Trusted CI Webinar Series

Title: Improving the Security of Open-Source Software Infra.

Presenters: Gedare Bloom (University of Colorado Colorado Springs)

Host: Jeannette Dopheide

Slides: https://tinyurl.com/mv47dahe

The meeting will begin shortly.

Participants are muted. Click the chat button to ask a question.

This meeting will be recorded.

The Trusted CI Webinar Series is supported by National Science Foundation grant #2241313.

TRUSTED CI
THE NSF CYBERSECURITY CENTER OF EXCELLENCE

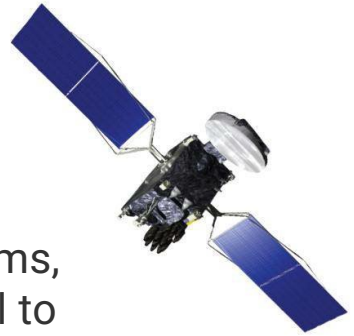
Improving the Security of Open-Source Software Infrastructure

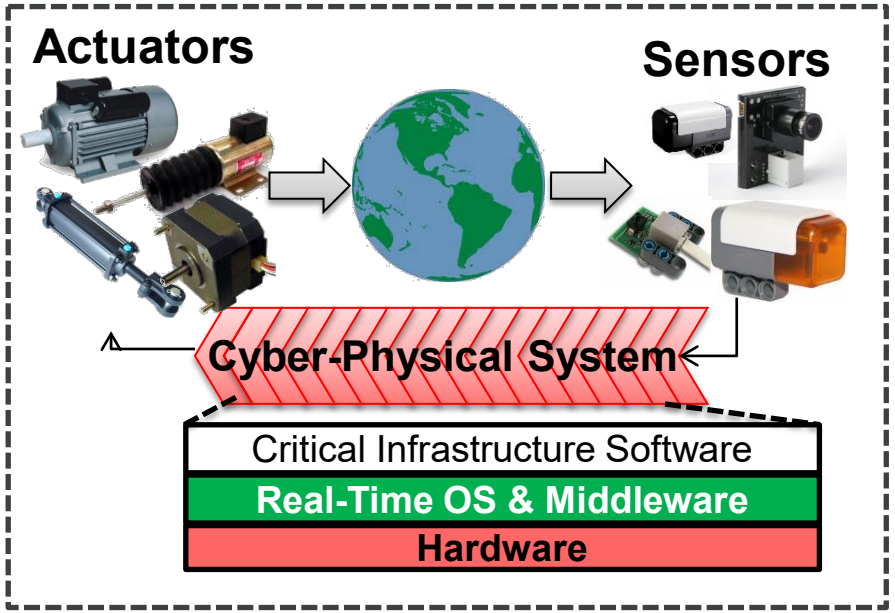
Gedare Bloom, Ph.D.
Computer Science
University of Colorado Colorado Springs

Trusted CI
Webinar
January 23, 2023

Cyber-Physical Systems and Critical Infrastructure

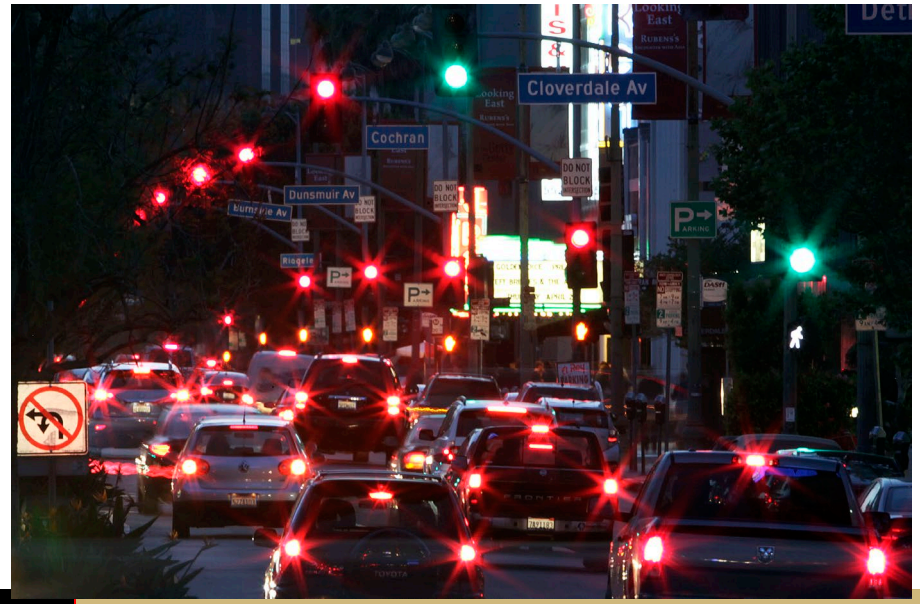
Cyber-physical Systems (CPS) are **engineered systems that are built from, and depend upon, the seamless integration of computation and physical components**.





There are **16 critical infrastructure sectors** whose assets, systems, and networks, whether physical or virtual, are considered so vital to the United States that their incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof.

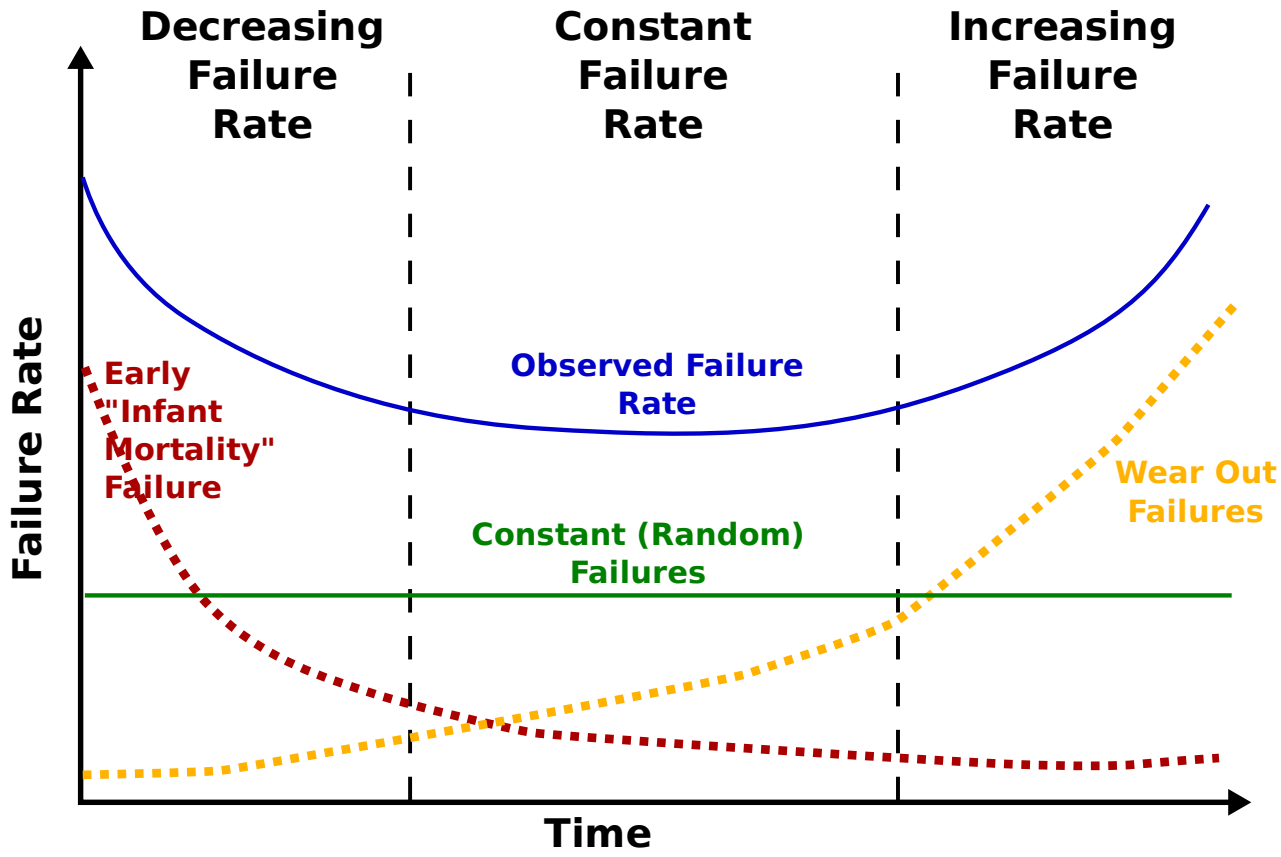
- Defense Industrial Base, Energy, Communications, Information Technology, Transportation



Key Assumption of CPS/Infrastructure is that Reliability Grows!

We expect systems to become more reliable as we learn more about how to manufacture them and train people to use them.

- Exponential growth
- Bathtub curve



Security **decays** over time

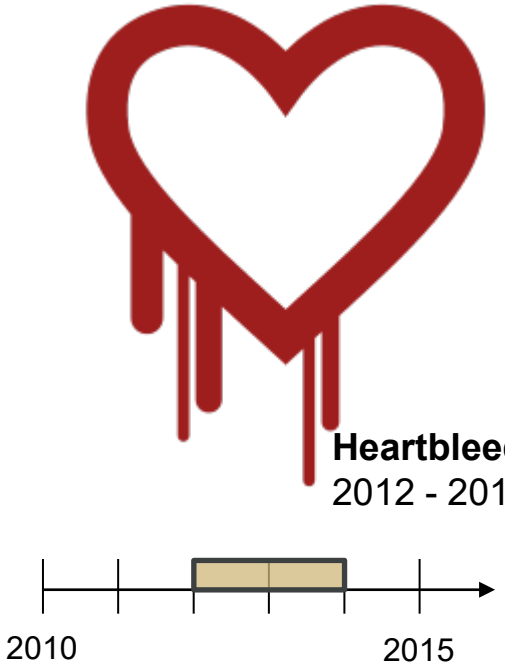
- Systems in use for a long time
- A lot of opportunities to find vulnerabilities

Source: https://goo.gl/JUBLmd

Security **decays** over time

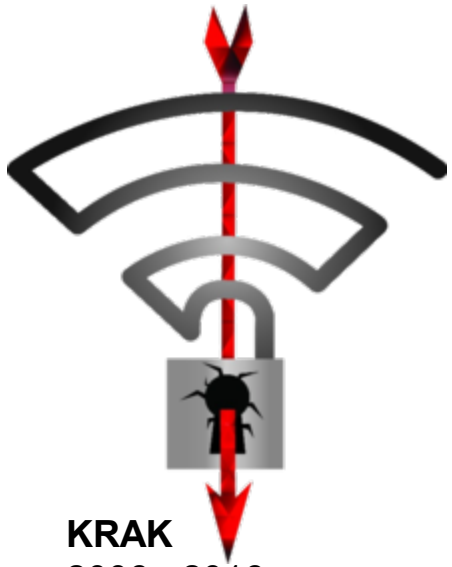
- Systems in use for a long time
- A lot of opportunities to find vulnerabilities

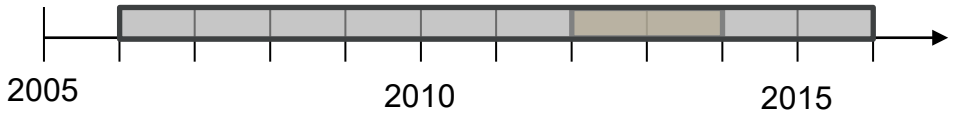
Source: https://goo.gl/JUBLmd



Security **decays** over time

- Systems in use for a long time
- A lot of opportunities to find vulnerabilities





Source: https://goo.gl/JUBLmd

Security **decays** over time

- Systems in use for a long time
- A lot of opportunities to find vulnerabilities

Source: https://goo.gl/JUBLmd

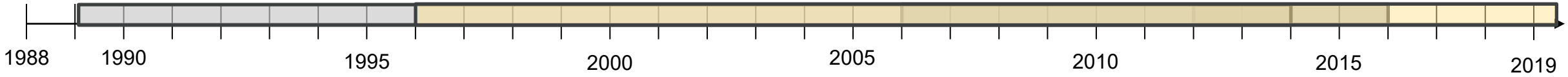
Shellshock
1989 – 2014

Security **decays** over time

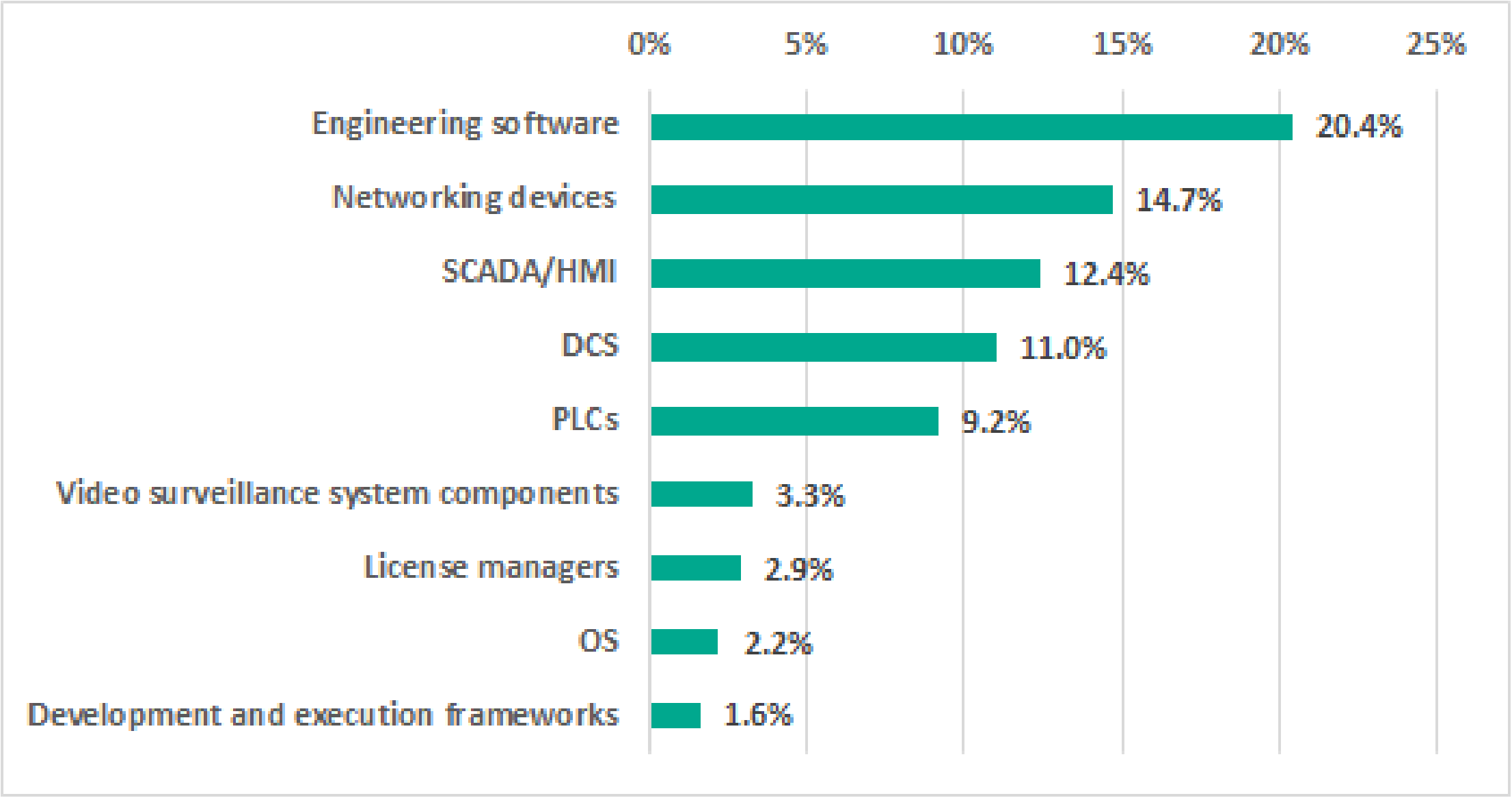
- Systems in use for a long time
- A lot of opportunities to find vulnerabilities

Source: https://goo.gl/JUBLmd

Meltdown
1996 – 2019

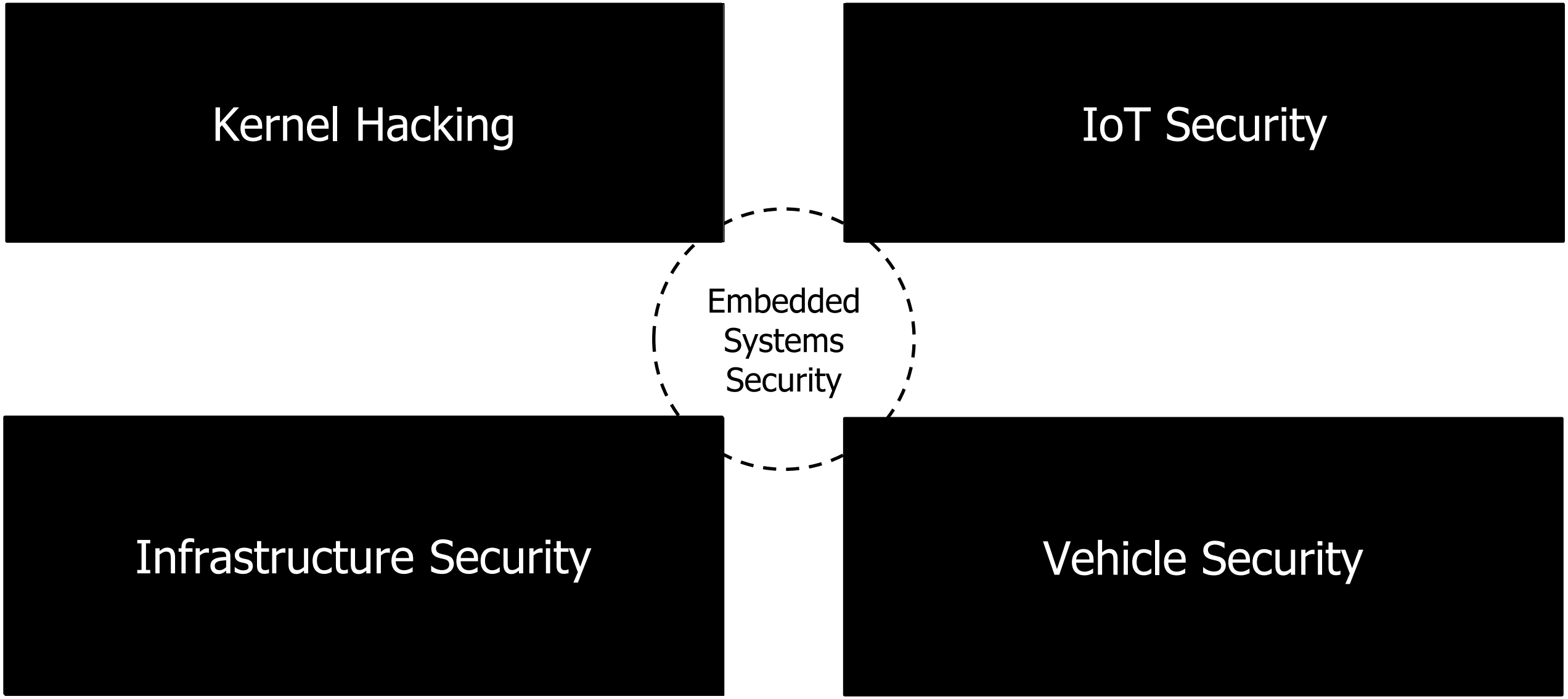


Industrial Control System Vulnerabilities (ca. 2019)



Source https://ics-cert.kaspersky.com/reports/2020/04/24/threat-landscape-for-industrial-automation-systems-vulnerabilities-identified-in-2019/

My Research


Kernel Hacking
IoT Security
Embedded Systems Security
Infrastructure Security
Vehicle Security

Security Hardening for Scientific Industrial Control Systems

NSF OAC-1839321, OAC-2001789: 10/01/2018-09/30/2023

Key Contributions

Adoption of secure software development life cycle (SDLC) in open-source communities for industrial control.

Increase confidence in fidelity of scientific data collection.

Reduce risks associated with misconfiguration in untrustworthy environments.

UCCS University of Colorado Colorado Springs

Fritz-Haber-Institut

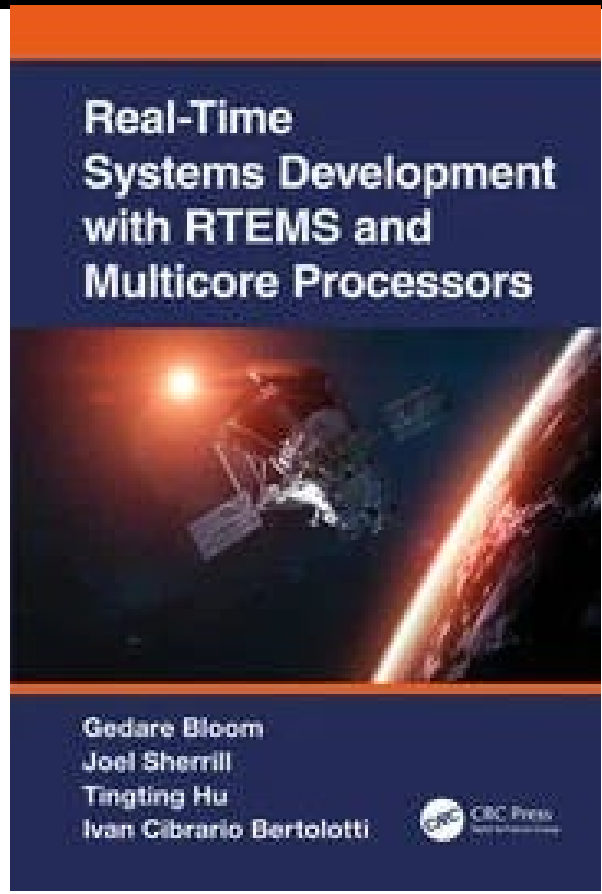
OAK RIDGE National Laboratory

Argonne NATIONAL LABORATORY

RTEMS Open-Source Real-Time Operating System

- One of 6 Maintainers
  - Port to first 64-bit architecture
  - Rewrote thread scheduler
  - Paravirtualization framework
  - Textbook
- Mentoring students
- Future: verification, security

rtems
Real-Time Executive for Multiprocessor Systems



NASA U.S.ARMY esa European Space Agency AIRBUS Advanced Photon Source Argonne National Laboratory JOHNS HOPKINS APPLIED PHYSICS LABORATORY OAK RIDGE National Laboratory EPICS BMW

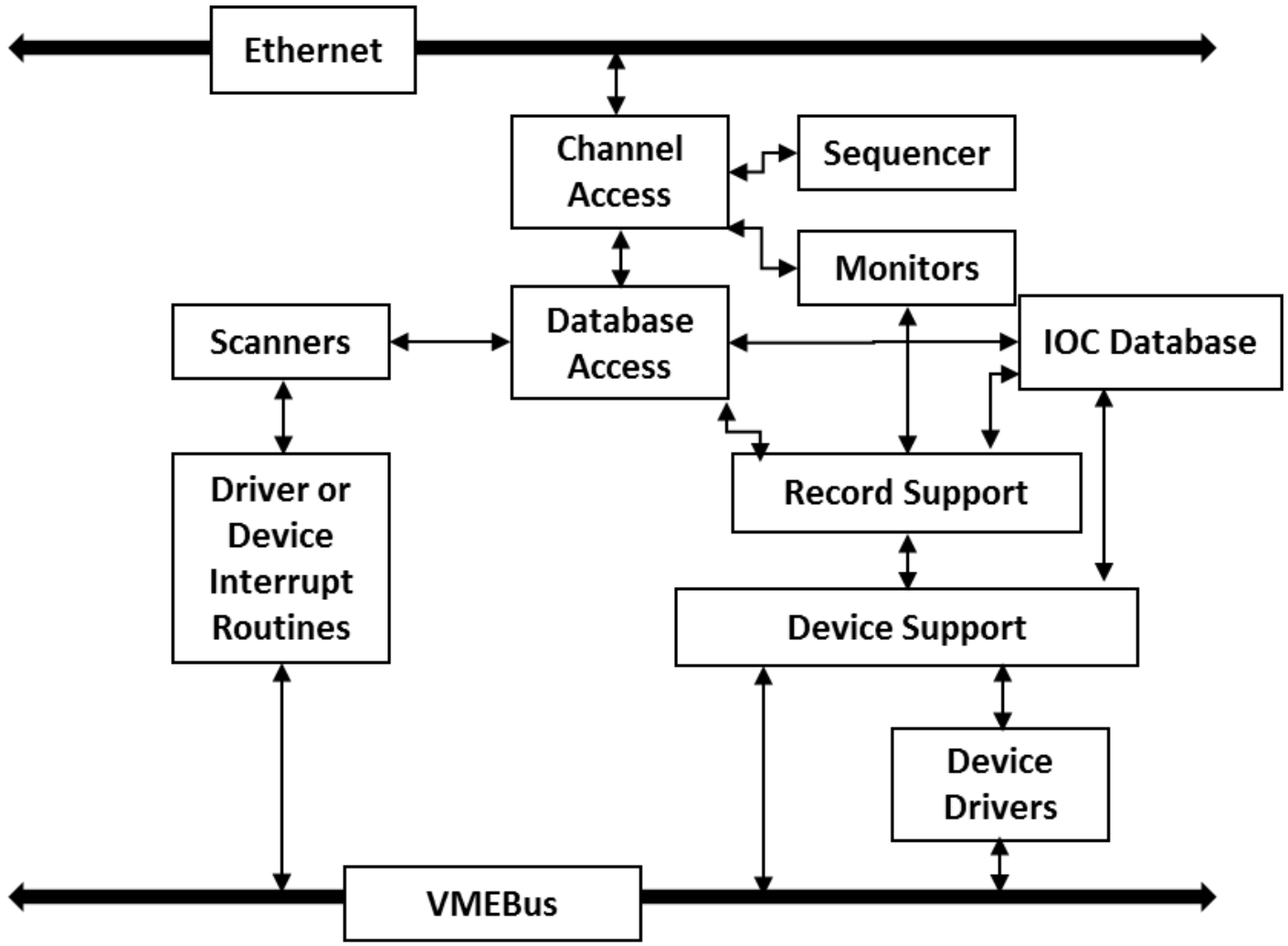
Google Summer of Code Google Code-in COBHAM LOYTEC electronics GmbH MITRE

EPICS: Experimental Physics Industrial Control System

- Open-source ICS software: Operating Systems, Device Drivers, Network Protocols, Logging, Monitoring
- High-energy physics and astronomy
- Not designed with security in mind

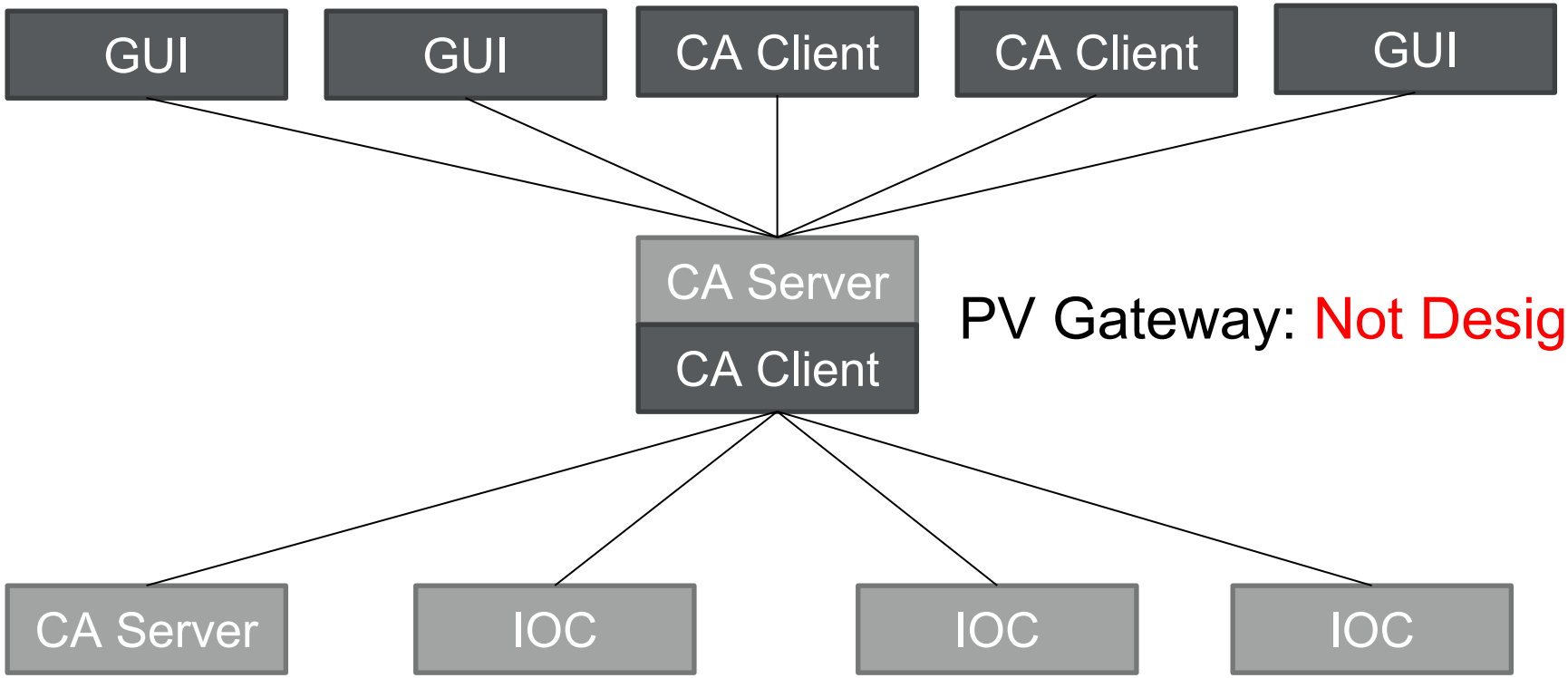
EPICS Basic Building Block: I/O Controller (IOC)

- IOC encapsulates a compute node
- Runs an OS
- Manages a database of process variables (PVs)
- Communicates with other IOCs through the EPICS Channel Access Protocol
- Inherently trusts the Channel Access and VMEBus networks



EPICS PV Gateway

- Optional widely-used PV Gateway software can manage and multiplex connections between operators and IOCs



PV Gateway: Not Designed for Security

Unprotected from attackers (and interns)

- **C/C++ will never (!) be secure**

- May be capable of injecting bugs in EPICS software products
- May be capable of launching DOS/DDOS against OPI/PV Gateway/CA networks
- May be capable of subverting data quality and provenance of PV data

Goal: provide resiliency for EPICS throughout software development lifecycle

Security Hardening for EPICS: Research Plan

- Three-year project plan started 10/1/2018
  - 2 Graduate Students, 1 Postdoc, 1 PI

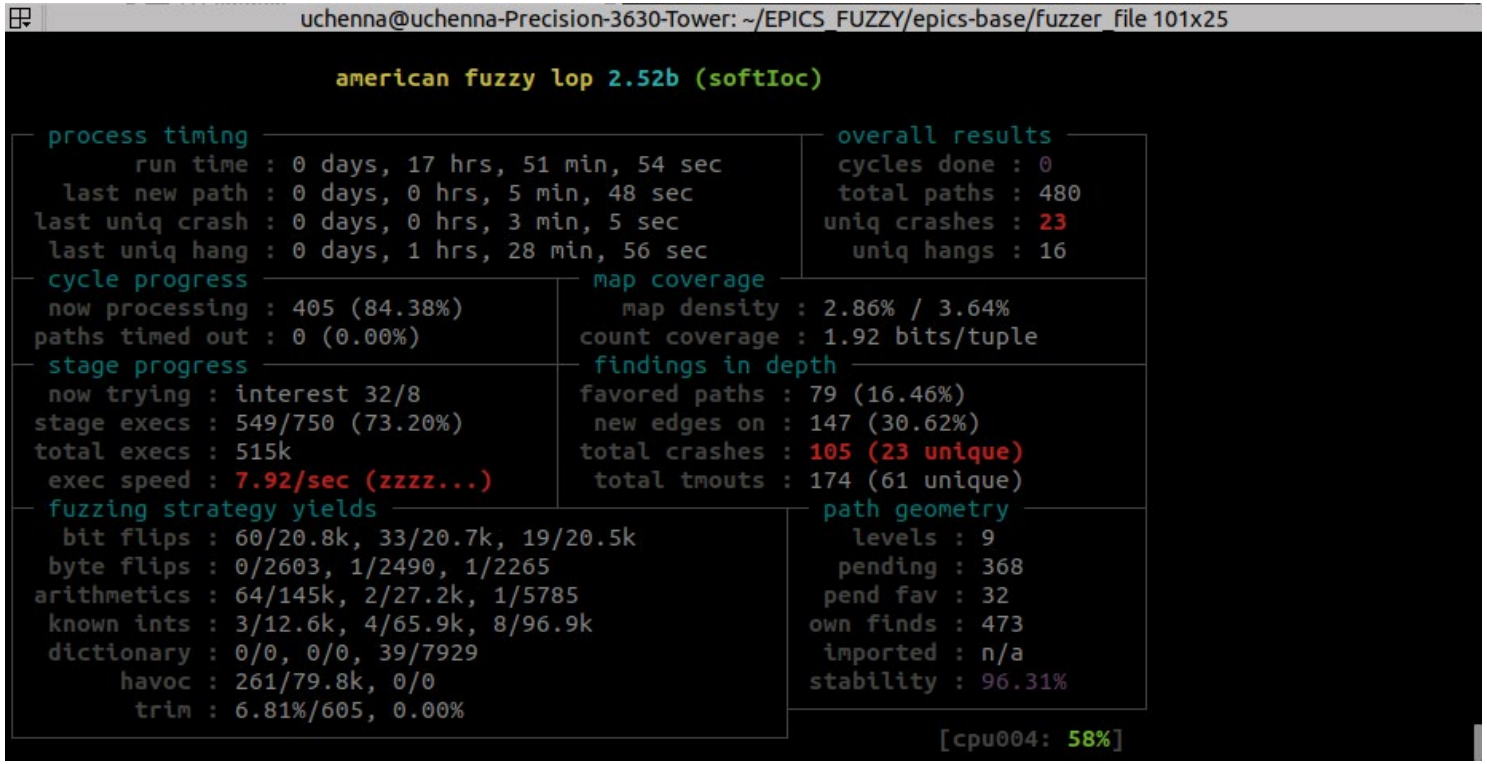
	Year 1				Year 2				Year 3			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Aim 1: Security Throughout the Software Development Life Cycle.												
Task 1.1: Apply Static Analysis to EPICS.												
Task 1.2: Apply Security Fuzz Testing to EPICS.												
Task 1.3: Integrity Protection of EPICS Software Products.												
Task 1.4: Secure Boot and Update.												
Aim 2: Enhancing and Leveraging Operating System Security Services.												
Task 2.1: Establish Common Cryptographic Libraries for EPICS.												
Task 2.2: Port Secure Communication tools to IOC OSs.												
Task 2.3: Add Memory Protection to IOC OS layers.												
Aim 3: Analyzing and Improving Network Security for EPICS Protocols.												
Task 3.1: Formally Model and Analyze EPICS PV Gateway.												
Task 3.2: Enhance Security Logging of EPICS and PV Gateway.												
Task 3.3: Add Network IDS to EPICS PV Gateway.												

Static Analysis and Fuzz Testing

- Apply Static Analysis to EPICS
  - Limited success to date. Challenges remain in automatic triage.
  - Scanning selected software components with Coverity, Codiga, Clang
- Apply Security Fuzz Testing to EPICS
  - Active and Ongoing Work.
  - AFL Fuzzing and PhD Dissertation Research on ICS/OT fuzzing

Improve Fuzzing with Static Analysis

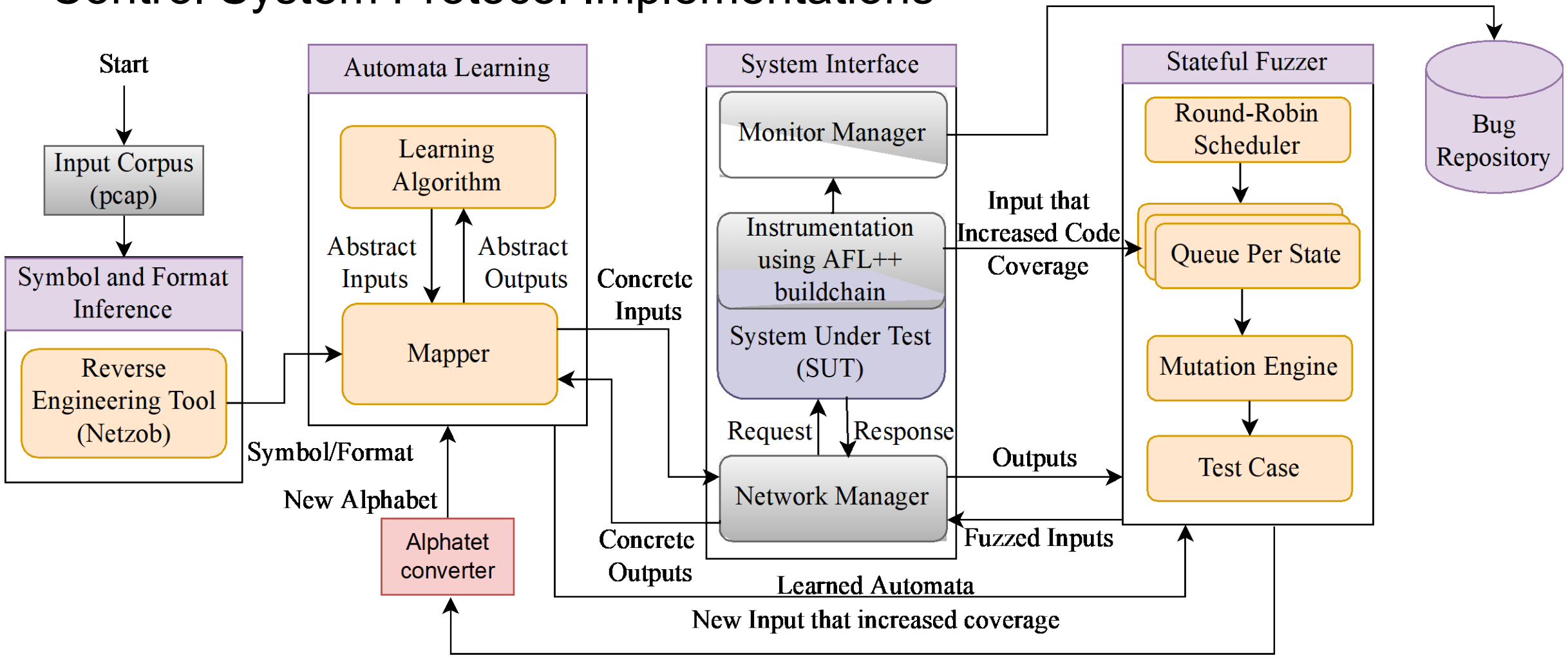
- Fuzzing base software modules of EPICS using out-of-the-box AFL setup
  - Quick "Win" by an REU Student



AFL running on softIoc

Current Research (Work-in-Progress): MICFICS

- MICFICS: Model Inference Coverage-Guided Fuzzing for Industrial Control System Protocol Implementations



Current Research: Fuzzing EPICS stateful protocols

- Application of MICFICS to Portable Channel Access Server
  - Fuzz EPICS Portable Channel access server
    - Using sample collected pcap file from listening to communication between server and client
    or
    - Manually crafting input symbols from known communication commands
  - Fuzz EPICS pvAccess server (TCP based) example implementation

- Fuzz IoC initialization module with and without access security module.
  - This module has multiple states in the “initHookState”

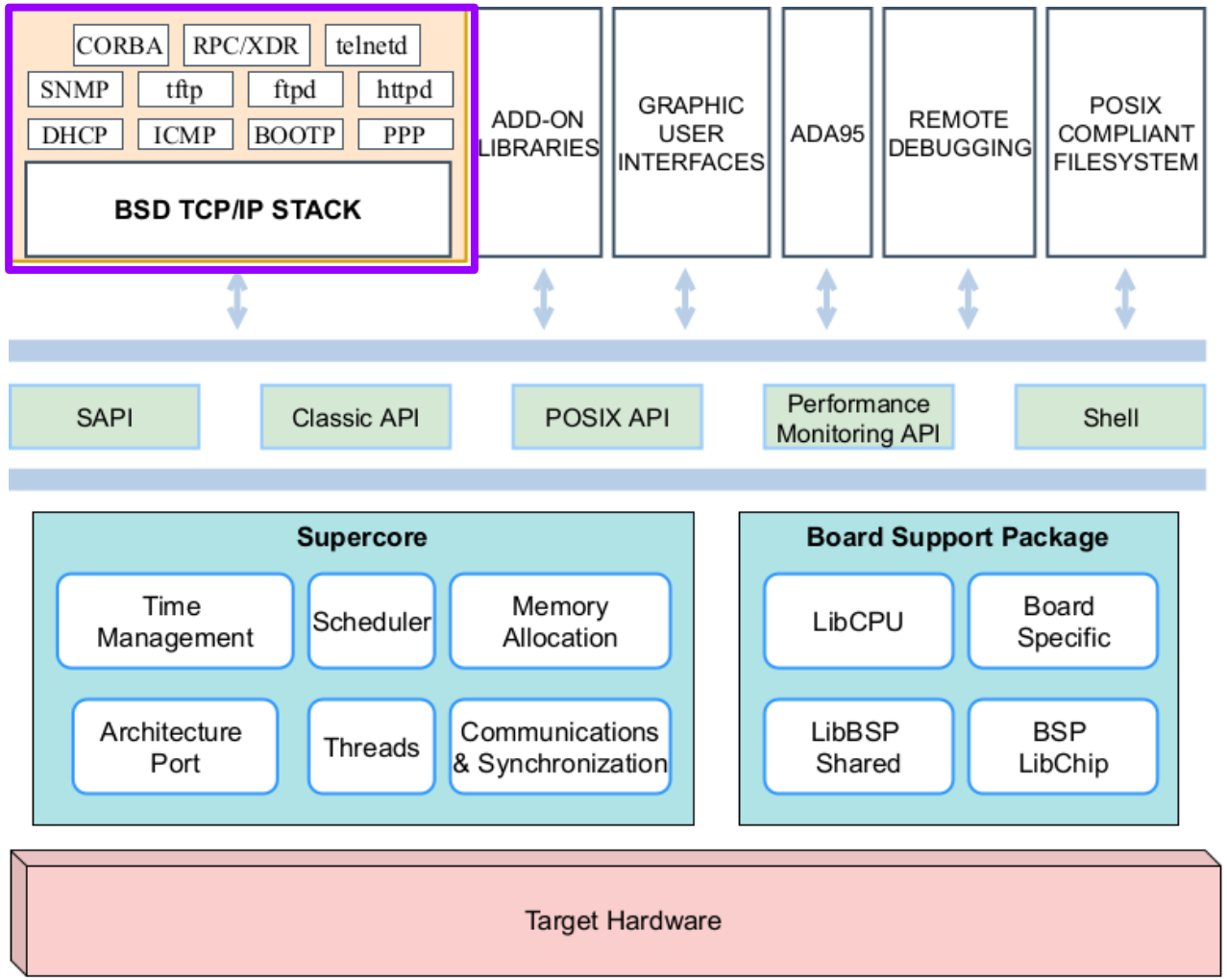
Enhancing and Leveraging OS Security Services

- Establish Common Cryptographic Libraries for EPICS
- Port Secure Communication Tools to IOC Operating Systems

- V. Banerjee, S. Hounsinou, H. Gerber, and G. Bloom, Modular Network Stacks in the Real-Time Executive for Multiprocessor Systems, in 2021 Resilience Week (RWS), Oct. 2021, pp. 1–7.

RTEMS High-level Architecture

- The network stack implementation is generally a part of the OS kernel.
- In RTEMS, the network implementation is derived from an old BSD network stack.
  - This default network implementation is now called the **legacy stack.**

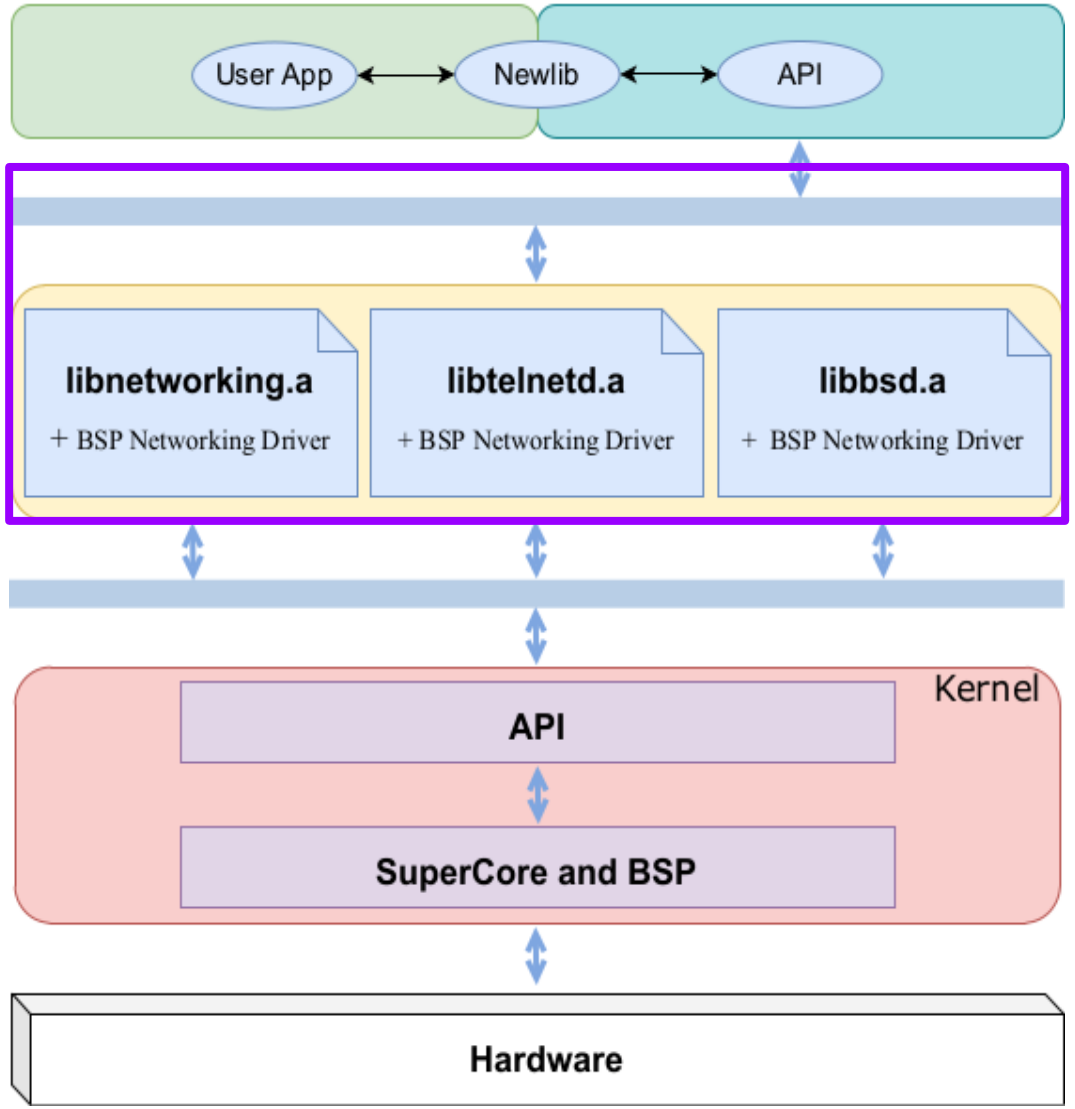


Problems with the Legacy Stack

- Difficult to update the network stack.
- Legacy stack is a derivative of an old BSD stack from the late 90s and it doesn’t support IPv6.
- The legacy stack does not provide modern networking features.
- USA Govt. Memo #M-21-07: Phase out IPv4 and transition to IPv6

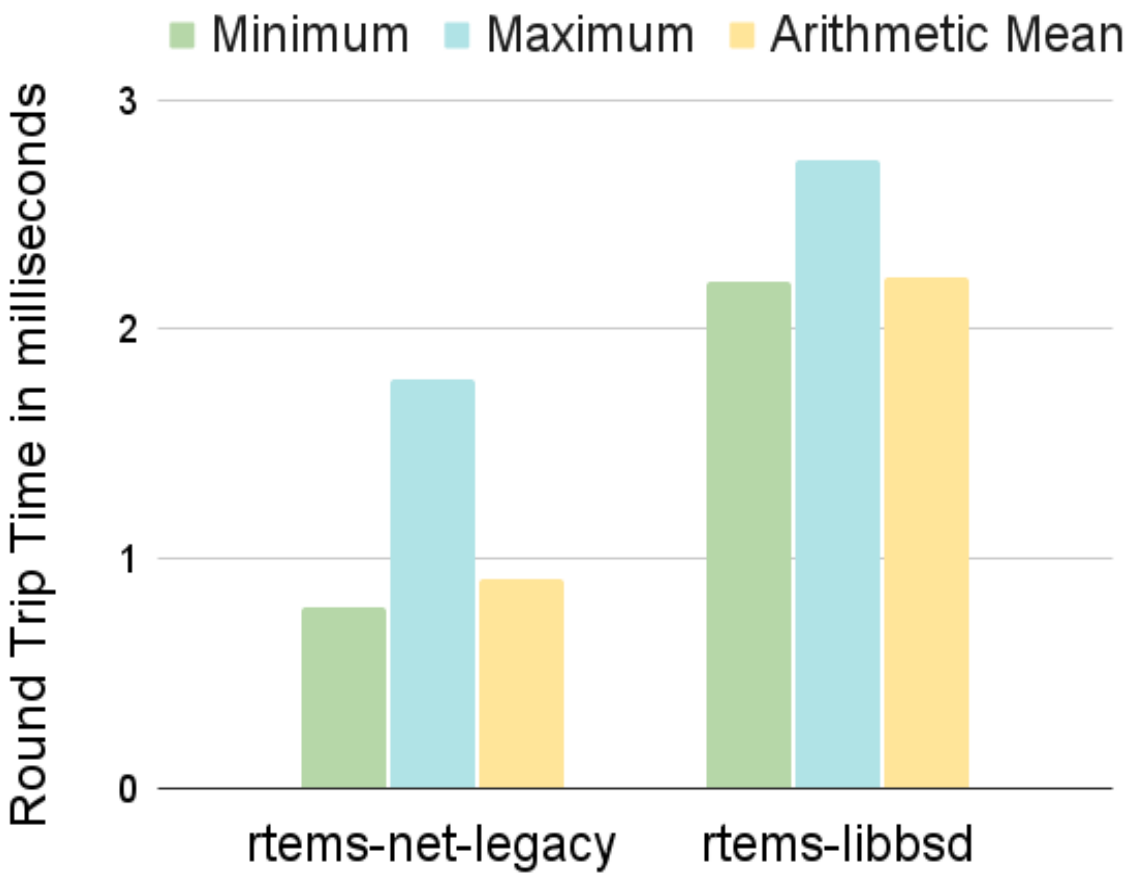
Solution: *Networking-as-a-Library*

- Networking stack is built into a static library
- Makes the OS more adaptable to different user requirements.
- RTEMS is the first hard RTOS to have the networking stack separate from the kernel in form of modular static libraries.



Comparing Networking Libraries

Round-Trip Time Analysis: the rtems-libbsd has a higher a performance impact due to higher latency



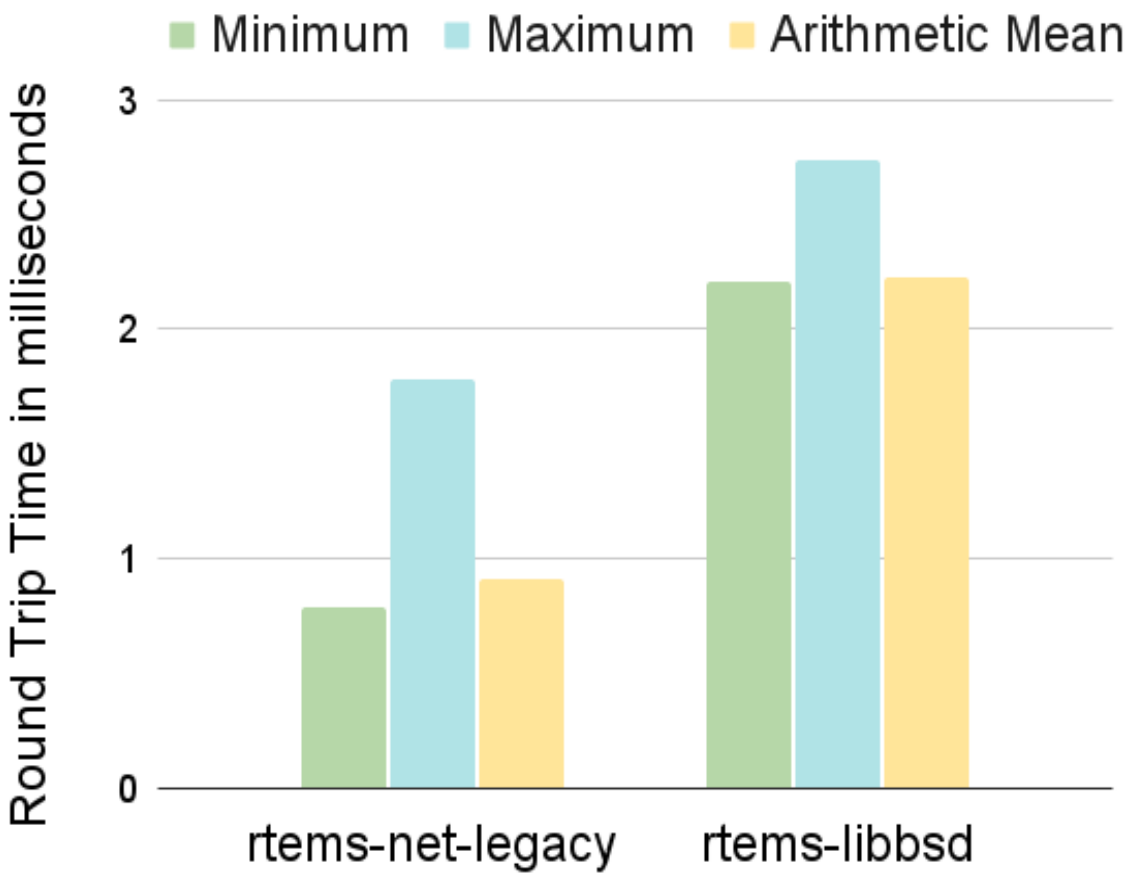
- We need a lightweight network stack.

Network stack	.text	.data	.bss	Total Size
rtems-libbsd	1273	58.4	24	**1332**
rtems-net-legacy	244.4	6	44	**250.5**

Memory Footprint Comparison

Comparing Networking Libraries

Round-Trip Time Analysis: the rtems-libbsd has a higher a performance impact due to higher latency



- We need a lightweight network stack.
- lwIP is promising for memory constrained devices.

Network stack	.text	.data	.bss	Total Size
rtems-libbsd	1273	58.4	24	**1332**
rtems-net-legacy	244.4	6	44	**250.5**
rtems-lwip	293	1.7	59	**294**

Memory Footprint Comparison

Summary of Refactoring Effort

- Major rework of the networking support in RTEMS for EPICS.
  - Not initially planned!
- Pulled the aging network infrastructure out of RTEMS into a separate repository for deprecation.
  - Migration of about 275,000 source lines of code out of RTEMS
  - Simplifies the process to upgrade the networking infrastructure for EPICS to enable access to more secure, state-of-the-art networking protocols and services including SSL and SSH (as planned)
- Establishment of an official rtems-lwip.git repository containing infrastructure for the lwIP networking stack with RTEMS.
  - https://git.rtems.org/rtems-lwip/ a new "top-level" repository for public use.
  - Other people are now using and contributing to this repository.

Software Infrastructure Cybersecurity Publications

[ISORC] P. Dangal and G. Bloom, Towards Industrial Security Through Real-time Analytics, ISORC, 2020.

[RWS] V. Banerjee, S. Hounsinou, H. Gerber, and G. Bloom, Modular Network Stacks in the Real-Time Executive for Multiprocessor Systems, in 2021 Resilience Week (RWS), Oct. 2021, pp. 1–7.

[RTSS-WIP] 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 2021 IEEE Real-Time Systems Symposium (RTSS), Dec. 2021, pp. 524–527.

[CCNC] H. Lawrence, U. Ezeobi, G. Bloom, Y. Zhuang, Shining New Light on Useful Features for Network Intrusion Detection Algorithms, in 2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC), Jan. 2022, pp. 369-377.

[CPSIoTSec] V. Banerjee, S. Hounsinou, H. Olufowobi, M. Hasan, and G. Bloom. Secure Reboots for Real-Time Cyber-Physical Systems. In Proceedings of the 4th Workshop on CPS & IoT Security and Privacy (CPSIoTSec'22), Nov. 2022, pp. 27–33.

Working with Open-Source Communities: Some Lessons Learned

- Engage the community early and often
  - As a maintainer of RTEMS I had credibility with EPICS community
  - Get involved with the community first!
- People react very differently about security
  - Topics often got political. Some people don't care at all, and some care a lot.
  - Just because you can and will do the work, they might not want you to!
- Amorphous R&D plans: community needs shift during project period
- Choose assessment mechanisms carefully
  - Initially planned to do surveys and other IRB-approved studies on the community
    - Quickly discarded that plan for several reasons: trust, validity
  - Focus instead on publicly verifiable assessment data

More Lessons Learned: Short-Term Pains for Long-Term Gains

- Need to align novel research with community/infrastructure needs
  - Many problems facing open-source communities are already solved elsewhere
    - May have to be creative to cast the work done by students as novel
      - Examples: Modular Networking, Secure (Re)boots
    - Find opportunities to bring known solutions to open-source while solving novel problems
      - Examples: Fuzzing ICS
- Don’t assign non-publishable work to (graduate) students
  - I initially wanted all my students to get the experience of being engaged open-source developers. It isn’t realistic. Let students take it on voluntarily.
  - Know what is/isn’t publishable.
    - Non-publishable work: Might have to pay a professional or do it yourself

Embedded Systems Security Lab (ESSL @ UCCS)

Lab Director

Gedare Bloom, Ph.D.
Associate Professor

Lab Supervisor

Sena Hounsinou, Ph.D.
On the Job Market!

Affiliated Alumni

Habeeb Olufowobi, Ph.D.
Assistant Professor
University of Texas Arlington

Ebelechukwu Nwafor, Ph.D.
Assistant Professor
Villanova University

Ph.D. Students

Vijay Banerjee

Uchenna Ezeobi

Doug Healy

Bobby Eimer
Rodney Jones
Farhad Mofidi
Zainab Olalekan*
Minhajul Alam Rahat*
Jordan Scott
Joshua Seaton*

Omolade Ikumapayi

Constance Hendrix

Katrina Rosemond

This work is supported by NSF CNS-2011620, NSF OAC-2001789, NSF-2046705, NSA H98230-21-1-0155 and Colorado State Bill 18-086. The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of any other person or organization.

Questions?

Click on the chat icon to type a question

TRUSTED CI
THE NSF CYBERSECURITY CENTER OF EXCELLENCE

Community Updates

- Next Webinar: February 27th @ 10am Central
  - Topic: Security Program for the NIH's Common Fund Data Ecosystem
  - Presenter: Rick Wagner (UCSD)
- OmniSOC Con 2023, February 21-22 (virtual)
  - **https://omnisoc.iu.edu/events/omnisoc-con**
- EDUCAUSE CPPC, May 1–3 (Bellevue, WA)
  - Call for proposals ends today, registration opens in March
  - **https://events.educause.edu/cybersecurity-and-privacy-professionals-conference/2023**

About the Trusted CI Webinar series

To view presentations, join the announcements mailing list, or submit requests to present, visit: **trustedci.org/webinars** or email **webinars@trustedci.org**

The Trusted CI Webinar Series is supported by National Science Foundation grant #2241313.

TRUSTED CI
THE NSF CYBERSECURITY CENTER OF EXCELLENCE