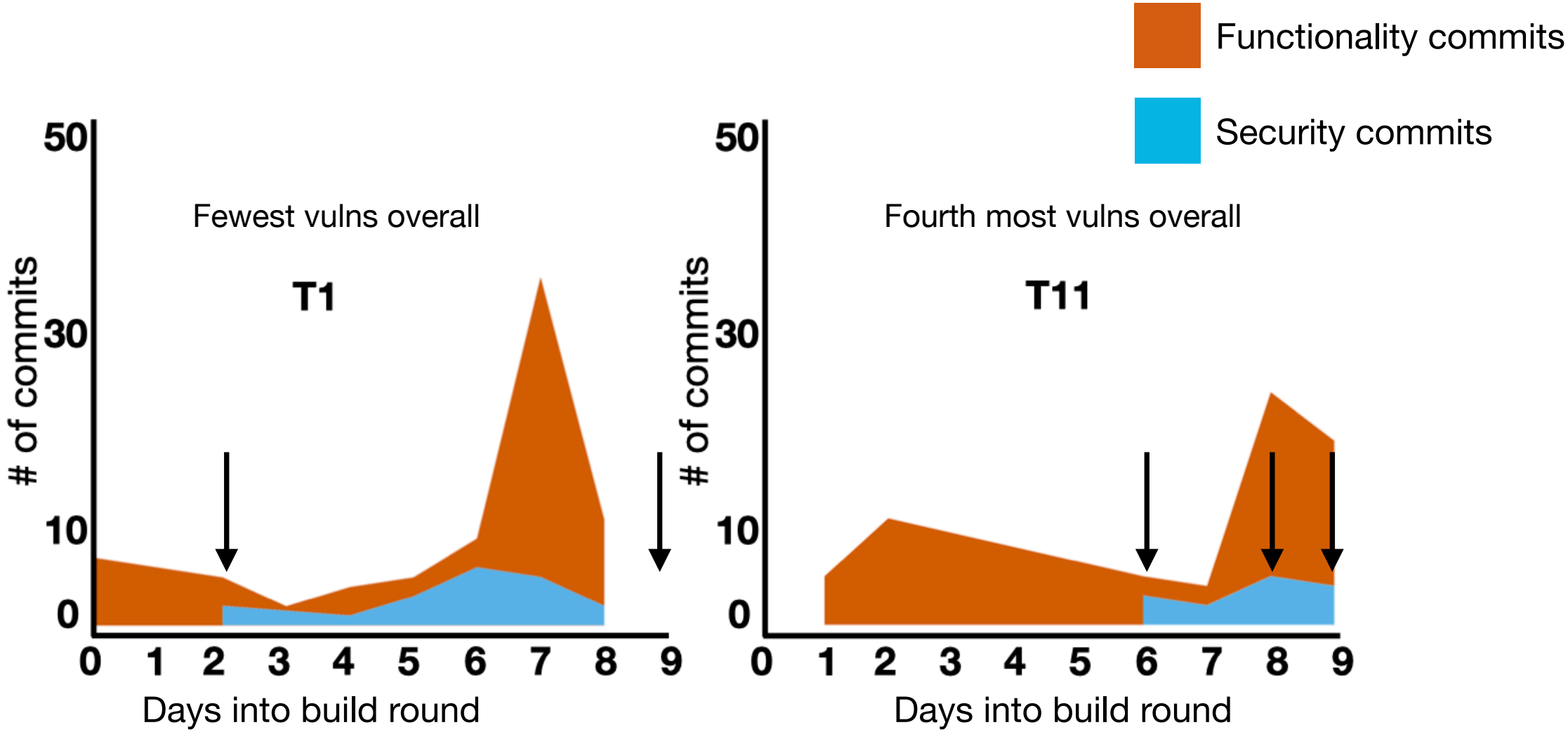
Impact of design on security

- Teams with detailed initial designs tended to introduce fewer vulnerabilities
  - Specifically No Implementation and Mistakes
- Teams with detailed initial designs tended to stick with them
  - Even if the initial design had a vulnerability
  - Especially prevalent with Misunderstandings

Impact of timeline on security

- Teams with fewest vulnerabilities tended to work on security throughout
- Teams that waited to work on security ran out of time
  - Resulting in many vulnerabilities

Impact of timeline on security



Research questions

- What type of vulnerabilities do developers introduce and why?
- What types of vulnerabilities are found during review and why?
- Why and how do developers fix different types of vulnerabilities?

Different vulnerabilities are discovered differently

- No Implementations found when looking broadly for related problem
  - Found one access control vulnerability while attempting to exploit another
- Mistakes were found through broad testing
  - Emulating the use of a fuzzer
- Misunderstandings required targeted testing
  - Many left unexploited

Research questions

- What type of vulnerabilities do developers introduce and why?
- What types of vulnerabilities are found during review and why?
- Why and how do developers fix different types of vulnerabilities?

Different vulnerabilities are fixed differently

- Half of No Implementations left unfixed at end of study
  - Rearchitecting whole system
- Misunderstandings were rarely fixed until exploited
  - But were overwhelmingly fixed once exploited

Implications

- Vulnerabilities differ in more than content
  - No “one size fits all” solution
- Importance of best practices
  - Incremental development
  - Detailed design
- Including security experts at beginning of development cycle

Questions:
kfulton@umd.edu
https://www.cs.umd.edu/~kfulton/

I am on the job market this year!