IEEE CYBERSECURITY DEVELOPMENT CONFERENCE 2018

TRIP REPORT

Isaac Richter

http://cambridge.hyatt.com/en/hotel/home.html

TUTORIALS/WORKSHOPS

- **Principles and Practices of Secure Coding**
  Sazzadur Rahaman, Na Meng, Daphne Yao (Virginia Tech)
- **Secure Coding Practices, Automated Assessment Tools and the SWAMP**
  Barton P. Miller and Elisa Heymann (UW Madison)
- **Continuous Verification of Critical Software**
  Mike Dodds, Stephen Magill, Aaron Tomb (Galois, Inc.)
- **DeepState: Bringing Vulnerability Detection Tools into the Development Cycle**
  Peter Goodman, Gustavo Grieco (Trail of Bits, Inc.), Alex Groce (Northern Arizona University)

TUTORIALS/WORKSHOPS

- **Secure Your Things: Secure Development of IoT Software with Frama-C**

  Allan Blanchard (Inria Lille – Nord Europe, France), Nikolai Kosmatov (CEA, Software Reliability and Security Lab, France), Frédéric Loulergue (Northern Arizona University)

- **Building Secure Consortium Blockchains for Decentralized Applications**

  Chengjun Cai, Huayi Duan, and Cong Wang (City University of Hong Kong)

- **Parry and RIPOSTE: Honing Cybersecurity Skills with Challenge-Based Exercises**

  Jan Werner (University of North Carolina at Chapel Hill), Fabian Monrose (UNC Chapel Hill)

KEYNOTES

- **Building and Deploying Secure Systems in Practice: Lessons, Challenges and Future Directions**

  Dawn Song, UC Berkeley

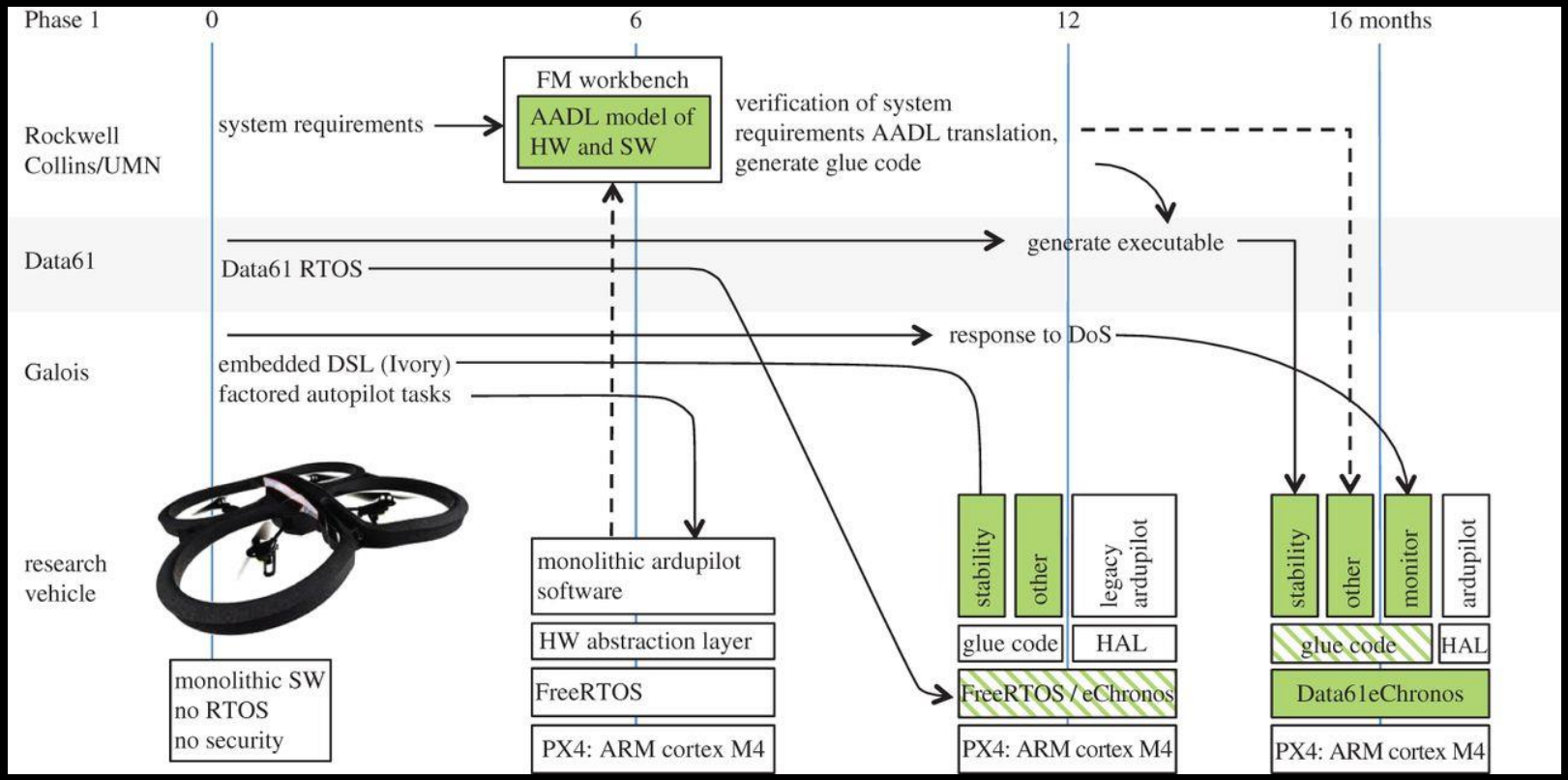
  - Automating Security
  - Privacy-preserving analytics

- **Provably Eliminating Exploitable Bugs**

  Kathleen Fisher, Tufts University (Former Program Manager of DARPA's HACMS Program)

  - Formal Verification
  - Provably-correct code

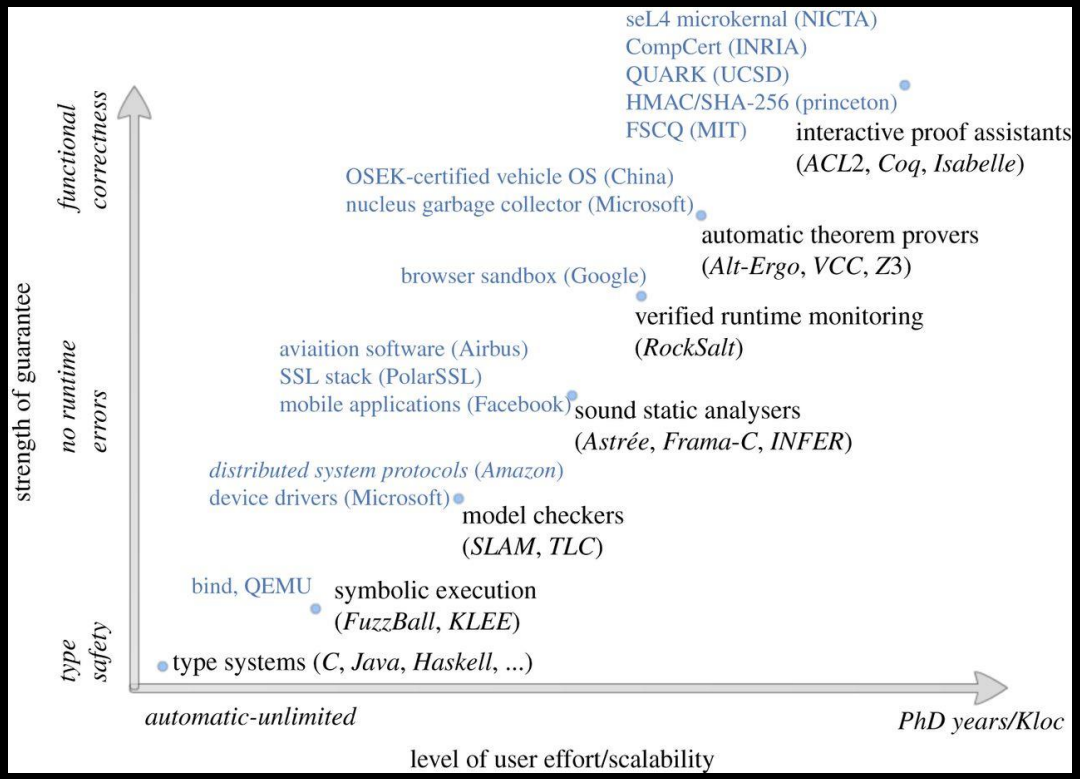
PROVABLY ELIMINATING EXPLOITABLE BUGS



Phase 1 architecture of the SMACCMCopter. Green boxes denote high-assurance components.

Kathleen Fisher et al. Phil. Trans. R. Soc. A 2017;375:20150401

PROVABLY ELIMINATING EXPLOITABLE BUGS



Formal-method tools. Tool classes and example tools are to the right of the plotted points. Example systems analysed using a particular type of tool are on the left.

Kathleen Fisher et al. Phil. Trans. R. Soc. A 2017;375:20150401

POSTERS

- **A Test Infrastructure for Self-Adaptive Software Systems**

  E. Kilmer, T. Braje, D. Doyle, T. Meunier, P. Zucker, J. Hughes, M. Depot, M. Mazumder, G. Baah, K. Chadha, R. Cunningham (MIT Lincoln Labs)

- **Automating Threat Intelligence for SDL**

  R. Kannavara, M. Lindholm, P. Shrivastav (Intel), J. Vangore, W. Roberts (Olivet Nazarene University)

- **Trapping Spectres in Speculation Domains**

  I. Richter, Y. Du, J. Criswell (University of Rochester)

- **Transforming Code to Drop Dead Privileges**

  X. Hu (BitFusion.io), J. Zhou, S. Gravani, J. Criswell (University of Rochester)

- **Diversity for Software Resilience**

  A. Gearhart (Johns Hopkins University Applied Physics Laboratory)

POSTERS

- **Data Integrity**

  T. McBride, (NIST), A. Townsend, M. Ekstrom, L. Lusty, J. Sexton (MITRE)

- **Extracting Anti-specifications from Vulnerabilities for Program Hardening**

  M. Ahmed, D. Yao (Virginia Tech) H. Cai (Washington State University)

- **Automatic Patch Generation for Security Functional Vulnerabilities with GAN**

  Y. Xiao, D. Yao (Virginia Tech)

- **Toward Secure and Serverless Trigger-Action Platforms**

  P Datta (UIUC), T. Morris, H. Vijayakumar, M. Grace (Samsung Research), A. Bates (UIUC), A. Rahmati, (Samsung Research, SUNY Stony Brook)

POSTERS

- **Automatic Detection of Confused-Deputy Attacks on ARM TrustZone Environments**

  J. Budenske, A. Budenske (Cyberific Secure Autonomous Systems)

- **Command, Control and Coordination of Moving Target Defenses**

  M Carvalho (Florida Institute of Technology)

- **Moving Target Defenses and Cyber Resiliency**

  R. McQuaid, D. Bodeau, R. Graubart (MITRE)

BEST PRACTICES

- **Formal Proofs, the Fine Print and Side Effects**

  Toby Murray (University of Melbourne) and Paul van Oorschot (Carleton University)

- **Integrating Cyber Vulnerability Assessments Earlier into the Systems Development Lifecycle**

  Sonja Glumich, Juanita Riley, Paul Ratazzi, and Amanda Ozanam (Air Force Research Laboratory Information Directorate)

- **DECREE: A Platform and Benchmark Corpus for Repeatable and Reproducible Security Experiments**

  Lok Yan (Air Force Research Laboratory), Benjamin Price (MIT Lincoln Laboratory), Michael Zhivich (MIT Lincoln Laboratory), Brian Caswell (Lunge Technology), Christopher Eagle (Naval Postgraduate School), Michael Frantzen (Kudu Dynamics), Holt Sorenson (Google Inc.), Michael Thompson (Naval Postgraduate School), Timothy Vidas (Carnegie Mellon University), Jason Wright (Thought Networks), Vernon Rivet (MIT Lincoln Laboratory), Samuel Colt VanWinkle (MIT Lincoln Laboratory), and Clark Wood (MIT Lincoln Laboratory)

- **Profiling Vulnerabilities on the Attack Surface**

  Toby Murray (University of Melbourne) and Paul van Oorschot (Carleton University)

Formal Proofs, the Fine Print and Side Effects

Toby Murray and Paul C. van Oorschot

Formal Proofs: Successes

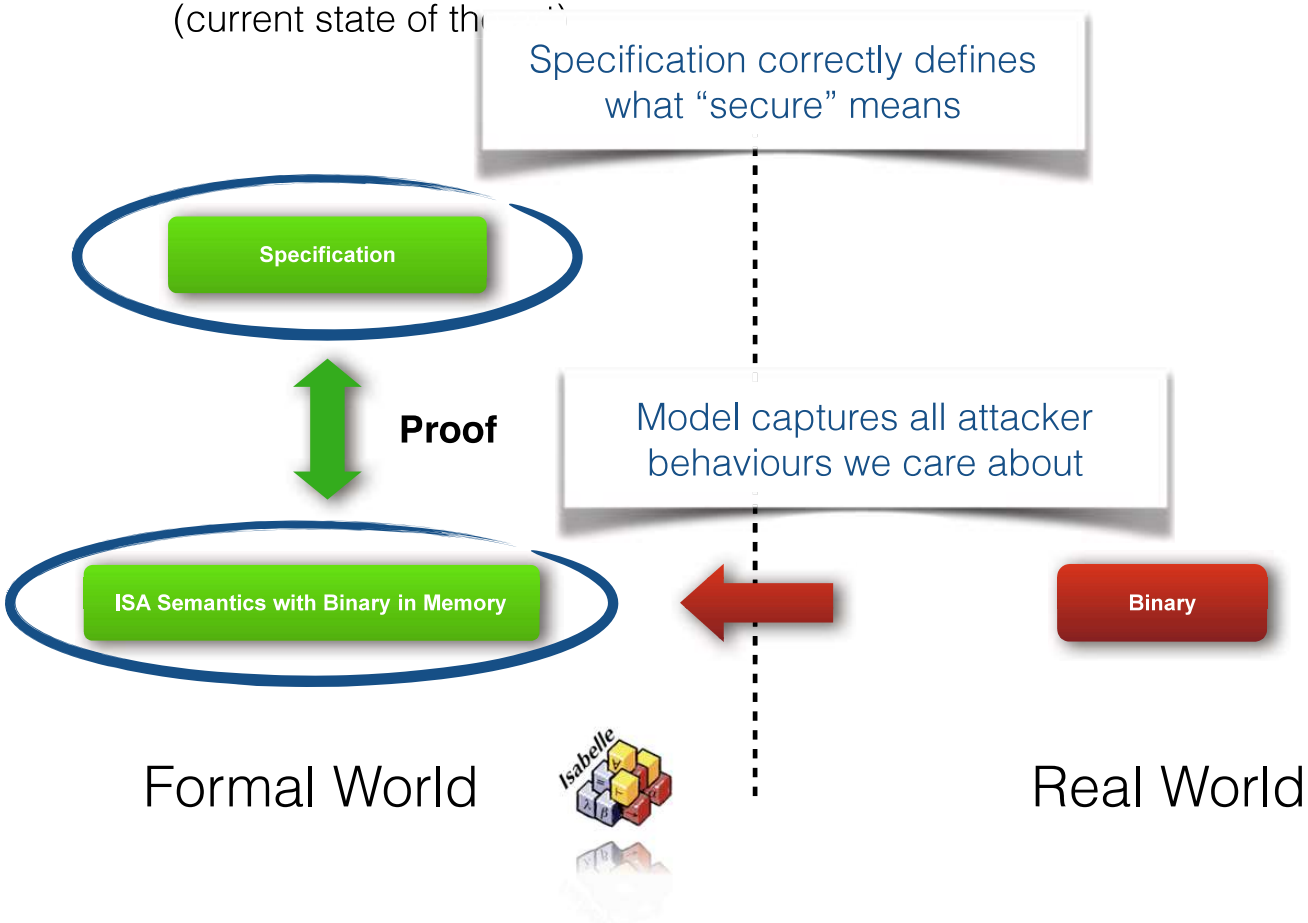
Formally verified
OS microkernel

Project Everest

Formally verified crypto
libraries

What is a high-fidelity proof?

(current state of th



Example

A claim we might want to prove:

The code below when run on modern x86 CPUs, can cause modifications only within those data cache sets that can be occupied by the physical memory corresponding to the program variables `i`, `r` and the array `a`, whose length is `ARRAY_LEN`.

```
if (i < ARRAY_LEN){
  r = a[i];         }
```

Is it true? No (e.g. Spectre)

But it can be proved if the ISA model doesn't include speculative execution

Example

A claim we might want to prove:

The code below when run on modern x86 CPUs, can cause modifications only within those data cache sets that can be occupied by the physical memory corresponding to the program variables `i`, `r` and the array `a`, whose length is `ARRAY_LEN`.

```
if (i < ARRAY_LEN){
  r = a[i];         }
```

Is it true? No (e.g. Spectre)

But it can be proved if the ISA model doesn't include speculative execution

P1: Proofs as Qualified Guarantees

Security-related **guarantees** are: claims of invulnerability to specific attack(s), qualified by lists of assumptions

non-experts (almost everyone) misinterpret these as absolute guarantees against **all** possible attacks

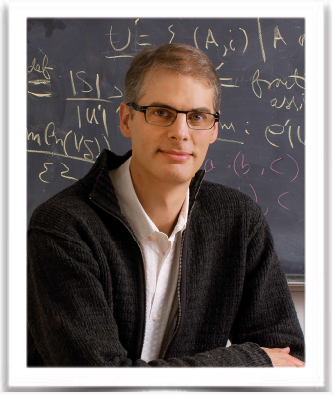
(what else should we expect if we use language like “proven secure”)

A proof provides guarantees
subject to the accuracy of
the proof’s assumptions and model …

Understanding Guarantees

Proofs cannot make guarantees above whatever formal statement you proved

How difficult are they to understand, even for experts?



Benjamin Pierce (UPenn)

2015: studied seL4 proofs to teach them as part of a graduate class.

Took **one person week** to understand the **top level** statements well enough to produce 2 lectures

Fine Print

Are proof statements and assumptions akin to fine print on insurance contracts?

Never read by those who most need to read them

“User”’s responsibility to ensure assumptions match reality

Non-experts in no position to validate assumptions and model

Often not written down but **buried deep in formal models**

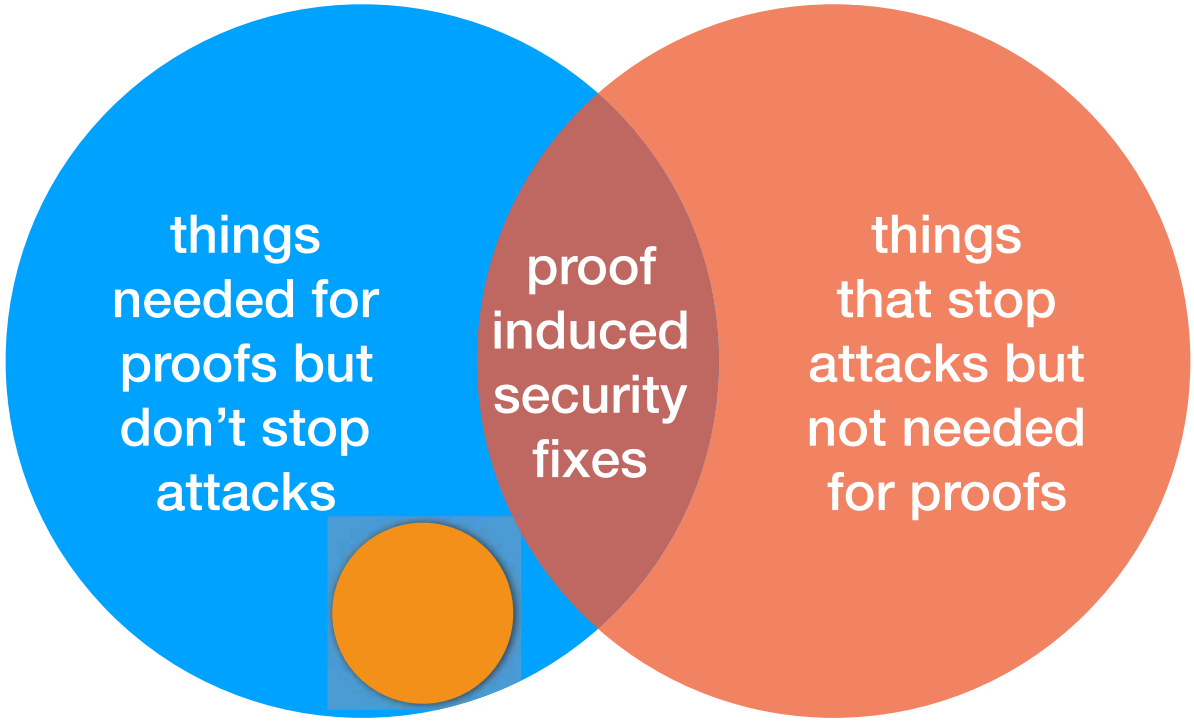
(e.g. subtleties of modelling UNPREDICTABLE behaviour in ARM ISA)

Language

- Laypeople think “proof” means 100% guarantee of security
- But software proofs have **so many** assumptions
  - This makes them quite different to theorems people learned in high school (e.g. Pythagoras)

- Rhetorical: Is this why you’ve never heard a civil engineer say that they “proved” or “formally verified” that a bridge won’t fall down?
  - Civil engineers don’t generally have adaptive attackers trying to break their models, yet still avoid this language
  - Should FM-security folks (me!) use different language?
  - “in particular, it would help if they did not call their verifications ‘proofs.’”
    - De Millo, Lipton and Perlis, *CACM* 1979

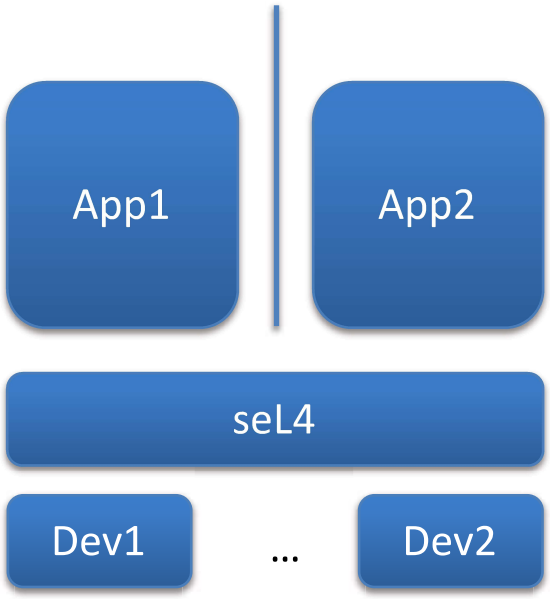
Proof Side-Effects



changes that introduce *new* vulnerabilities, unseen by proof

Deployment Restriction: Example

seL4 confidentiality proofs require device interrupts be disabled



Suppose we care only about isolating **memory** (not interrupts)

Do the seL4 confidentiality proofs help?

Some Research Questions

Q1: Can we find means to **know and measure the relationship between proof side effects and changes that stop attacks**, how these sets intersect, and the intersection sizes?

Q2: Can we find means to **measure the residual value of proofs**, when not all assumptions hold in practice; can we presently even begin to attempt such a measurement?

Q3: How can we better tag formally verified software to **explain the fine print that accompanies the proofs**?

Q4: What effort can be undertaken to explore formal or other methods to **track and validate that** (both implicit and explicit) **security assumptions** in large-scale formal models **hold in practice?**

DATA ACCESS SECURITY

- **Tyche: A Risk-Based Permission Model for Smart Homes**

  Amir Rahmati (Samsung Research America/Stony Brook University), Earlence Fernandes (University of Washington), Kevin Eykholt (University of Michigan), and Atul Prakash (University of Michigan)

- **Detecting leaks of sensitive data due to stale reads**

  Will Snavely, William Klieber, Ryan Steele, David Svoboda, and Andrew Kotov (Software Engineering Institute – Carnegie Mellon University)

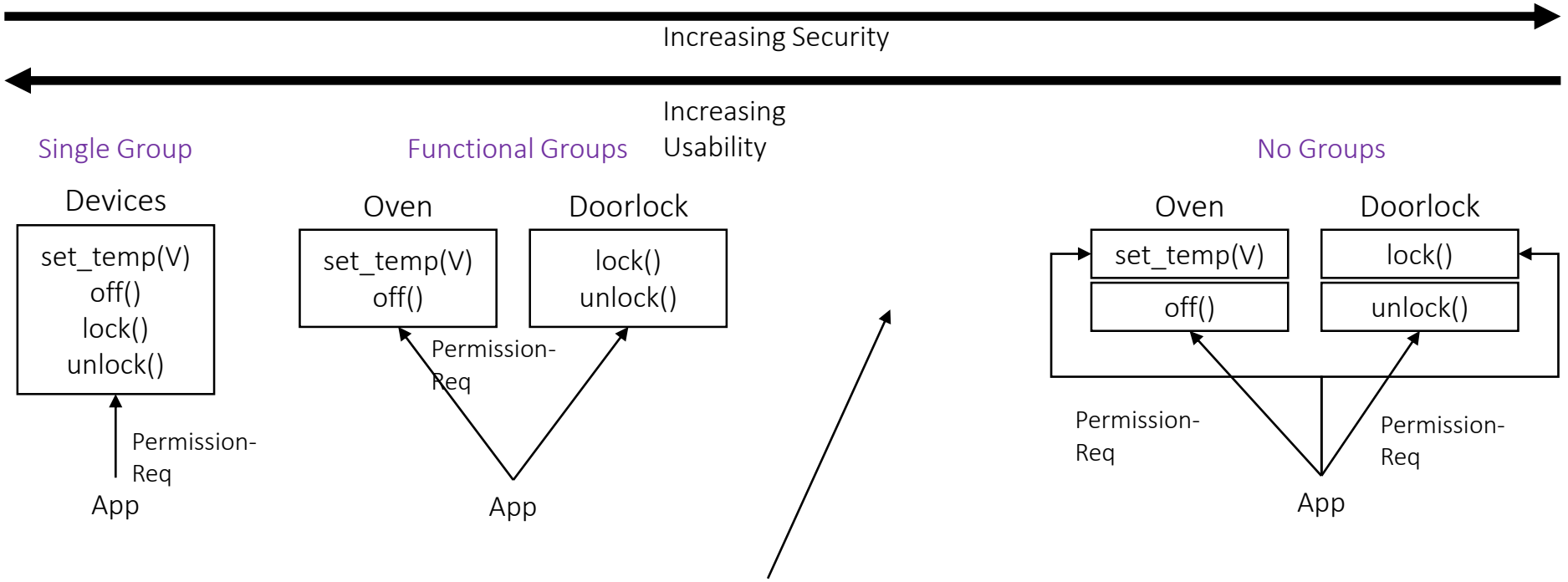
- **Transforming Code to Drop Dead Privileges**

  Xiaoyu Hu (BitFusion.io Inc.), Jie Zhou, Spyridoula Gravani, and John Criswell (University of Rochester)

Tyche: A Risk-Based Permission Model for Smart Homes

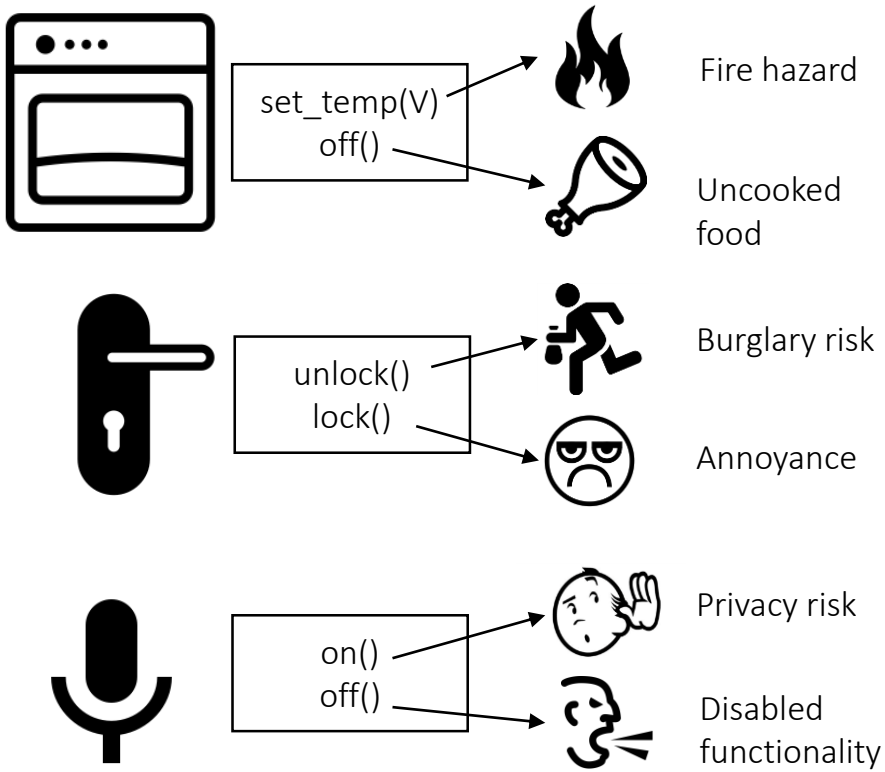
Amir Rahmati, Earlence Fernandes, Kevin Eykholt, Atul Prakash

Permission Model Spectrum



Is there a secure and usable middle-ground in the context of smart homes?

Intuitive Risk Asymmetry in Device Operations



- How do we “measure” risk?
- How can we group operations in terms of risk, and then enforce it?
- Does risk-grouping strike a reasonable balance in security and usability?

Tyche: Risk-Based Permissions for Smart Homes

- Like any other software system, permission models for smart homes lie on a spectrum
  - The point chosen in the spectrum can lead to attacks, or to poor usability
- Physical devices exhibit **intuitive risk asymmetry**
  - We can measure risk using a study with domain experts
  - Risk perceptions of domain **experts co-related with informed end-users**
- We analyzed 3 apps using the risk-based model, and show that they can remain functional, with 60% less access to high-risk operations
- We have laid a foundation and design process for future smart home permission models

Earlence Fernandes, earlence.com, earlence@cs.washington.edu

SECURE CODING AND ANALYSIS

- **Checked C: Making C Safe by Extension**

  Archibald Samuel Elliott (University of Washington), Andrew Ruef, Michael Hicks (University of Maryland), and David Tarditi (Microsoft Research)

- **SGL: A domain-specific language for large-scale analysis of open-source code**

  Darius Foo, Ang Ming Yi, Jason Yeo, and Asankhaya Sharma (SourceClear)

- **Light-touch Interventions to Improve Software Development Security**

  Charles Weir, Lynne Blair (Lancaster University), Ingolf Becker, Angela Sasse (University College London), and James Noble (Victoria University of Wellington)

- **A Lingua Franca for Security by Design**

  Alexander van den Berghe (imec-DistriNet, KU Leuven), Koen Yskout (imec-DistriNet, KU Leuven), Riccardo Scandariato (Software Engineering Division, University of Gothenburg), and Wouter Joosen (imec-DistriNet, KU Leuven)

Security Lancaster | Lancaster University

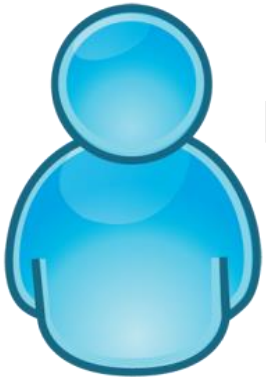
Light-touch Interventions to Improve Software Development Security

Charles Weir, Lynne Blair (LU),
Ingolf Becker, Angela Sasse (UCL), James Noble (VWU)

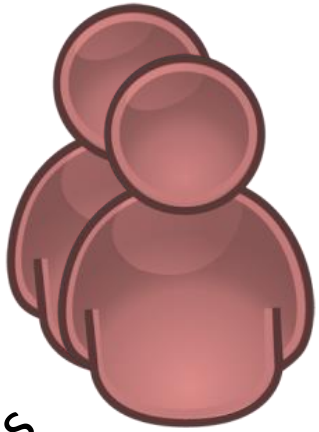
Secure Development Process

Intervener

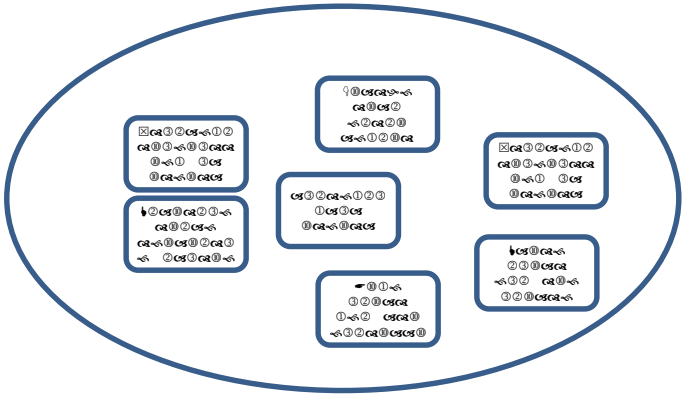
Developers



Instructs



Provides



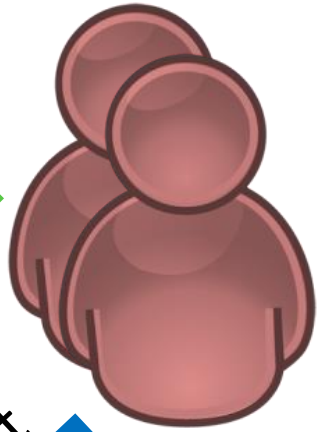
Tools and Techniques

Developer Centric

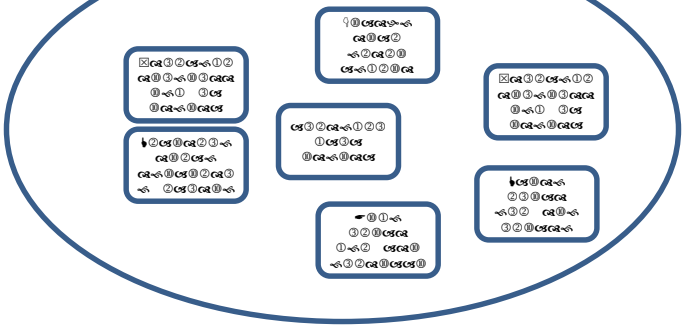
Intervener

Developers

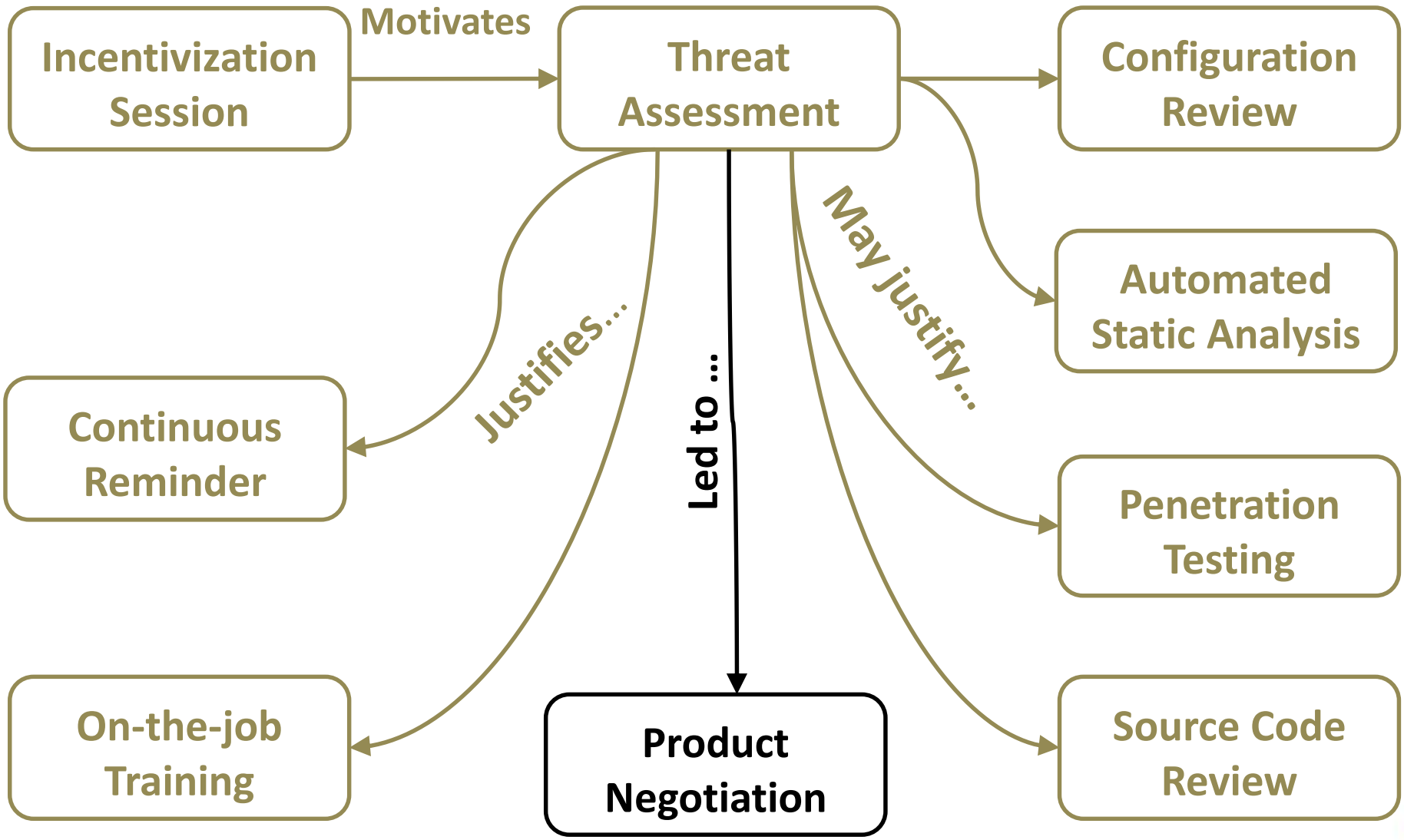
Persuades



Select



Tools and Techniques


Incentivization Session
Motivates
Threat Assessment
Configuration Review
Automated Static Analysis
Penetration Testing
Source Code Review
Product Negotiation
On-the-job Training
Continuous Reminder
Justifies…
Led to …
May justify…

-	+
Introducing the Component Analyzer	Supplier support and enthusiasm
Upgrade costs	Upgrade stories, Traffic lights
Cost of security enhancements	Benefits seen by product management
Difficulty of learning security	Group learning

ENTERPRISE THREAT MODELING

- **Scalable Static Analysis to Detect Security Vulnerabilities: Challenges and Solutions**

  Francois Gauthier, Nathan Keynes, Nicholas Allen, Diane Corney, and Padmanabhan Krishnan (Oracle Labs, Australia)

- **Applied Threat Driven Security Verification**

  Danny Dhillon and Vishal Mishra (Dell)

- **Rethinking Secure DevOps Threat Modeling: The Need for a Dual Velocity Approach**

  Altaz Valani (Security Compass)

- **Automating Threat Intelligence for SDL**

  Raghudeep Kannavara (Intel), Jacob Vangore, William Roberts (Olivet Nazarene University), Marcus Lindholm, and Priti Shrivastav (Intel)

VULNERABILITY ASSESSMENT

- **Towards Understanding the Adoption of Anti-Spoofing Protocols in Email Systems**

  Hang Hu, Peng Peng, and Gang Wang (Virginia Tech)

- **There's a Hole in the Bottom of the C: On the Effectiveness of Allocation Protection**

  Ronald Gil (MIT CSAIL), Hamed Okhravi (MIT Lincoln Laboratory), and Howard Shrobe (MIT CSAIL)

- **Security Concerns and Best Practices for Automation of Software Deployment Processes – An Industrial Case Study**

  Vaishnavi Mohan (Deloitte Analytics Institute), Lotfi ben Othmane (Iowa State University), and Andre Kres (IBM)

NEW SECURITY NEEDS AND APPROACHES

- **Reducing Attack Surface via Executable Transformation**

  Sukarno Mertoguno, Ryan Craven, Daniel Koller, and Matthew Mickelson (ONR)

- **Designing Secure and Resilient Embedded Avionics Systems**

  Jason H. Li (Intelligent Automation), Douglas Schafer (AFRL), David Whelihan (MIT Lincoln Labs), Stefano Lassini (GE Aviation Systems), Nicholas Evancich, Kyung Joon Kwak (Intelligent Automation), Mike Vai, and Haley Whitman (MIT Lincoln Labs)

- **Data Integrity: Recovering from Ransomware and Other Destructive Events**

  Timothy McBride (NIST), Anne Townsend, Michael Ekstrom, Lauren Lusty, and Julian Sexton (MITRE)

- **Securing Wireless Infusion Pumps**

  Andrea Arbelaez (NIST), Sue Wang, Sallie Edwards, Kevin Littlefield, and Kangmin Zheng (MITRE)

NEW SECURITY NEEDS AND APPROACHES

- **Best Practice for Developing Secure and Trusted Enterprise Storage & Computing Products**

  Xuan Tang (Dell)

- **Experiment: Sizing Exposed Credentials in GitHub Public Repositories for CI/CD**

  Hasan Yasar (Software Engineering Institute, CMU)

MY TAKEAWAYS

- Security is possible, but we often don't bother
  Often due to cost and perceived need
- Tools are getting better…
  …but are only as good as their inputs
- Educators are starting to realize the need for good education
- US Government is understanding the need for better security… and helping make it happen