Security and Privacy in Internet of Things: Threats and Solutions

Dawn Song

UC Berkeley

Berkeley
UNIVERSITY OF CALIFORNIA

Massachusetts Institute of Technology

NSF

MICHIGAN

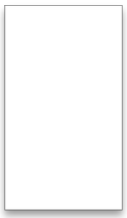
VANDERBILT UNIVERSITY

The world is becoming more and more connected

2 Billion

500 Million

600 Million



500 Million

23 Million

90 Million

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

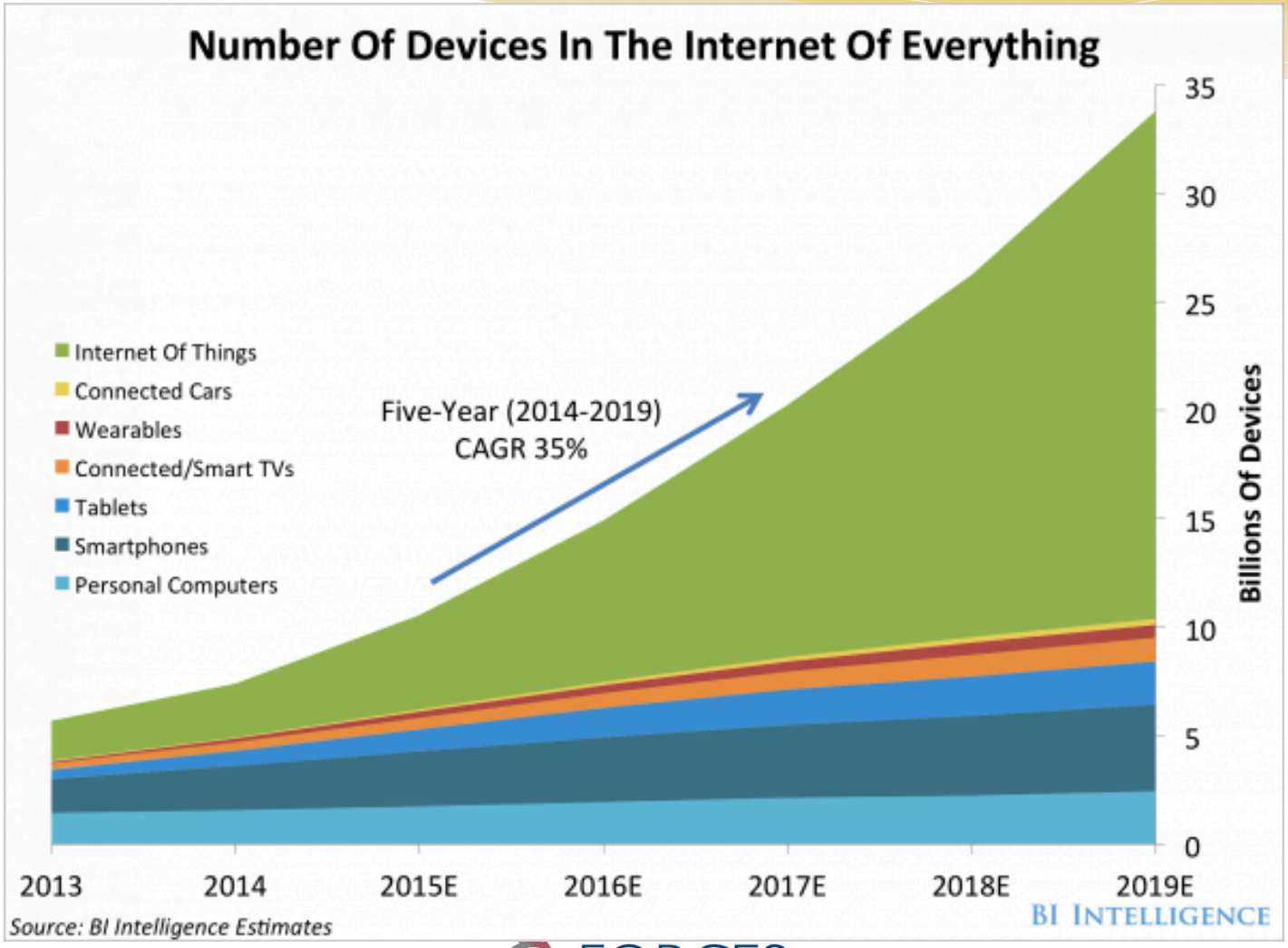
Connected Parking Meters

San Francisco: connected parking meters help drivers find open parking spots without driving around the same block several times.

Connected Diaper

Features an embedded chip that sends SMS messages to parents or babysitters when the diaper is wet. Only cost an extra two cents to produce than normal diapers.

Over 50 Billion connected devices by 2020



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

The world is becoming more and more connected

Malware enters new landscape as more parts of the world get connected

ThingBots

More than 750,000 phishing and spam emails launched from ”ThingBots” including smart televisions, fridge. [Proofpoint]

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Malware enters new landscape as more parts of the world get connected

* Legacy, traditional vulnerabilities & attacks in new landscape

Apple Watches Patch Critical Buffer-overflow Vulnerabilities

"Processing a maliciously crafted font file may lead to arbitrary code execution.

A memory corruption issue existed in the processing of font files.

This issue was addressed through improved bounds checking." – Apple Advisory

DARPA Hacks GM's OnStar To Remote Control A Chevrolet Impala (Feb 2015)

Charlie Miller

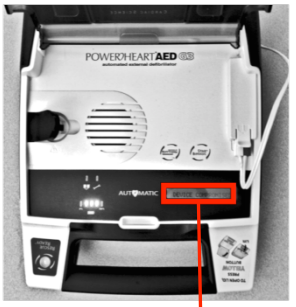
Chris Valasek

* Dial into the OnStar system (locally), and feed it with malicious packets (containing code), and take control of the car

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

First Security Analysis on Medical Devices

- Cardiac Science G3 Plus model 9390A
- Analysis
  - Manual reverse engineering using IDA Pro
    - *MDLink*, *AEDUpdate* and device *firmware*
  - Automatic binary analysis
    - BitBlaze binary analysis infrastructure
    - BitFuzz, the dynamic symbolic execution tool
- Vulnerabilities lead to distributed worm in AED
  1. AED Firmware - Replacement
  2. AEDUpdate - Buffer overflow
  3. AEDUpdate - Plain text user credentials
  4. MDLink - Weak password scheme



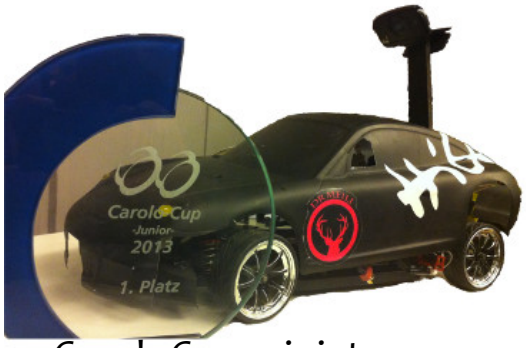
The case for Software Security Evaluations of Medical Devices [HealthSec'11]

OpenDavinci

* A realtime-capable software development and runtime environment for CPS.

* Use cases

UC Berkeley's AGV



CaroloCup miniature competition 2013 & 2014

Univ. of Arizona's AGV

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Smart Fuzzing: Results (on OpenDavinci)

* target app: ***odrecintegrity***

Metrics	Value
run time	25 hours
total execs	11.5M times
total crashes	238K
unique crashes	**31**

* target app: ***odsplit***

Metrics	Value
run time	25 days
total execs	2.21M times
total crashes	2.16M
unique crashes	**5000+**

* Analysis reveals vulnerabilities in OpenDavinci

Malware enters new landscape as more parts of the world get connected

* Legacy, traditional vulnerabilities & attacks in new landscape

* New classes of vulnerabilities & attacks on new platform

Smart Locks

Keys

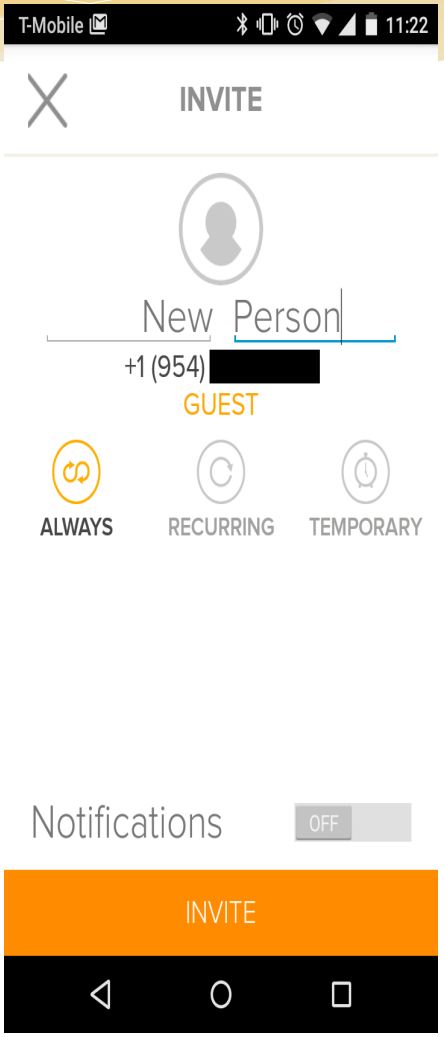
Smart Locks (Interior)

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Digital Keys: Versatile Access Control

Key Types: Better Access Granularity

1. **Owner**
2. **Resident:** Tenant, Boyfriend/ Girlfriend
3. **Recurring Guest:** Babysitter, Dog Walker
4. **Temporary Guest:** Friend



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Security Model

Attacker Model

* Revoked attacker
* Thief
* Physically-present attacker
* Relay attacker

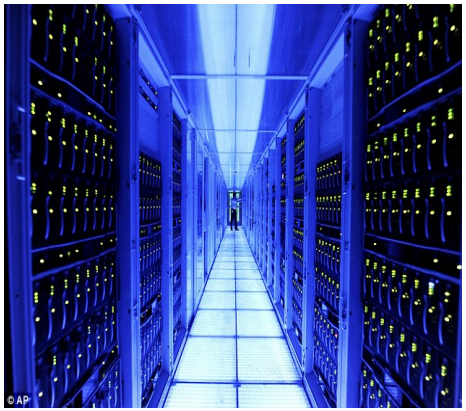
Security Goals

* Prevent unauthorized access
* Feature (access log) integrity
* Privacy from device manufacturer

Security Analysis Summary

1. State consistency attacks
2. Privacy Leakage
3. Ownership mental model violations
4. Unwanted unlocking attacks

Device-Gateway-Cloud Architecture



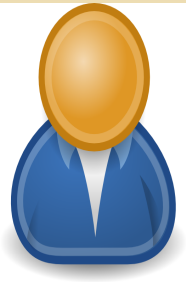
Device

Gateway

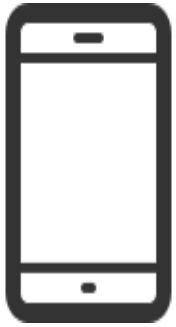
Cloud

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Normal Revocation in DGC Architecture



Revoke Phone G's access



Remove access to Grant's door

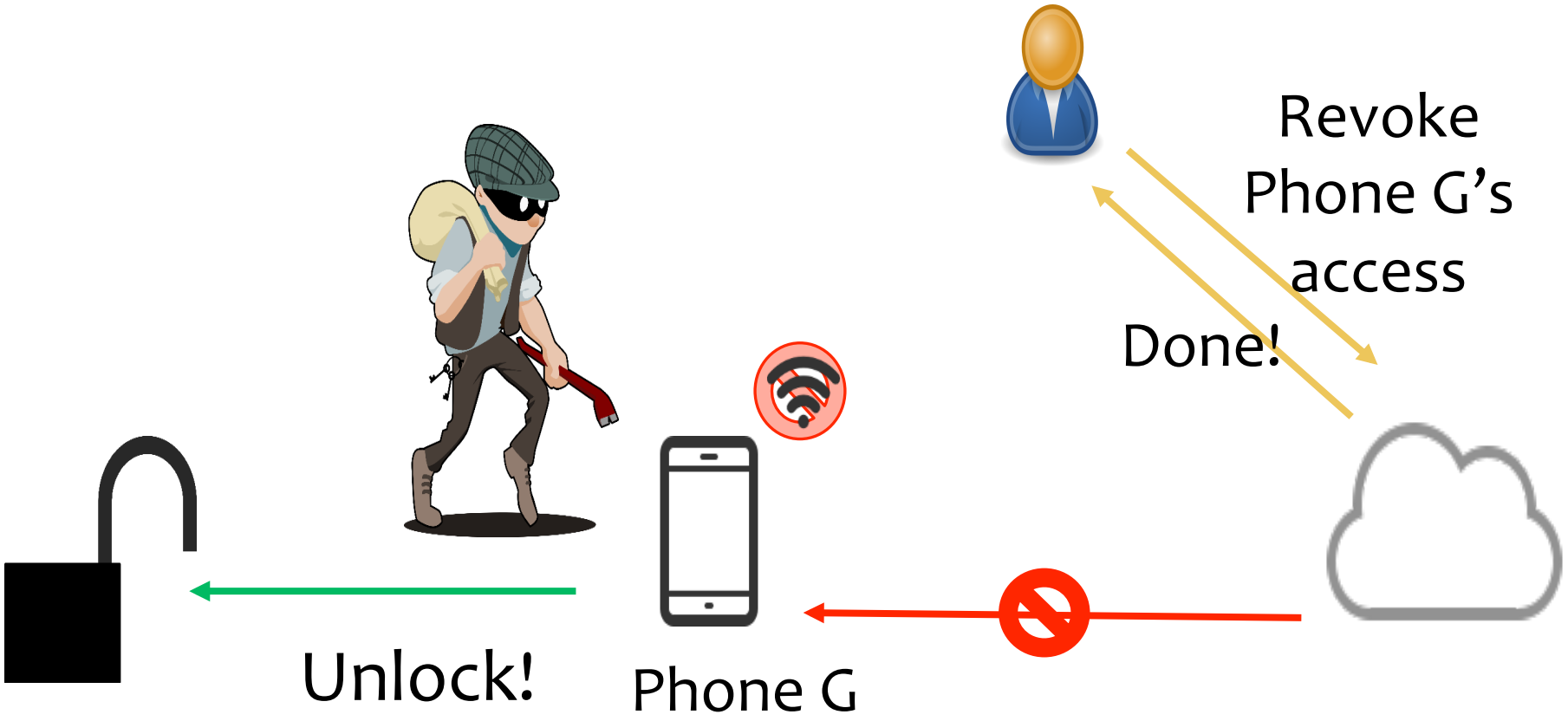
Lock Company's Server

Phone G

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Simple Attack: Offline Mode

Mallory steals phone and switches it offline



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Root Problem: CAP Theorem

* CAP Theorem: Consistency, Availability, Partitioning
* Smart lock company's servers and users' smart locks on doors = nodes in distributed system
* Each user interaction (mobile phone) forms ephemeral edge
  * Natural partitioning in system as users come and go
* Sometimes user interaction doesn't connect lock & servers: partition occurs.
  * Availability? (Allow interaction)
  * Consistency? (Ignore user's requests until server can be reached via user's phone)

Eventual Consistency

* Approach 1: Mandate consistency (lock must be able to talk to server before allowing action)

* Problems: lack of mobile data, server outage, lock company out of business

* Better Approach: Eventual Consistency

Eventual Consistency

* During every interaction, smart phone attempts to contact server & relay signed state updates to lock
* If phone cannot contact server or signed updates don't verify, lock uses existing ACL to fulfill user's request
* Security Property: lock will have updated ACL as soon as honest user interacts w/ the lock
  * Fast enough for vast majority of home access control needs

Malware enters new landscape as more parts of the world get connected

* Legacy, traditional vulnerabilities & attacks in new landscape
* New classes of vulnerabilities & attacks on new platform
* New threat models with new technology

Traditional Defenses

* Detecting and blocking malware
  * Anti-virus
  * APT
* Patching exploited vulnerabilities
* Most of commercial security solutions today

Reactive Defense

Reactive Defense Is Insufficient

* Focus on attacks
* Cat-&-mouse game
  * Needs to change as attacks change
* Malware can cause real physical damage
* Deploying patches may be difficult
  * May require additional certification

How to break the cat-and-mouse cycle?

* Eradicate root cause of attacks
  * Eliminate classes of vulnerabilities

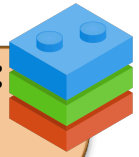
* Most Programs Contain Vulnerabilities
  * Developers unaware of security issues
  * Security is complex and hard to get right
  * Insufficient security support in most programming language/ framework

Proactive Defense

* Focus on enforcing specific security properties, instead of attacks
  * Define security properties
    * Many possibilities
  * Build secure system satisfying certain security properties
    * free of/secure against certain classes of vulnerabilities
  * Regardless of attacks
* Does not change with attacks
  * Break cat-&-mouse cycle
  * In contrast to reactive defense

Proactive Defense:
Bug Finding

Proactive Defense:
Secure by Construction



Proactive Defense:
Secure by Learning

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

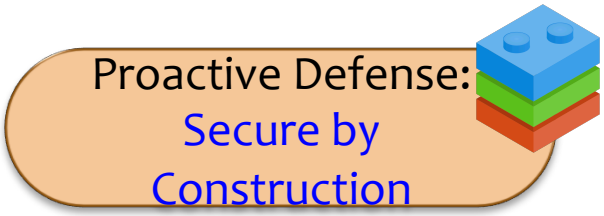
Proactive Defense: Automatic Bug Finding

* Lots of work in research; some work in industry
* Challenges:
  * Cannot guarantee finding all vulnerabilities (of certain classes)
    * High false positive/false negative
  * Asymmetry
    * Attacker only needs to find one vulnerability
  * Race with the attacker
    * Who finds the vulnerability first

Proactive Defense: Secure by Construction

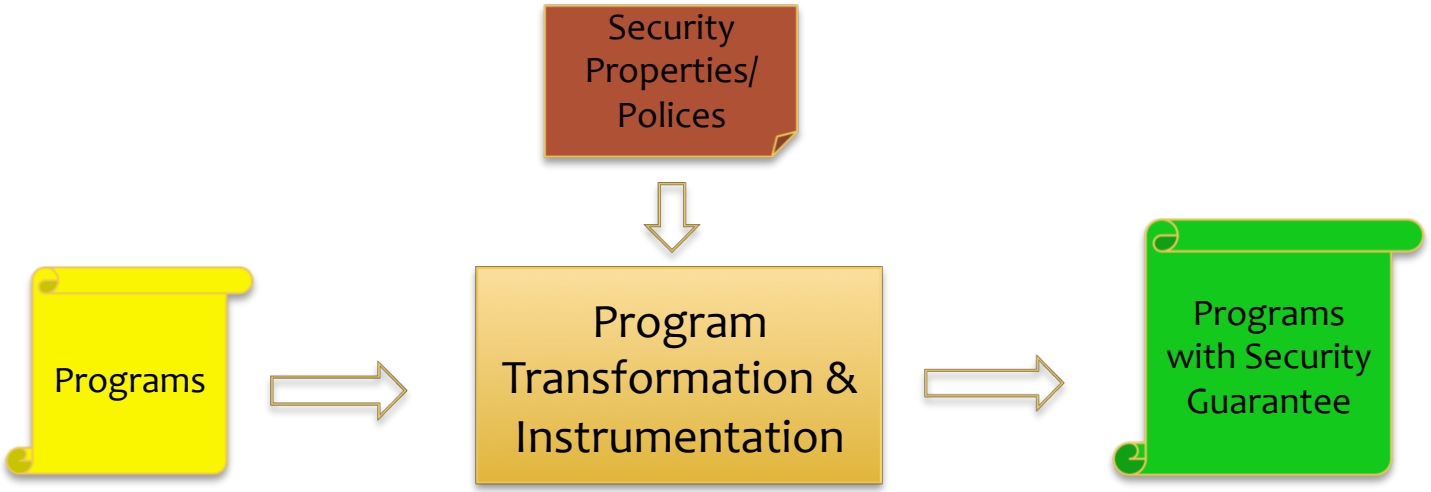
* Provide first principle and security mechanisms to ensure security property by construction
  * Instead of relying on bug finding after software is written
  * Free of/secure against certain classes of vulnerabilities by the way how software is constructed

* Practical
  * Low performance overhead
  * Compatibility
  * Little to no effort from developer



Building Systems Secure by Construction

* Program transformation and instrumentation to guarantee certain security properties
  * Compilation stage or binary rewriting post compilation



  * Case studies:
    * Low-level language/system domain: program hardening for OpenDavinci
    * Secure and privacy-preserving collaborative analytics

vs.

FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

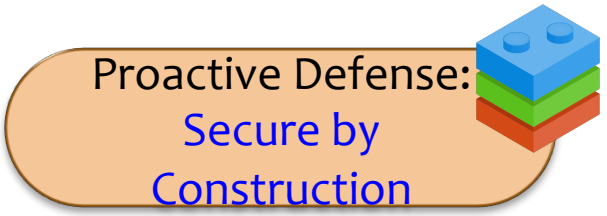
Proactive Defense: Secure by Construction

Automatic Program Hardening

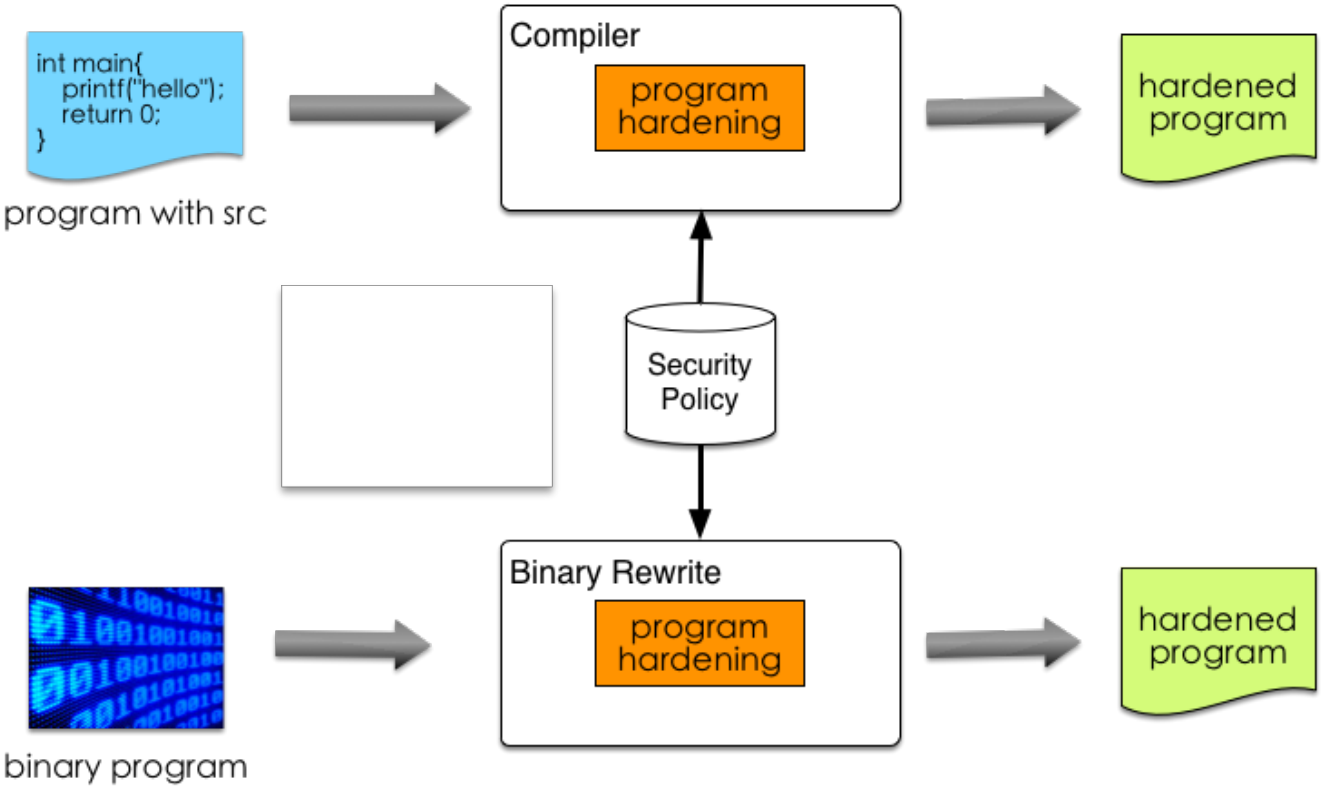
* Program in type unsafe language may always have memory-safety vulnerabilities

* Program hardening: make program difficult to be exploited even when it has vulnerabilities

* Goal:
  * Low performance overhead & strong security guarantee



Program Hardening: Our Solutions



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

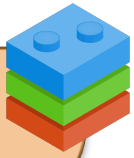
Proactive Defense: Secure by Construction

Program Hardening: Our Solutions

* *CCFIR: Compact Control-Flow Integrity and Randomization*
  * Easy to deploy: binary transformation
  * Ensure control-flow integrity
  * Low overhead: <4%
  * Earlier version won Special Recognition Award for Microsoft BlueHat Competition
* *VTint:*
  * Lightweight: <2% overhead
  * Binary-compatible
  * Defense against VTable hijacking
* *CPI: Code Pointer Integrity*
  * New security property: memory safety for code pointers
  * Sweet spot for strong security guarantee & efficiency
  * Harden the FreeBSD distribution

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

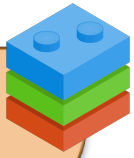
Proactive Defense:
Secure by Construction



Code Pointer Integrity (CPI)

* Implemented in LLVM
* Harden complete FreeBSD distribution (modulo kernel)
  * Protecting against control-flow hijacking attacks
  * >100 extra packages

Proactive Defense:
Secure by Construction



Harden OpenDavinci with CPI

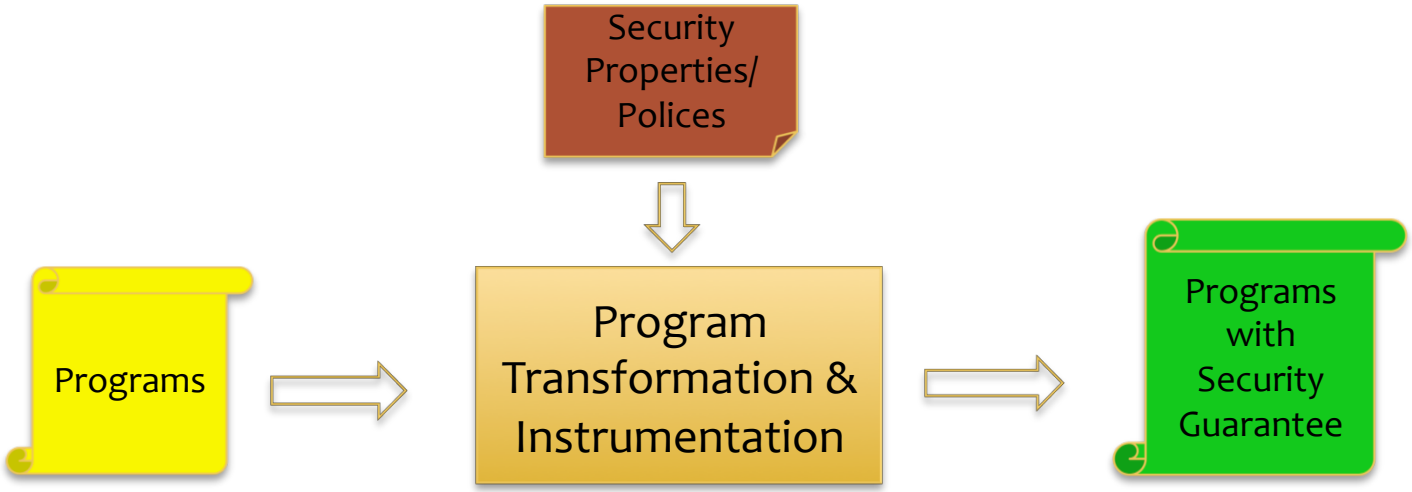
* Compilation time evaluation
  * The extra program hardening process takes a negligible time.

Time	Original compilation	CPI compilation
real	18m 45.762s	18m 50.381s
user	10m 1.032s	10m 2.336s
sys	0m 56.844s	0m 55.536s

* File size evaluation
  * All 30 hardened programs have the same size as non-hardened ones
* Work-in-progress
  * performance evaluation
  * security evaluation

Building Systems Secure by Construction

* Program transformation and instrumentation to guarantee certain security properties
  * Compilation stage or binary rewriting post compilation



  * Case studies:
    * Low-level language/system domain: program hardening for OpenDavinci
    * Secure and privacy-preserving collaborative analytics

vs.

FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Proactive Defense: Secure by Construction

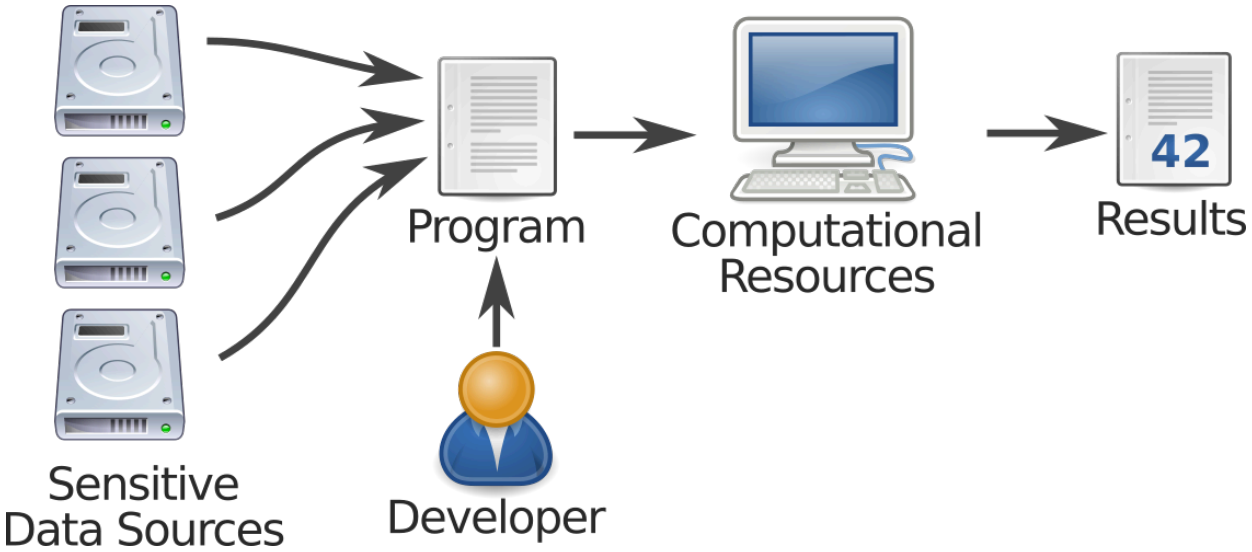
Collaborative Data Analytics in IoT

* Large-scale collaborative data analytics creates huge opportunities
  * Smart homes
  * Smart buildings
  * Smart cities
* But security & privacy concerns make data owners reluctant to share

Traditional Data Analytics

Existing collaborative data analytics requires data owners to trust:

- Programs querying the data are not malicious or buggy
- Computers running the programs do not leak data



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Challenges

- Unfriendly programming model
  - Requires programmer to code security checks in programs manually to enforce the security policy
  - Particularly hard for non-security expert
- Difficult to provide security & privacy guarantees
  - Rely on trust in program & computing resources
- Existing cryptographic solutions are not efficient
  - Secure Mulpti-party Computation (SMC) doesn't scale to desired data analytics tasks

Our Approach

- Strong security & privacy guarantee
  - Policy enforcement guaranteed, even when programs & OS are malicious
- Friendly programming model
  - Programmers do not need to write security checks manually
  - Security checks inferred and instrumented automatically from policy
- Practical performance
  - Leverage trusted hardware (SGX extensions) to both guarantee security and approach performance of existing analytics frameworks

Our Approach: Secure Collaborative Analytics

Attach policies to Data

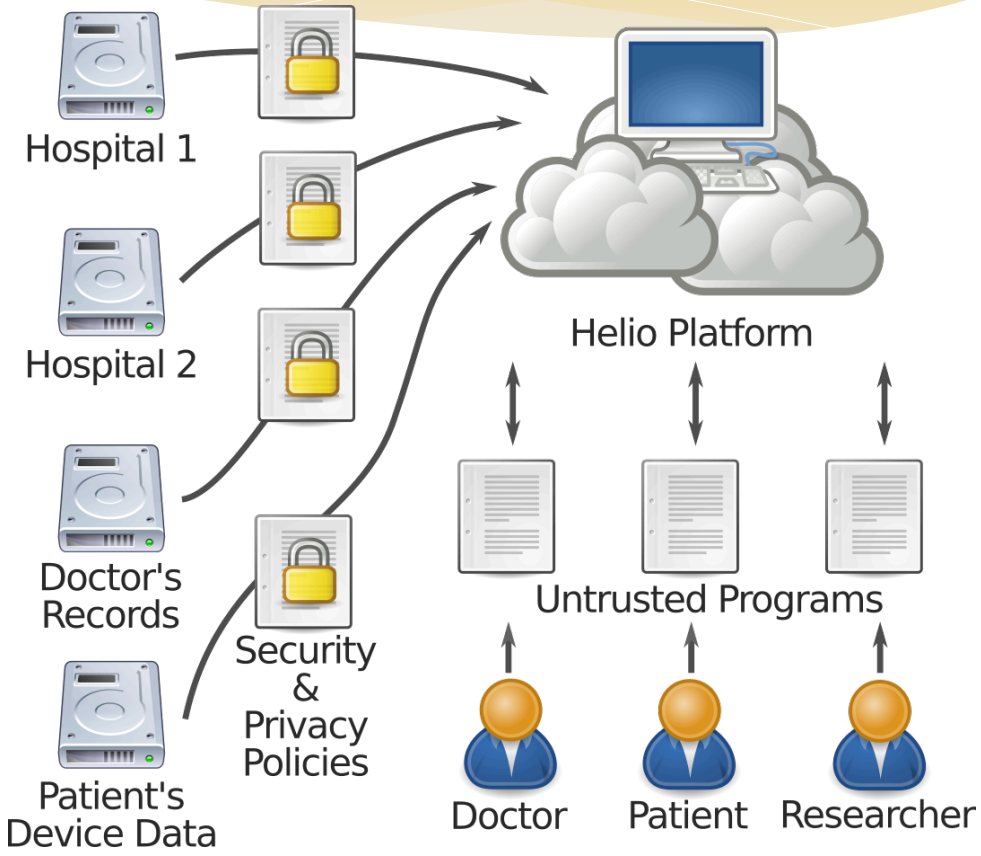
- System guarantees enforcement of policies

Program transformation for security

- Programmer doesn't have to write security checks

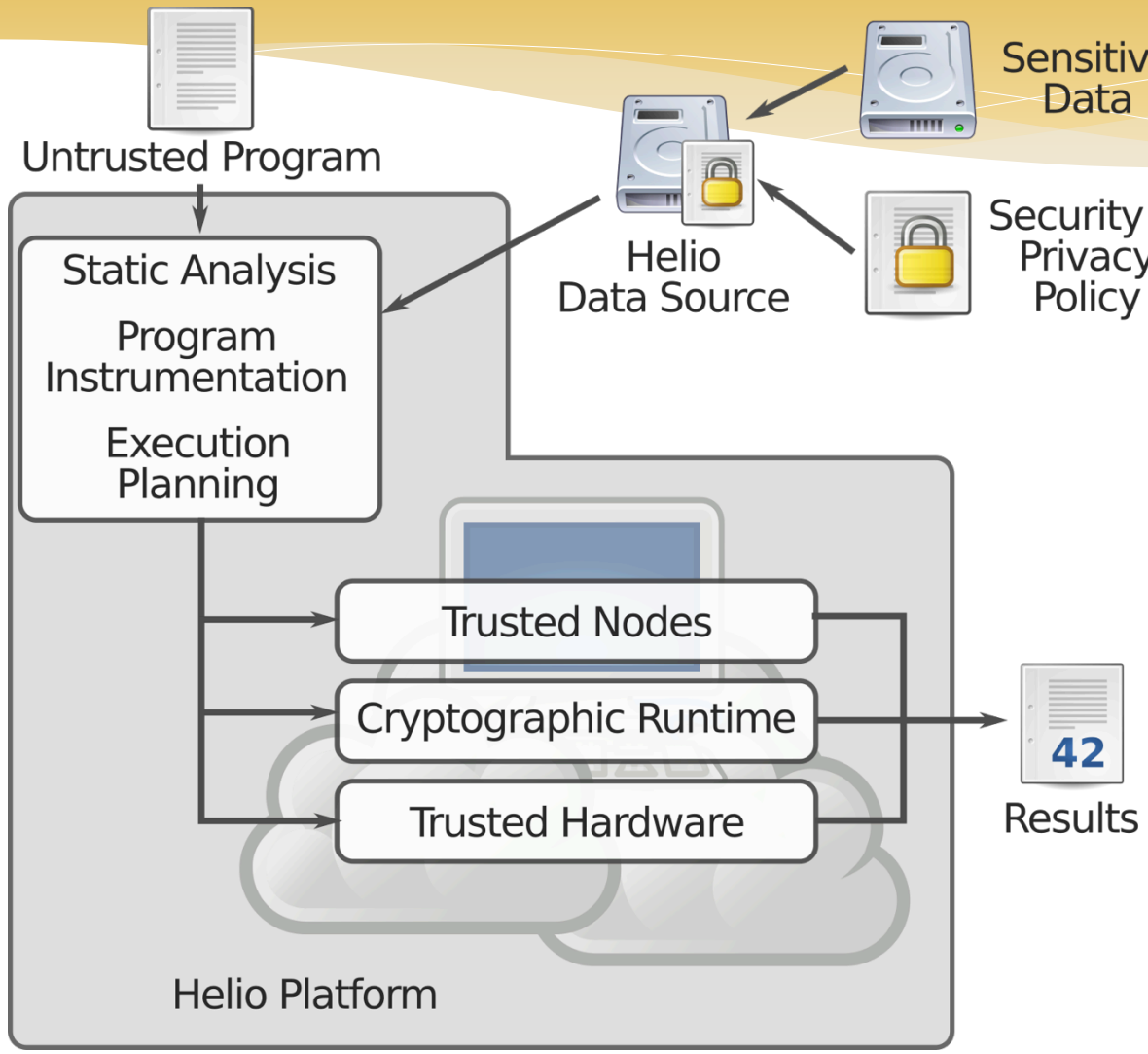
Uses Trusted Hardware

- Security guarantee with low overhead



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Our Approach: Secure Collaborative Analytics



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Towards Building Systems Secure by Construction

* Better leverage of engineering effort
* De facto in 10 or 20 years?

Reacting to Attacks

Effort ~ # of attacks
Each vulnerability ->
many or infinite # of attacks

Racing with Attacks

Effort ~ # of programs

Eradicating Attacks

Effort ~ # of core security primitives & mechanisms

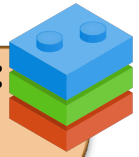
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Proactive Defense

* Focus on enforcing specific security properties, instead of attacks
  * Define security properties
    * Many possibilities
  * Build secure system satisfying certain security properties
    * free of/secure against certain classes of vulnerabilities
  * Regardless of attacks
* Does not change with attacks
  * Break cat-&-mouse cycle
  * In contrast to reactive defense

Proactive Defense: Bug Finding

Proactive Defense: Secure by Construction



Proactive Defense: Secure by Learning

FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Secure by Learning

* Automatic learning of security policies in IoT/CPS
  * Enforces learned security policies
  * Anomaly detection

* Complement secure-by-construction approaches
  * Some aspects (e.g., user behaviors, sensor readings) or legacy components may not be covered by secure-by-construction methods
  * Particularly important in distributed, complex systems

* Leverage huge advances in ML in other domains
  * E.g., deep learning in vision, speech, text
  * Big data for security

The world is becoming more and more connected

2 Billion

500 Million

600 Million

500 Million

23 Million

90 Million

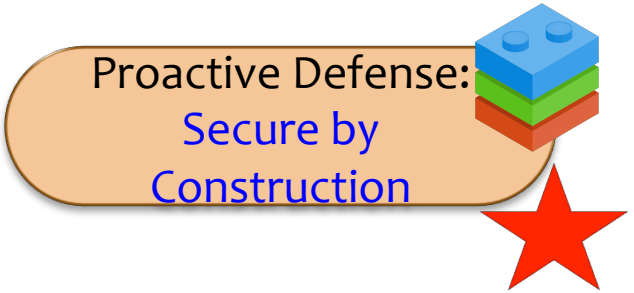
FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

Malware enters new landscape as more parts of the world get connected

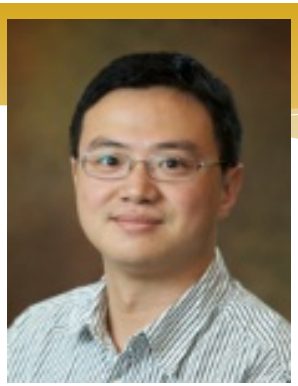
* Legacy, traditional vulnerabilities & attacks in new landscape
* New classes of vulnerabilities & attacks on new platform
* New threat models with new technology

Proactive Defense

* Focus on enforcing specific security properties, instead of attacks
  * Define security properties
    * Many possibilities
  * Build secure system satisfying certain security properties
    * free of/secure against certain classes of vulnerabilities
  * Regardless of attacks
* Does not change with attacks
  * Break cat-&-mouse cycle
  * In contrast to reactive defense



FORCES
FOUNDATIONS OF RESILIENT CYBER-PHYSICAL SYSTEMS

dawnsong@cs.berkeley.edu