CySER Seminar | Spring 2023
March 20, 2023

Exploring Platform Reboot As A Security Measure For Cyber-Physical Systems

Monowar Hasan

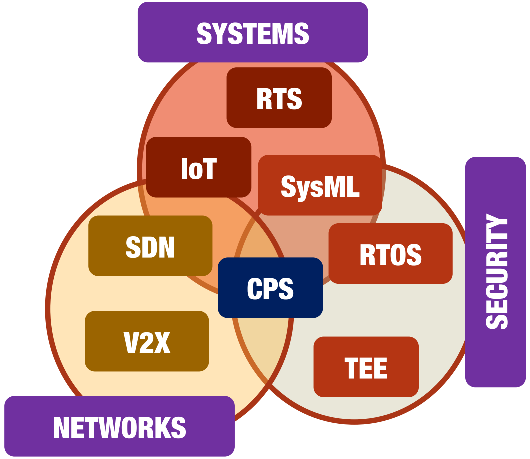
Assistant Professor, School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA
monowar.hasan@wsu.edu

WASHINGTON STATE
UNIVERSITY

About Me

- Assistant Professor
  - EECS@WSU
  - Cyber-Physical Systems Security Research Lab (CPS2RL) [https://cps2rl.github.io]
  - Past: Wichita State (Asst. Prof. 2021-2022), UIUC (PhD, 2020), UM (MSc, 2015)

- Research: Systems, Security, Networking
  - Security for real-time, IoT, and cyber-physical systems
  - Trustworthy ML for embedded/IoT systems
  - Resilient real-time networks using SDNs
  - Security and resource management for vehicular communication networks

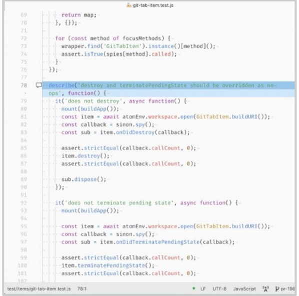


Today's Talk

Securing Cyber-Physical Systems by Platform Reboot

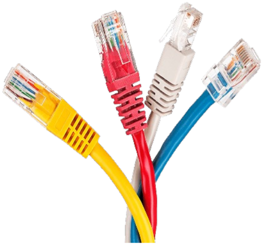
Cyber-Physical Systems (CPS)

CYBER


git-tab-item.test.js

```
      return map;
    }, {});

    for (const method of focusMethods) {
      wrapper.find('GitTabItem').instance()[method]();
      assert.isTrue(spies[method].called);
    }
  });

  describe('destroy and terminatePendingState should be overridden as no-
  ops', function() {
    it('does not destroy', async function() {
      mount(buildApp());
      const item = await atomEnv.workspace.open(GitTabItem.buildURI());
      const callback = sinon.spy();
      const sub = item.onDidDestroy(callback);

      assert.strictEqual(callback.callCount, 0);
      item.destroy();
      assert.strictEqual(callback.callCount, 0);

      sub.dispose();
    });

    it('does not terminate pending state', async function() {
      mount(buildApp());

      const item = await atomEnv.workspace.open(GitTabItem.buildURI());
      const callback = sinon.spy();
      const sub = item.onDidTerminatePendingState(callback);

      assert.strictEqual(callback.callCount, 0);
      item.terminatePendingState();
      assert.strictEqual(callback.callCount, 0);
```

Software, Control Algorithms, Code

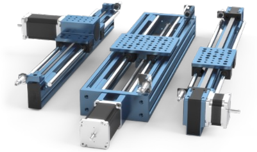


Networking, Communication

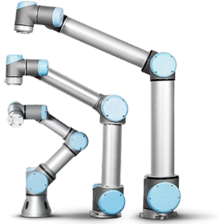
Microcontrollers, ECU, PLC

PHYSICAL

Sensors

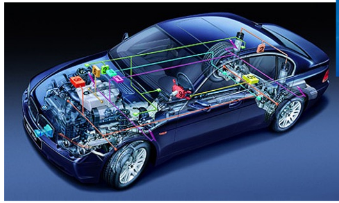


Actuators



Plant

Automobiles



Avionics

Control Systems

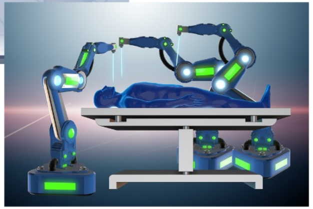
CPS Applications

Unmanned Vehicles

Autonomous Driving

Manufacturing

Healthcare



Surveillance

* Image courtesy: Google Image Search

CPS Security

Traditional CPS

- Custom Hardware
- Proprietary Operating System
- Proprietary Software
- Limited Network Connection

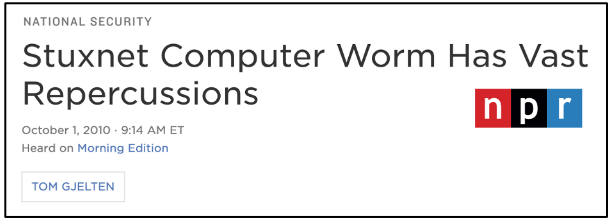
Modern CPS

- COTS Hardware
- Open Source Operating System
- Open Source Software
- More Connectivity → Internet!

Larger Attack Surface!

Modern CPS are vulnerable to security threats!

Increased Security Risks



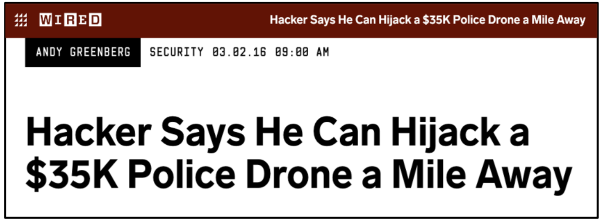
NATIONAL SECURITY

Stuxnet Computer Worm Has Vast Repercussions

npr

October 1, 2010 · 9:14 AM ET
Heard on Morning Edition

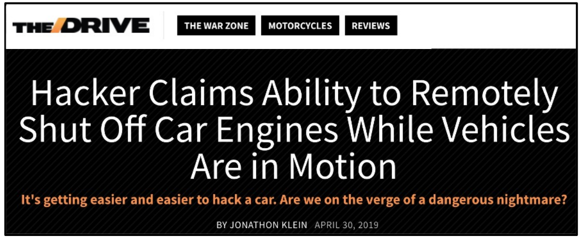
TOM GJELTEN



WIRED Hacker Says He Can Hijack a $35K Police Drone a Mile Away

ANDY GREENBERG SECURITY 03.02.16 09:00 AM

Hacker Says He Can Hijack a $35K Police Drone a Mile Away



THE DRIVE THE WAR ZONE MOTORCYCLES REVIEWS

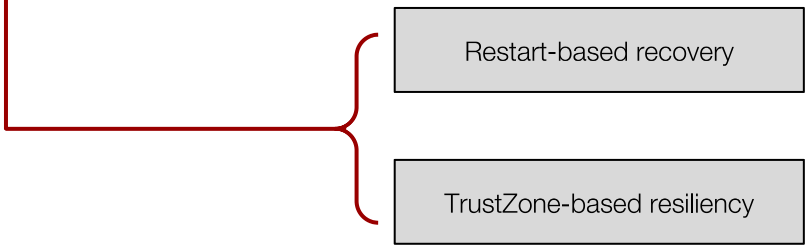
Hacker Claims Ability to Remotely Shut Off Car Engines While Vehicles Are in Motion

It's getting easier and easier to hack a car. Are we on the verge of a dangerous nightmare?

BY JONATHON KLEIN APRIL 30, 2019

Attack Resilient CPS Platforms

- Security issues → leads to safety issues
  - Difficult to ensure system won't be compromised
- Goal:
  - Provide guaranteed safety → under attack
- Proposed idea:
  - Proactive mechanism → prevents attack from progressing



Restart-based recovery

TrustZone-based resiliency

The Rest of Today's Talk

ReSecure [IoT'18, ICCPS'18]

Preserving Physical Safety under Cyber Attacks

[IoT'18] F. Abdi, C. Chen, M. Hasan, S. Liu, S. Mohan and M. Caccamo, "Preserving Physical Safety Under Cyber Attacks," iEEE Internet of Things Journal, Aug. 2019.

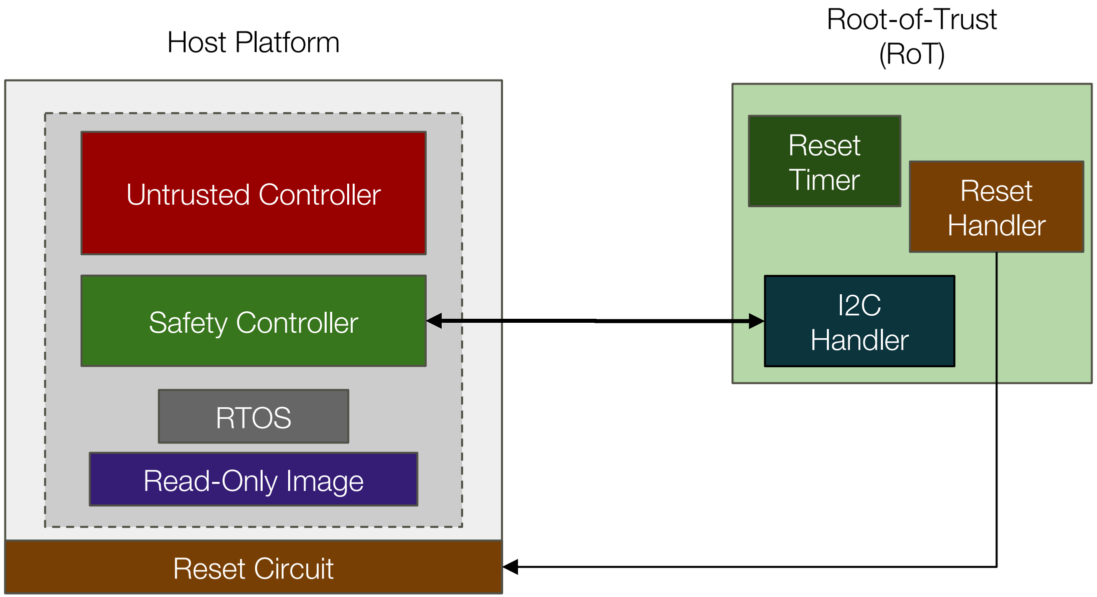
[ICCPS'18] F. Abdi, C. Chen, M. Hasan, S. Liu, S. Mohan and M. Caccamo, "Guaranteed Physical Security with Restart-Based Design for Cyber-Physical Systems," ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2018.

Our Approach: ReSecure [ICCPS'18]

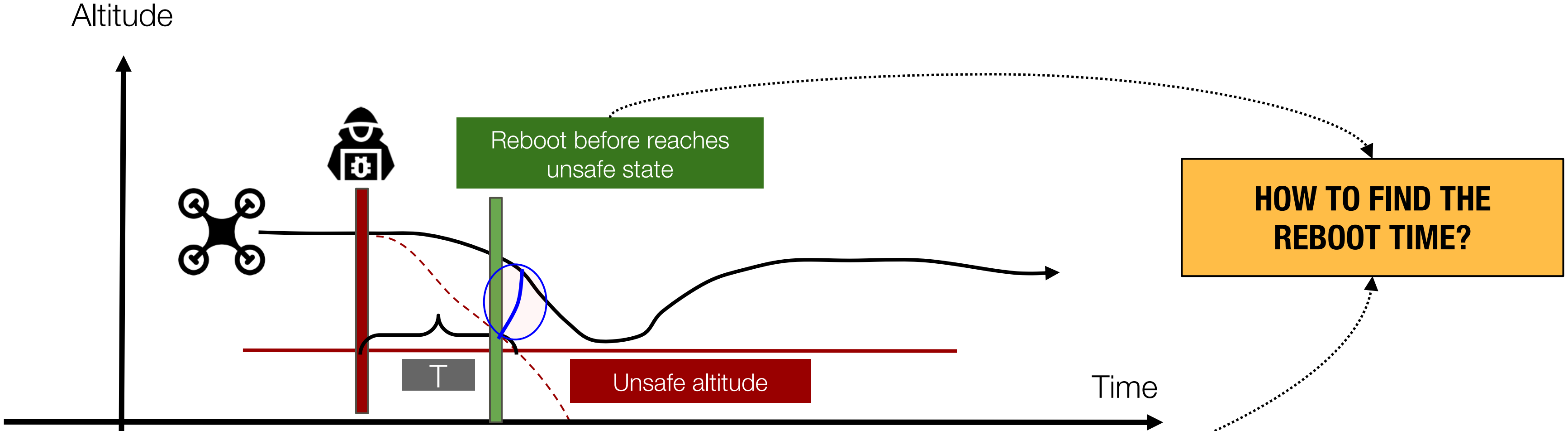
- Restart the system once a while to reset any attack progress
- Employ a Safety Controller (SC) and a Root-of-Trust (RoT) module

ReSecure: Design

- Host platform
  - Untrusted controller
  - Safety controller
- Root-of-Trust
  - Enforces restart



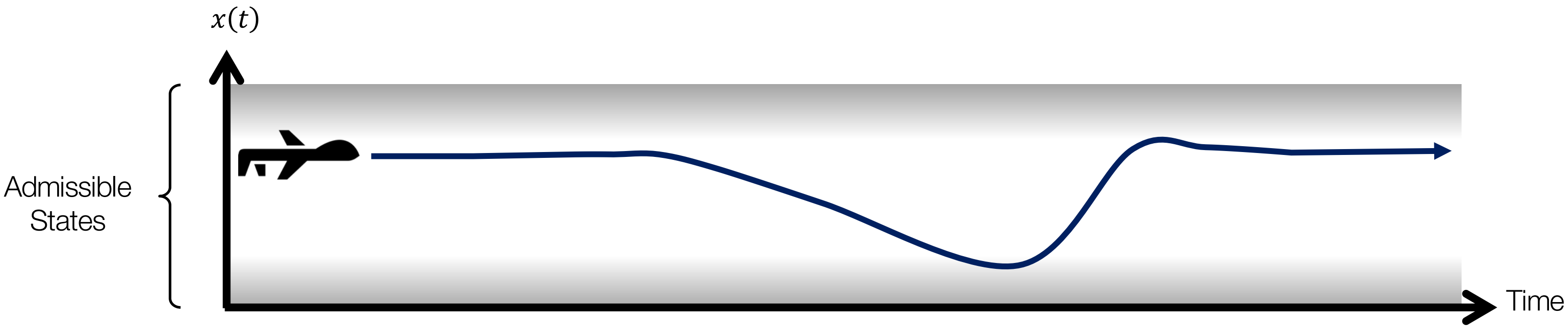
ReSecure: Overview


Altitude
Reboot before reaches unsafe state
HOW TO FIND THE REBOOT TIME?
T
Unsafe altitude
Time

- For a given state:
    - Calculate the shortest time T → system goes to “unsafe” state
- Reboot and reload the system before T

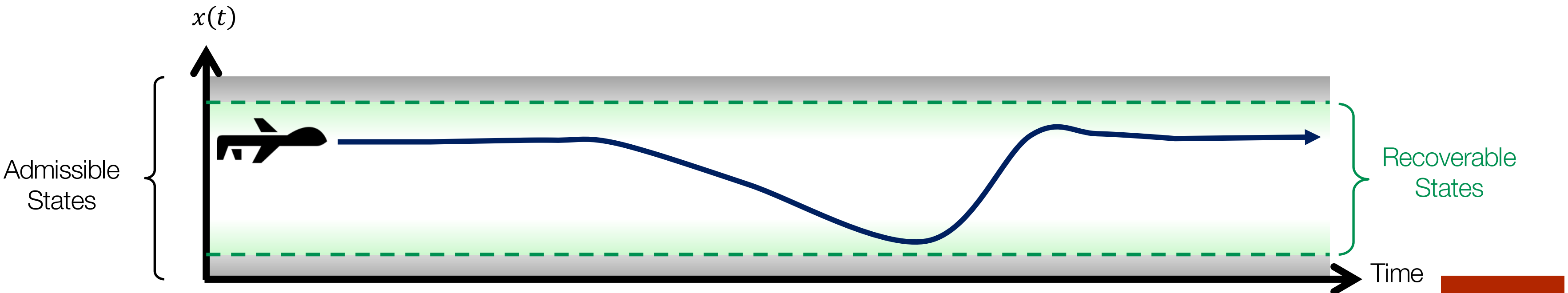
CPS States

- Admissible States S
  - States that do not violate any of the operational constraints of the physical plant
  - Safety invariant: system must always remain inside admissible states: $\forall t: x(t) \in S$



CPS States

- Admissible States S
  - States that do not violate any of the operational constraints of the physical plant
  - Safety invariant: system must always remain inside admissible states: $\forall t: x(t) \in S$
- Recoverable States R
  - Defined with regards to a given safety controller (SC)
  - A subset of admissible states ($R \subseteq S$) such that
    - if the given SC starts controlling system from $\mathrm{x} \in \mathrm{R}$, all future states will remain admissible



Determine Recoverable States

Reachability Analysis

- True Recoverable States:
  - All the states from which safety controller can stabilize the plant within α time.

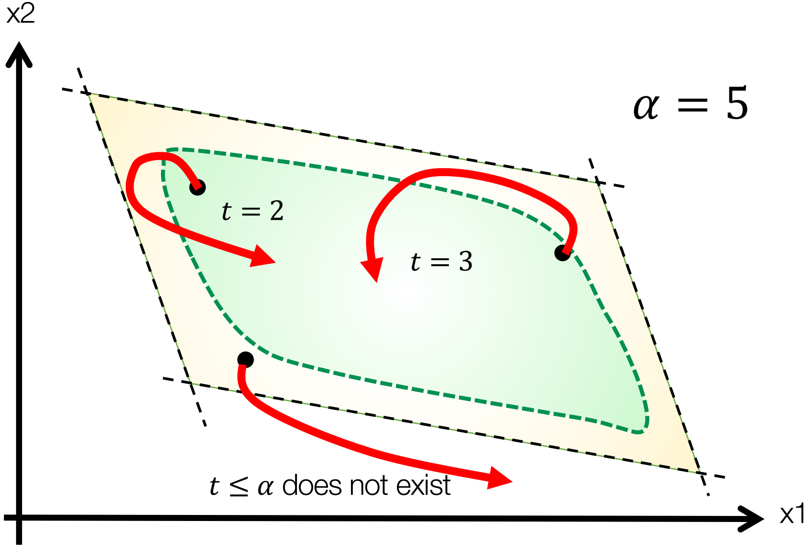
$$\Gamma_\alpha = \{\, x \mid$$

$$Reach_{\leq\alpha}(x, SC) \subseteq S \;\&$$

During recovering, the system should remain in admissible states.

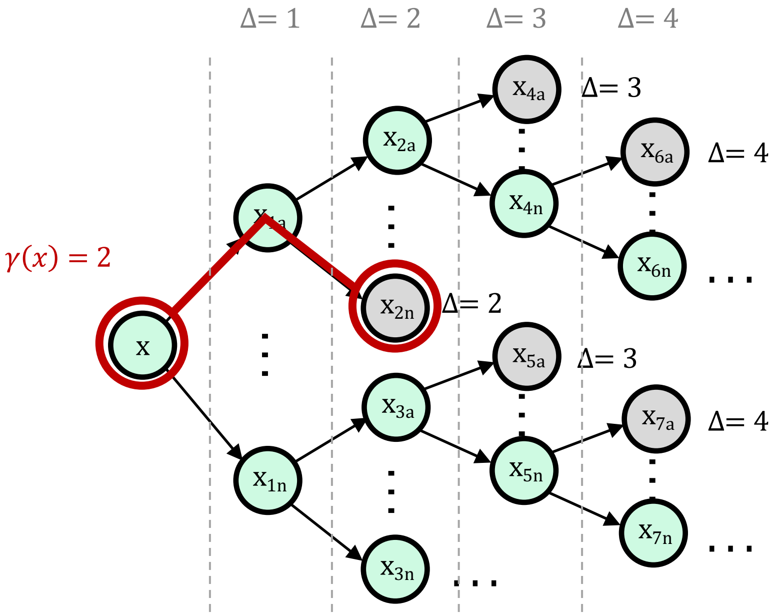
$$Reach_{=\alpha}(x, SC) \subseteq R\}$$

The destination should be a recoverable state.



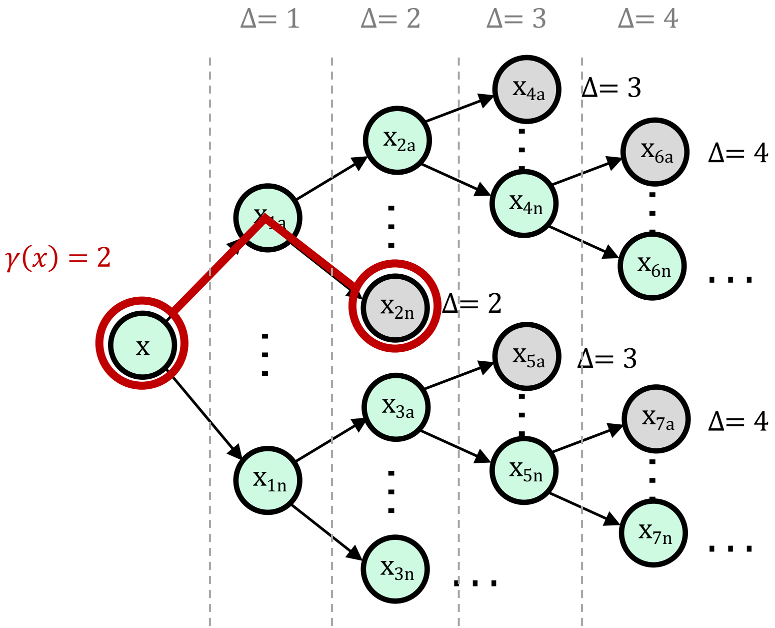
Determine Next Restart Time

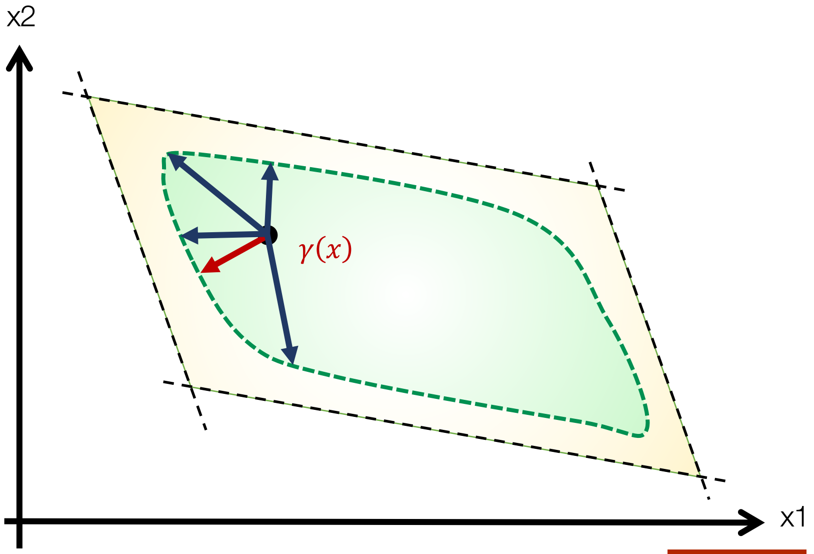
- From a given state:
  - Calculate the shortest time, $\gamma(x)$, to an unsafe state



Determine Next Restart Time

- From a given state:
  - Calculate the shortest time, $\gamma(x)$, to an unsafe state

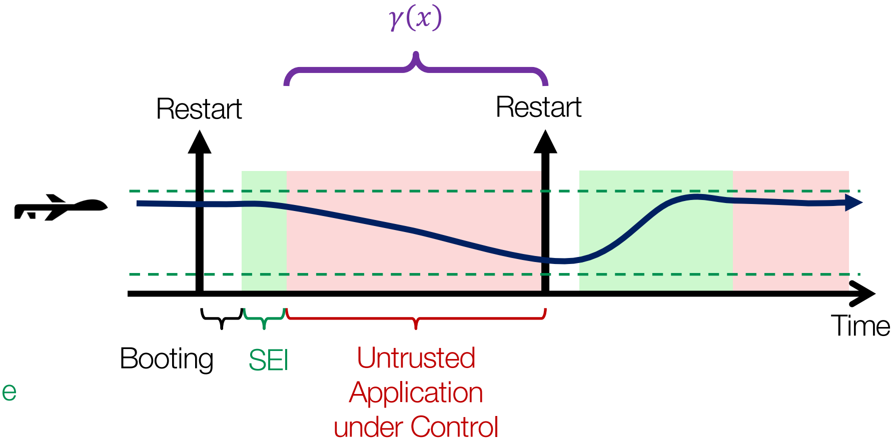




ReSecure: Workflow

- The system enters a Secure Execution Interval (SEI) during booting
  - The software is uncompromised
  - Access to RoT is enabled during SEI only

- Execution steps:
  1. Boot up (software is loaded)
  2. Enter SEI
  3. Run safety controller
  4. Check the system's state
  5. Compute next SEI time $\gamma(x)$
  6. Configure the restart timer on the RoT module (then RoT module closes I²C)
  7. Exit SEI, jump to user's application (the untrusted controller)



Restart-based Recovery

Remarks

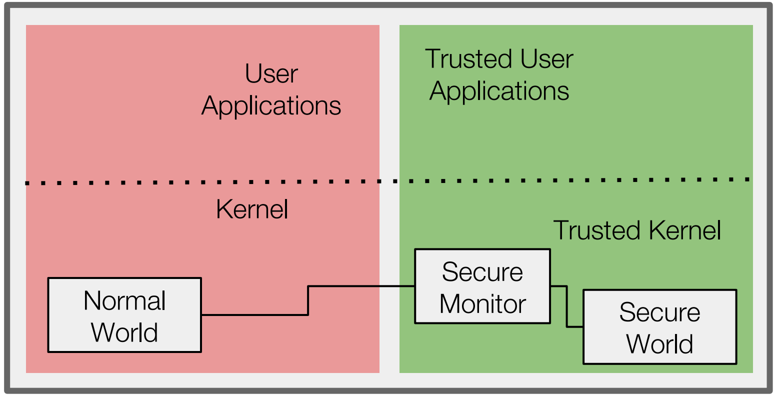
- Restarts are costly!
  - Platform specific
    - large restart time → not suitable for highly dynamic systems
- Require custom hardware
  - Root-of-Trust

Follow-up work [IoT'18]

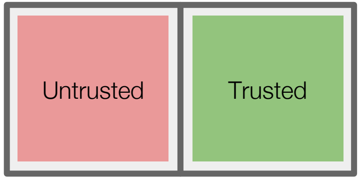
TrustZone-assisted recovery

arm TRUSTZONE

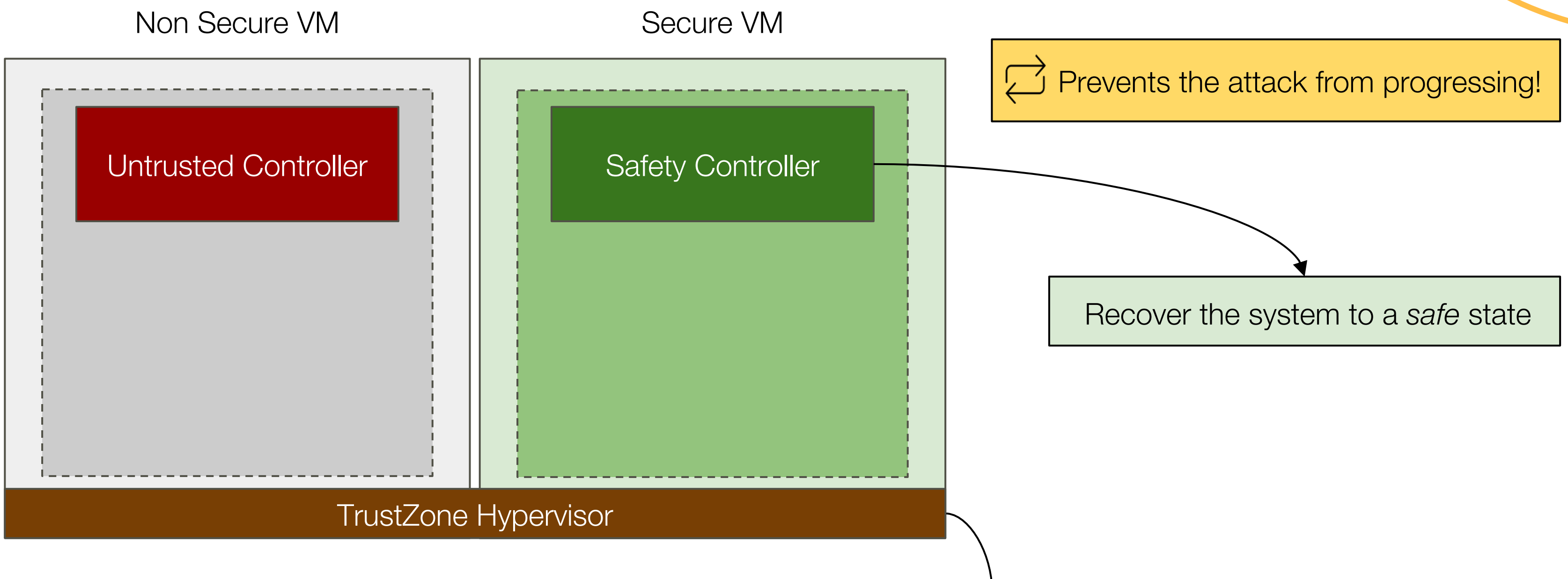
Background – ARM TrustZone



arm TRUSTZONE → isolates trusted software and data



TrustZone-based Recovery



Prevents the attack from progressing!

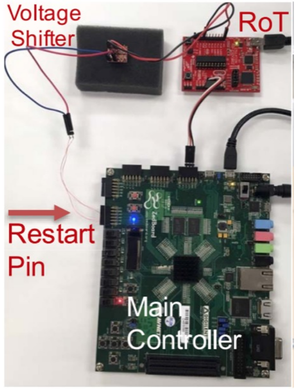
Recover the system to a *safe* state

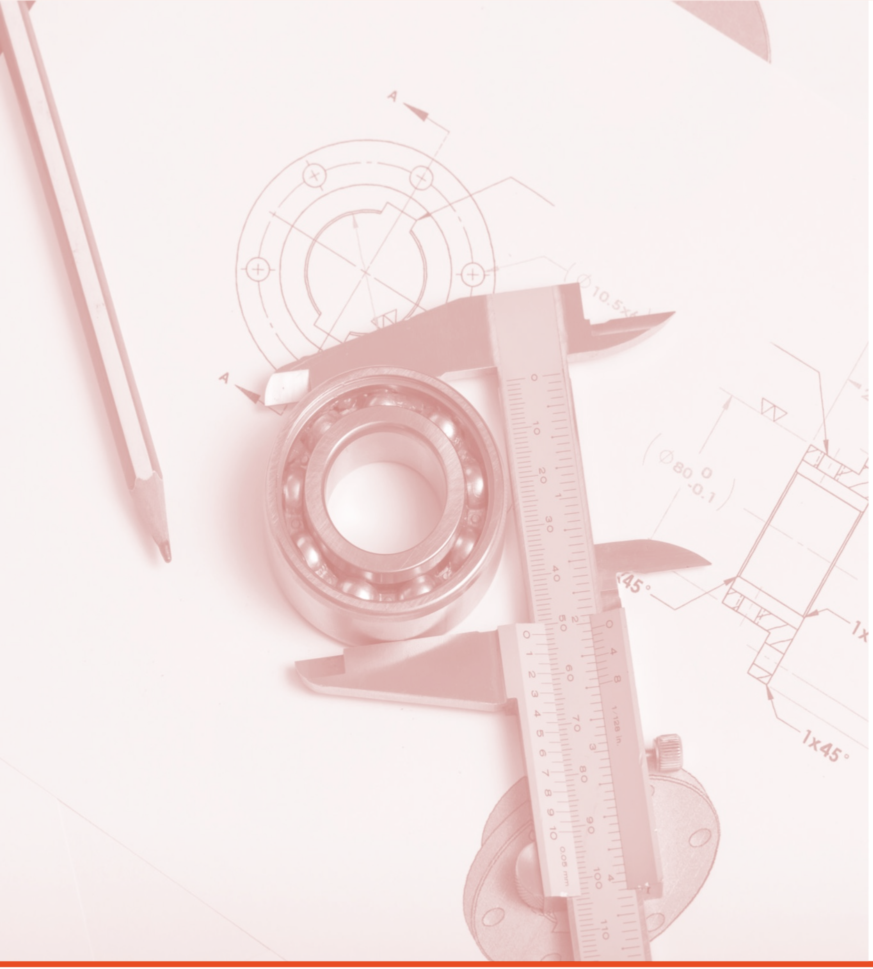
- For a given state:
  - Calculate the shortest time T → system goes to “unsafe” state
- Transfer the control to the safety controller before T

Implementation

- Host Platform:
  - Zedboard (Xilinx's Zynq-7000)
  - FreeRTOS
  - ARM TrustZone (LTZVisor hypervisor)

- Root-of-Trust:
  - MSP430G2452 micro-controller
  - 160-bit internal timer





Evaluation & Results

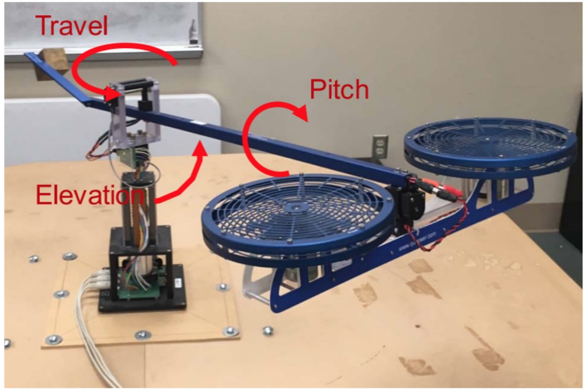
WASHINGTON STATE UNIVERSITY

- Resiliency under attack
- Impact of system dynamics
- Full system vs virtualization-based reboot

Experiment #1: Safety Guarantee

- Testbed: 3 DoF Helicopter

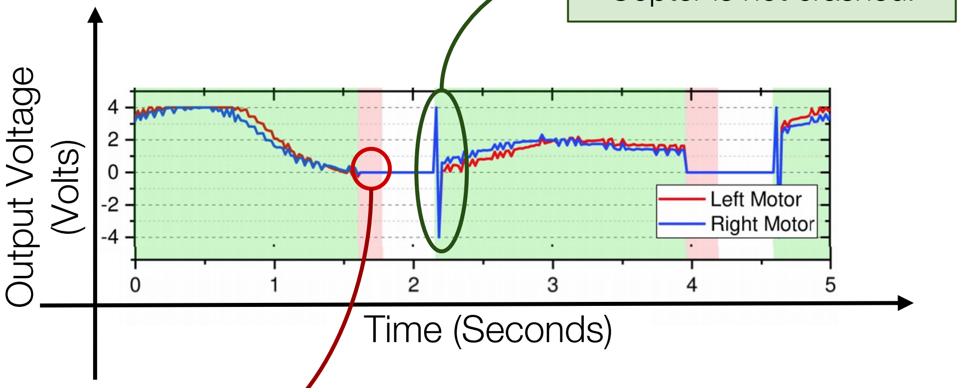
Safety Goal:
not to hit the surface of table



Results

- DoS Attack → turn off motors
  - Extreme case

- Green → Safety controller
- Red → Untrusted controller
- White → Reboot

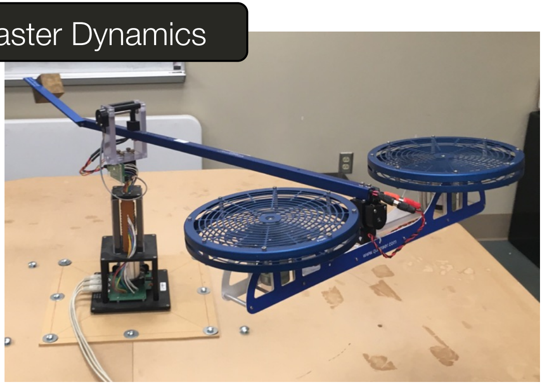


Control recovered
Copter is not crashed!

Motor turned off!

Experiment #2: Reboot vs System Dynamics

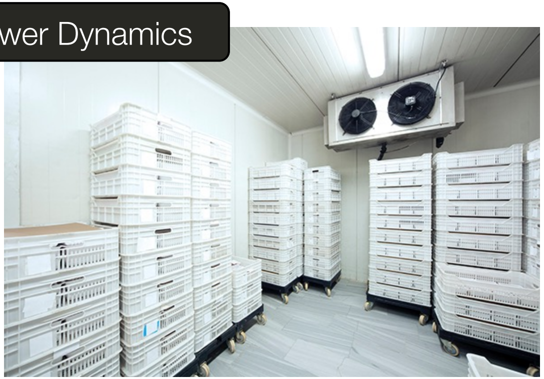
Faster Dynamics



3 Degree of Freedom Helicopter

Safety Goal:
not to hit the surface of table

Slower Dynamics

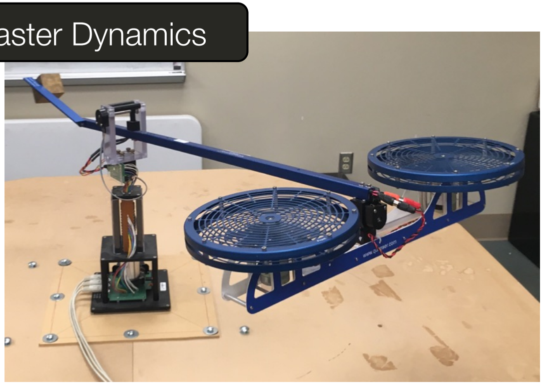


Warehouse Temperature Management
(Hardware in the loop Simulation)

Safety Goal:
keep the temperature [20^0C, 30^0C]

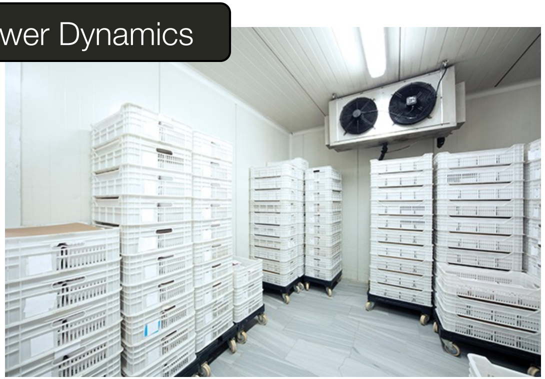
Experiment #2: Reboot vs System Dynamics

Faster Dynamics



Longest safe restart time: 1.23 Seconds
(Full Reboot Time: 390 ms)

Slower Dynamics

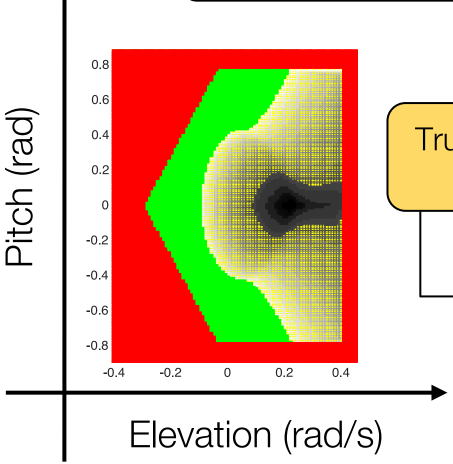


Longest safe restart time: 6235 Seconds
(Full Reboot Time: 390 ms)

Slower Systems → Larger Rebootable Window
More time to react and reboot!

Experiment #3: Full Platform vs TEE-assisted

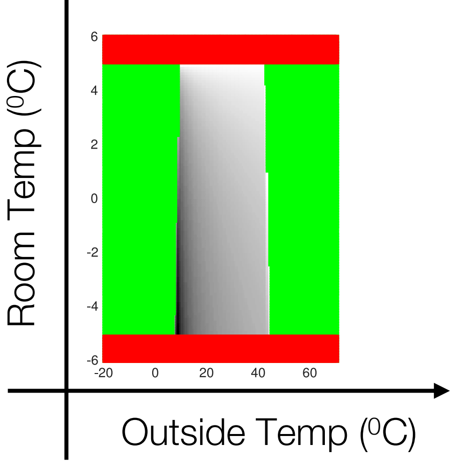
6 DoF Helicopter



TrustZone-assisted implementation → 234% increase of the rebootable region!

Highly depends on platform & system dynamics!

Warehouse Temp Monitor



- Green → Admissible
- Black → Restartable (Both)
- Yellow → Restartable (TrustZone)
- Red → Inadmissible

Length of the safety window >> Platform restart time
Performance improvement is insignificant!

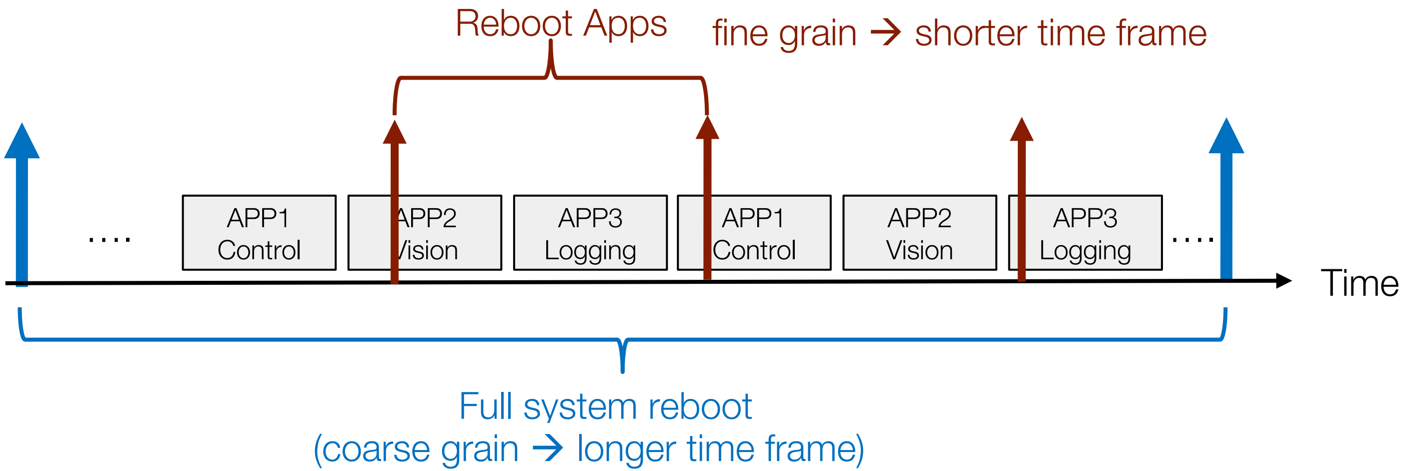
Summary

- Active reboot mechanism can *guarantee* safety
- Usability highly depends on the platform’s boot time & system dynamics
  - It is more suitable for systems with slower dynamics
  - Yet, still usable in systems with fast dynamics (depends on the platform)

Ongoing Work

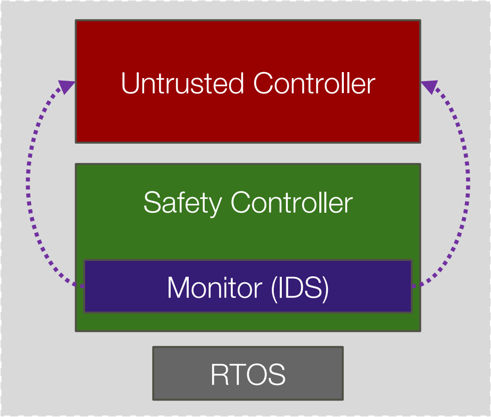
Challenges:
- Reboot frequency?
- Which apps to reboot?
- Temporal constraints?

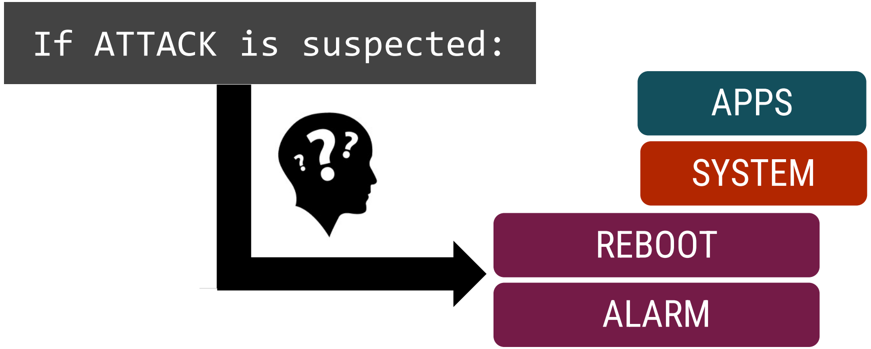
- <u>Proactive</u> → Application-level reboot



Ongoing Work

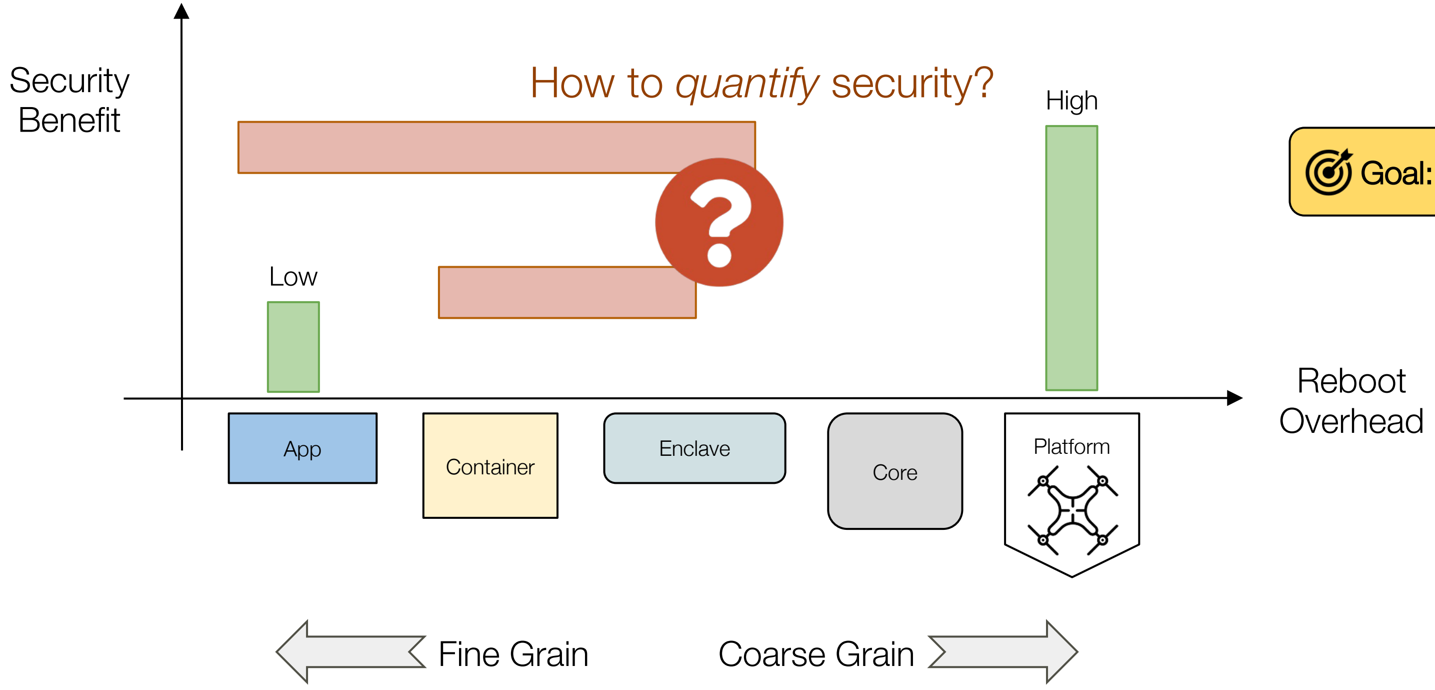
- Proactive & Reactive → Application & System-level reboot





Ongoing Work

Modular *Proactive* & *Reactive* Framework
Cost-Benefit Analysis



🎯 **Goal:** *A "tunable" framework*

Remarks

- Platform reboot: one way to secure critical CPS
  - Ensures physical safety
  - Prevents the attacks from progressing

- Threats to critical systems are increasing
  - Requires layered defense mechanisms

[CPSIoTSec'22] V. Banerjee, S. Hounsinou, H. Olufowobi, M. Hasan and G. Bloom, "Secure Reboots for Real-Time Cyber-Physical Systems," in Proc. of ACM Joint Workshop on CPS & IoT Security and Privacy (CPSIoTSec), Nov. 2022.

[RTSS/BP'21] S. Hounsinou, V. Banerjee, C. Peng, M. Hasan and G. Bloom, "Work-in-Progress: Enabling Secure Boot for Real-Time Restart-based Cyber-Physical systems," in Proc. of IEEE Real-Time Systems Symposium (RTSS), Brief Presentations (BP) track, Dec. 2021.

Today's Talk

[IoT'18] F. Abdi, C. Chen, M. Hasan, S. Liu, S. Mohan and M. Caccamo, "Preserving Physical Safety Under Cyber Attacks," iEEE Internet of Things Journal, Aug. 2019.

Today's Talk

[ICCPS'18] F. Abdi, C. Chen, M. Hasan, S. Liu, S. Mohan and M. Caccamo, "Guaranteed Physical Security with Restart-Based Design for Cyber-Physical Systems," ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), 2018.

Questions?

https://monowarhasan.info/

monowar.hasan@wsu.edu

@mnwrhsn