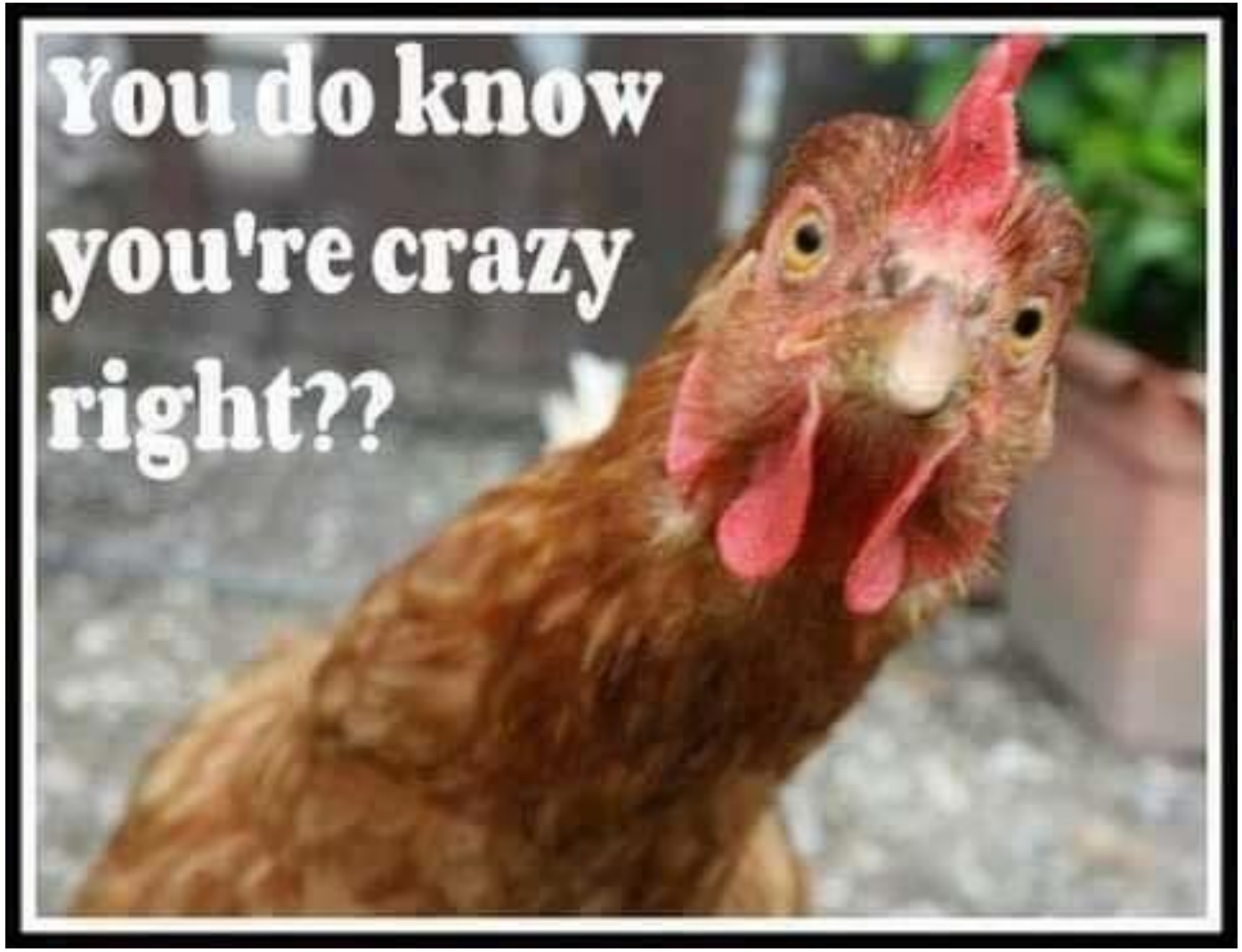
Security Hardening for EPICS/RTEMS

GEDARE BLOOM

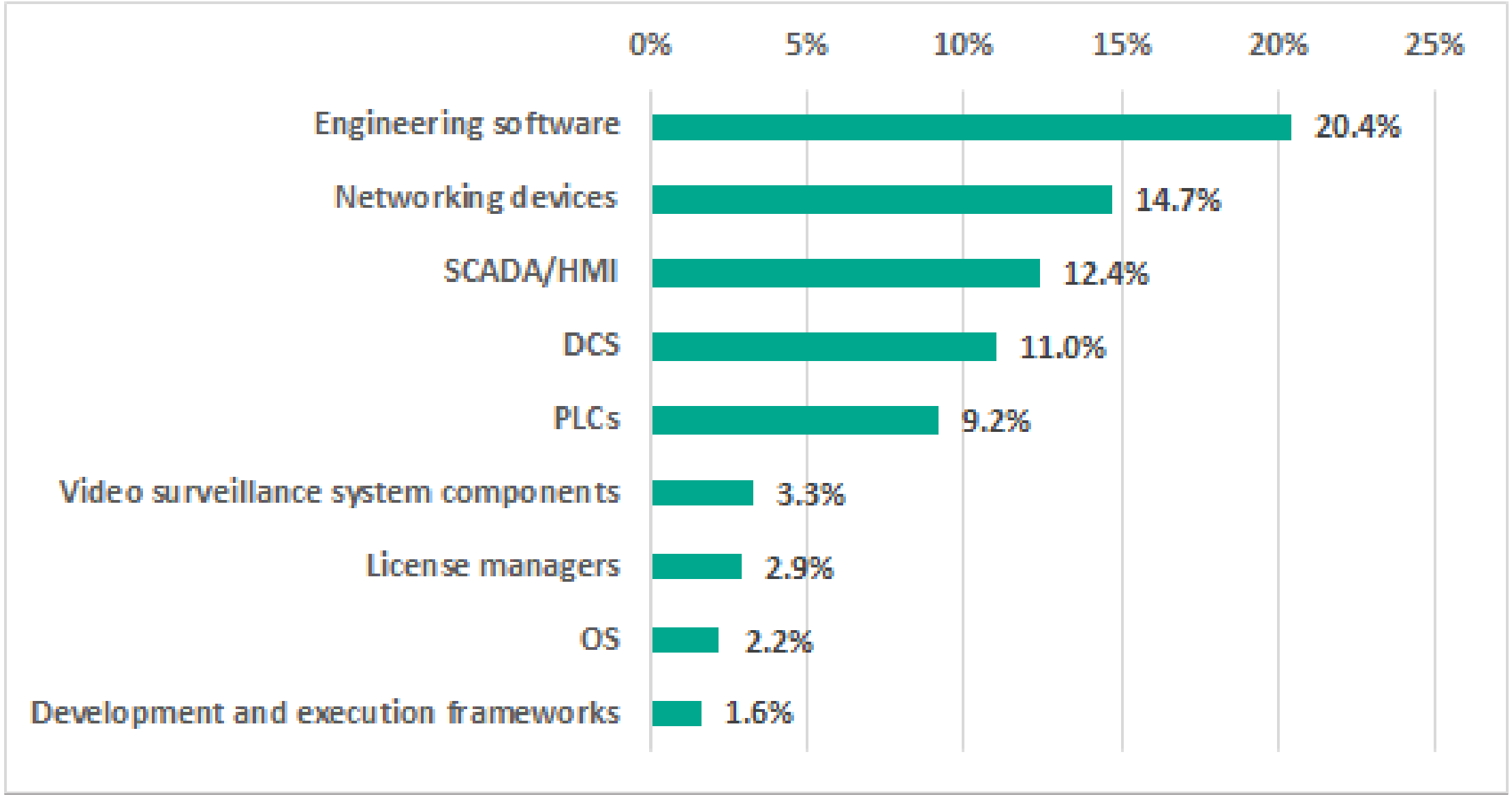
UNIVERSITY OF COLORADO COLORADO SPRINGS

EPICS COLLABORATION FALL 2020 MEETING

EPICS is insecure



EPICS is insecure?



Industrial Control System Vulnerabilities

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

EPICS is not protected against sophisticated attackers and unsophisticated interns

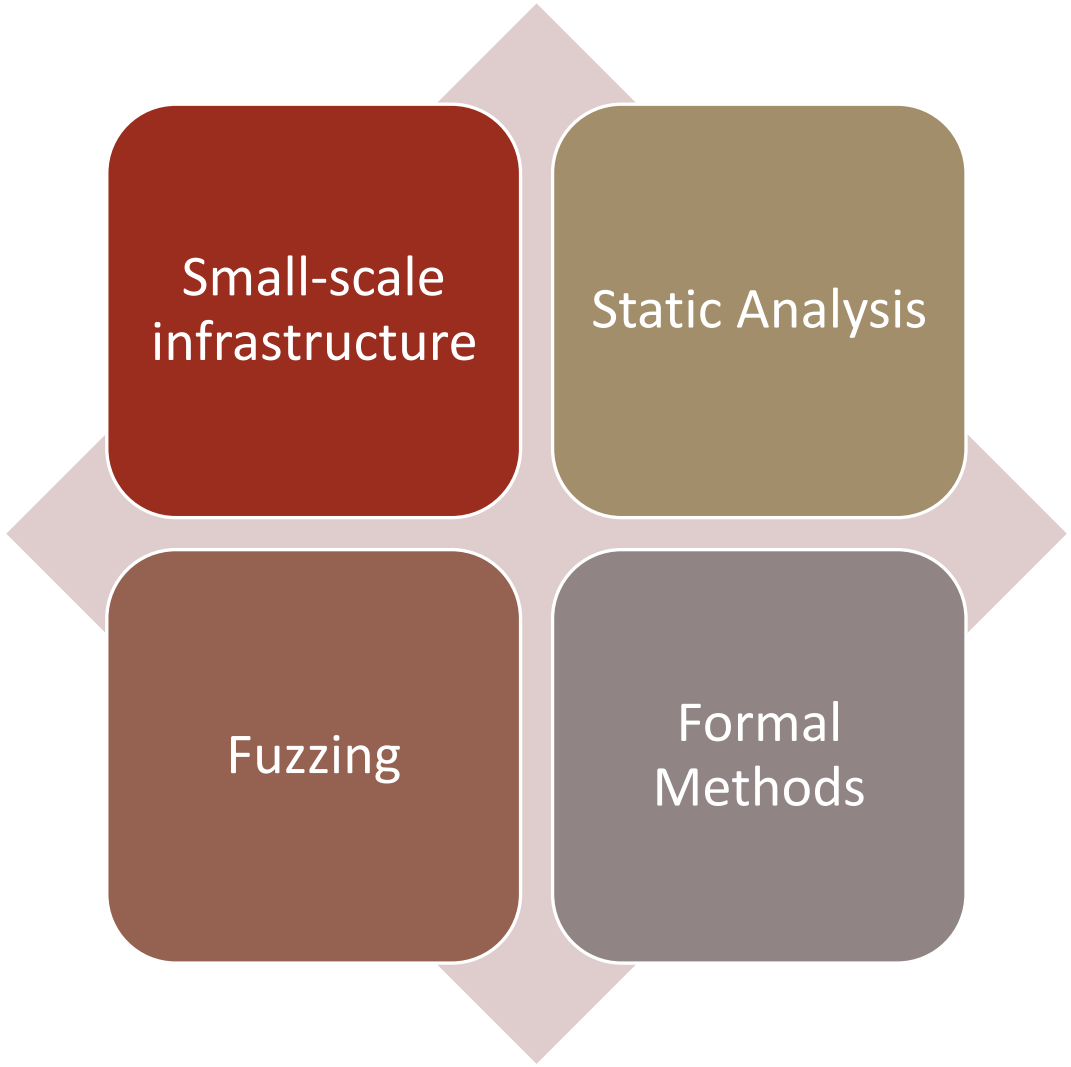
C/C++ will never (!) be secure

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 Testing

Small-Scale Research Infrastructure

Problem: Want to analyze security problems and solutions with *lean* supplies budget

- Academic lean budget: Supply costs around $1-2K

Goals:

- Run “realistic” PV/asyn workloads
- Evaluate runtime/security performance of modifications to EPICS (IOCs: RTEMS, Linux)
- Student setup and maintenance

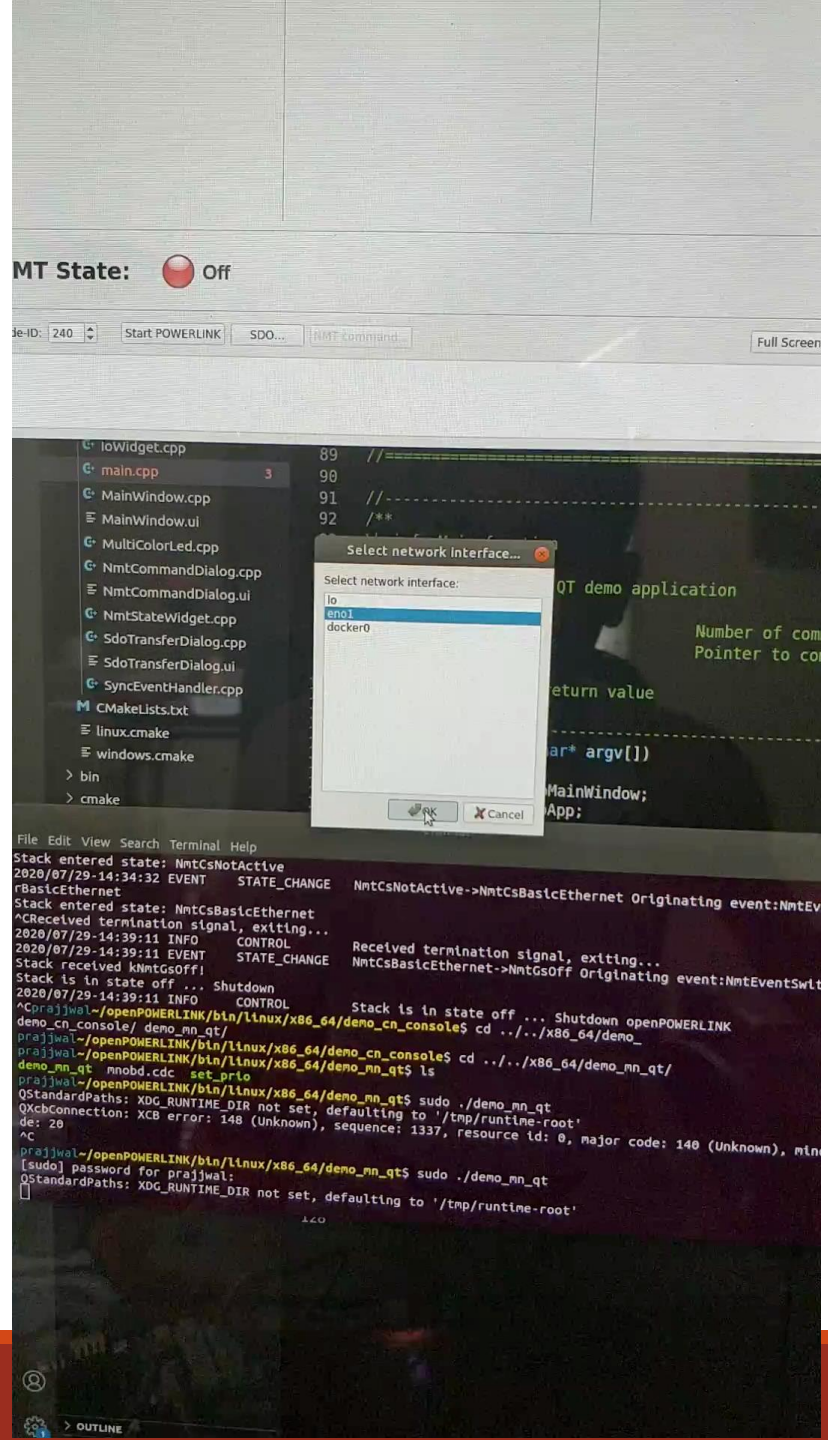
Real-time Analytics with Open Powerlink

Investigating system delay, network delay, actuation delay

- System delay: background tasks, user / kernel space
- Propagation delay: interference in wire
- Actuation delay: servo response time; screw, motor

Real-time analytics in an EPICS environment

- IOCs → OPI → On-Site Private Cloud (OSPC)
- Extension of our baremetal PowerLink stack
  - Video → PowerLink management node polling controlled node



Prajjwal Dangal, Gedare Bloom, Towards Industrial Security Through Real-time Analytics, in 2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC), pp. 156-157, 2020.

EPICS Simulation package with utilities for data acquisition, exporting, noise setting [2]

Ophyd provides EpicsMotors, Signal classes backed by PVs

Databroker, event driven

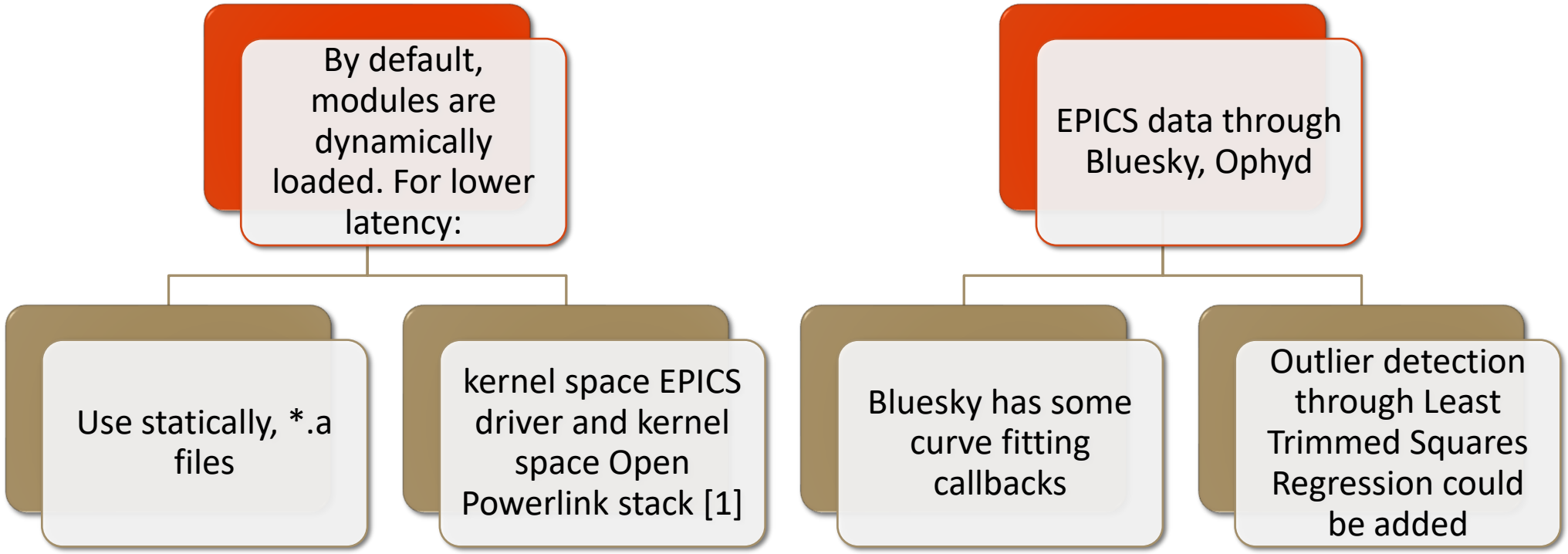
No physical test-bed required

Easy to reconfigure, update

Bluesky testbed-free infrastructure

https://blueskyproject.io/

Current Approach with PowerLink



Bluesky, “bluesky/tutorials,” https://github.com/bluesky/tutorials

beamlineconfiguration.py is the king ☺

Next Steps for Research Infrastructure

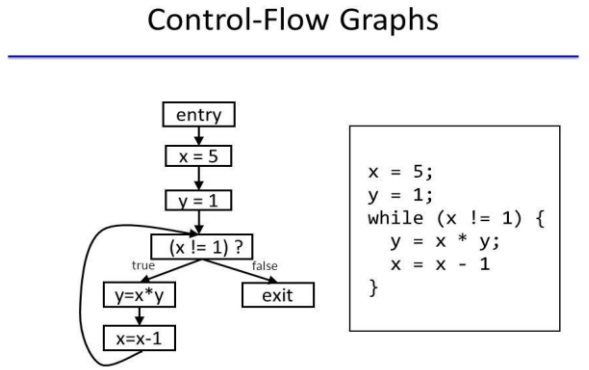
Using Javascript for Bluesky's data model backend as it uses documents

Port data analysis packages related to linear, logistic regression, etc.

Working with EPICS and RTEMS for real-time analytics

Decorate, modify Bluesky code through Numba, and Numpy

Static Analysis



- ✔ Analyze the software without executing
- x Cannot find runtime and configuration errors
- x Cannot test EPICS network

First approach: Coverity

- Mature product
- Open-source friendly
  - Kind of
- Hooks with Travis-ci

- Pain to extract defects
  - Investigating with RTEMS

Second approach: clang-analyzer

- Active open-source project
  - Lot of other analyzers – also trying
  - Static Value flow (SVF) analyzer
- Easy to work with
  - Defect report is like compiler

- Must compile with clang
  - Used it with RTEMS
  - Hopeful to apply to EPICS soon!

Security Fuzzing

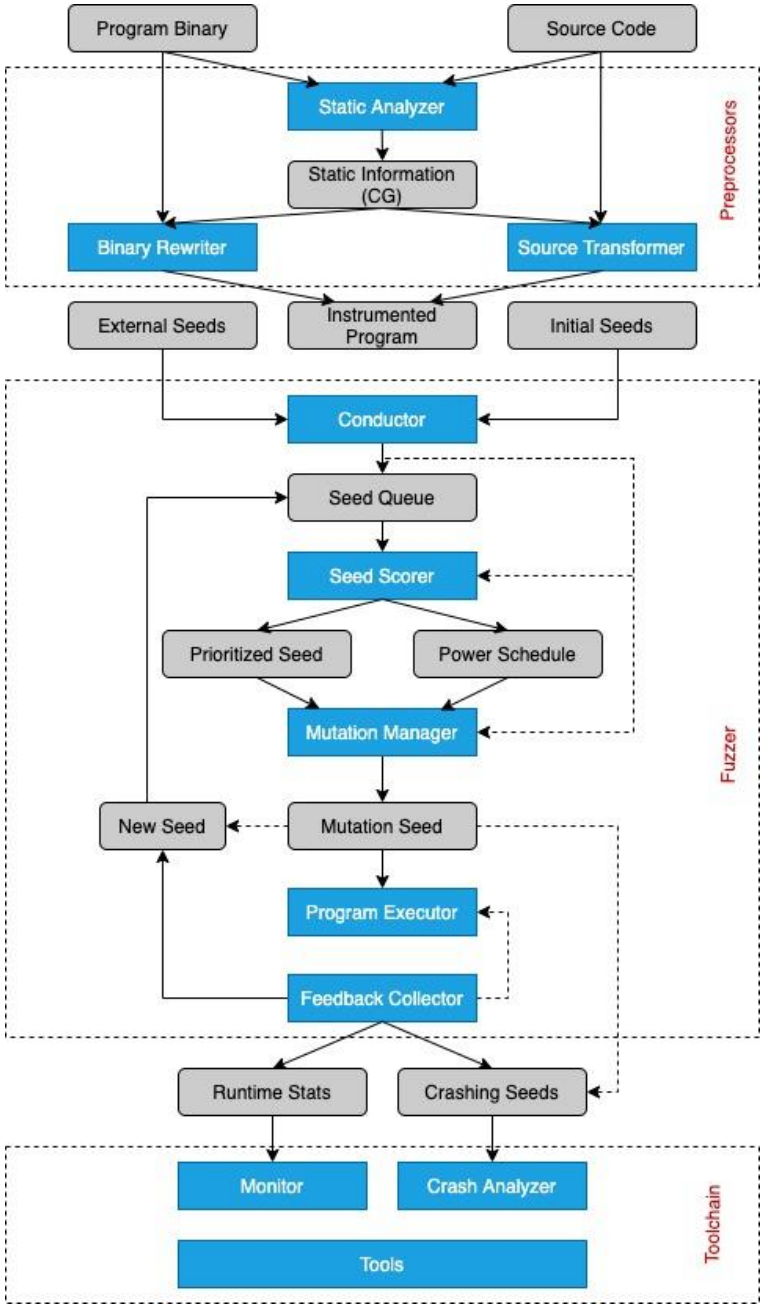
- ✔ Detect security loopholes and vulnerabilities on EPICS System
- ✔ Can detect zero-day vulnerabilities
- x Requires significant time to detect Bugs and vulnerabilities

Current state of the art is American Fuzzy Lop (AFL) fuzzer using mutation-based fuzzing

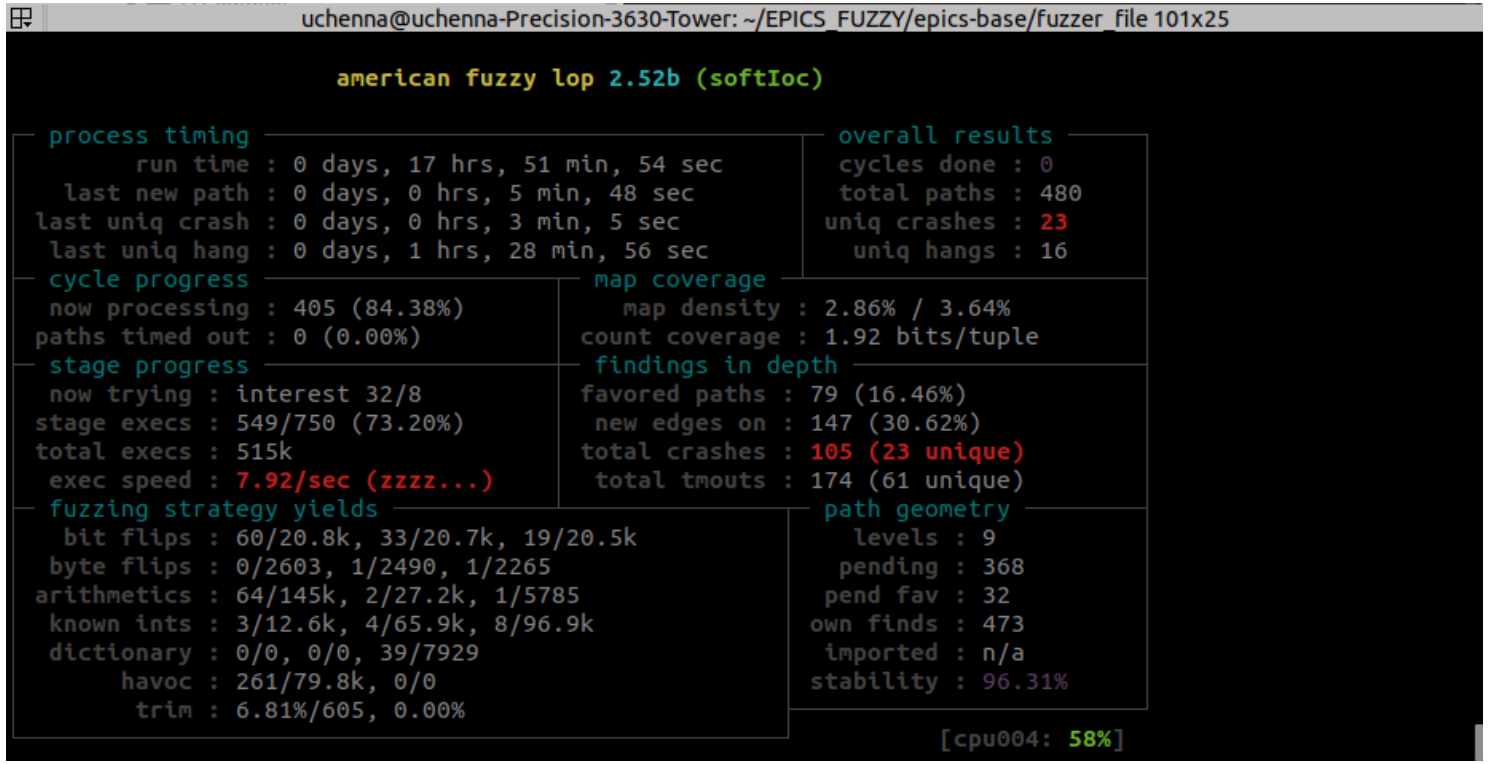
- As opposed to generation-based fuzzing

Our approach aims to improve AFL and apply to EPICS modules and CA

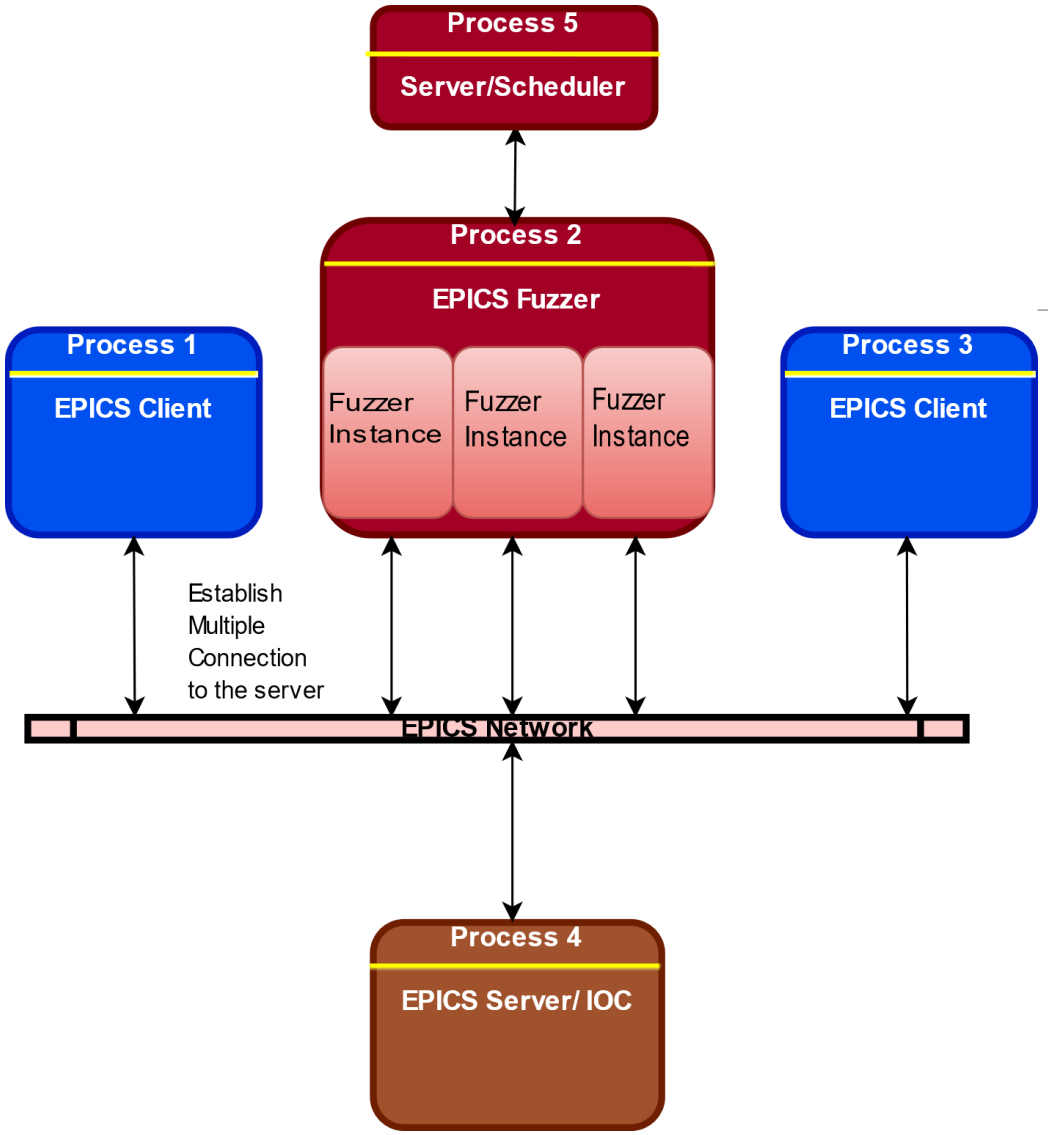
- Extension of AFL, AFL-NET, MOPT, UniFuzz and MultiFuzz to work for CA



Improve Fuzzing with Static Analysis



AFL running on softIoc



Current Research: Fuzzing Channel Access

The Fuzzer will have 1 scheduler/ server with N number of fuzzer (clients)

Each N clients can have M instances of Fuzzer (modified version of AFL)

Overload the network to investigate network vulnerabilities

Compare with existing network mutation fuzzer

Challenge: How to store multiple connections and synchronize paths covered and vulnerabilities found between fuzzers?

Formal Methods

THEOREM PROVING

Proof assistance to confirm/deny theorems formulated about system

- Manually labor intensive to formulate theorems so prover can reason over them

Size still a problem. Current approaches handle ~10k's lines of code

Strong expressivity in higher order logic

A few stable tools available

- Coq, HOL4, Isabella, PVS

MODEL CHECKING

Automatically check (verify) properties over an abstraction of system

- Brute force/heuristic approach without manual intervention

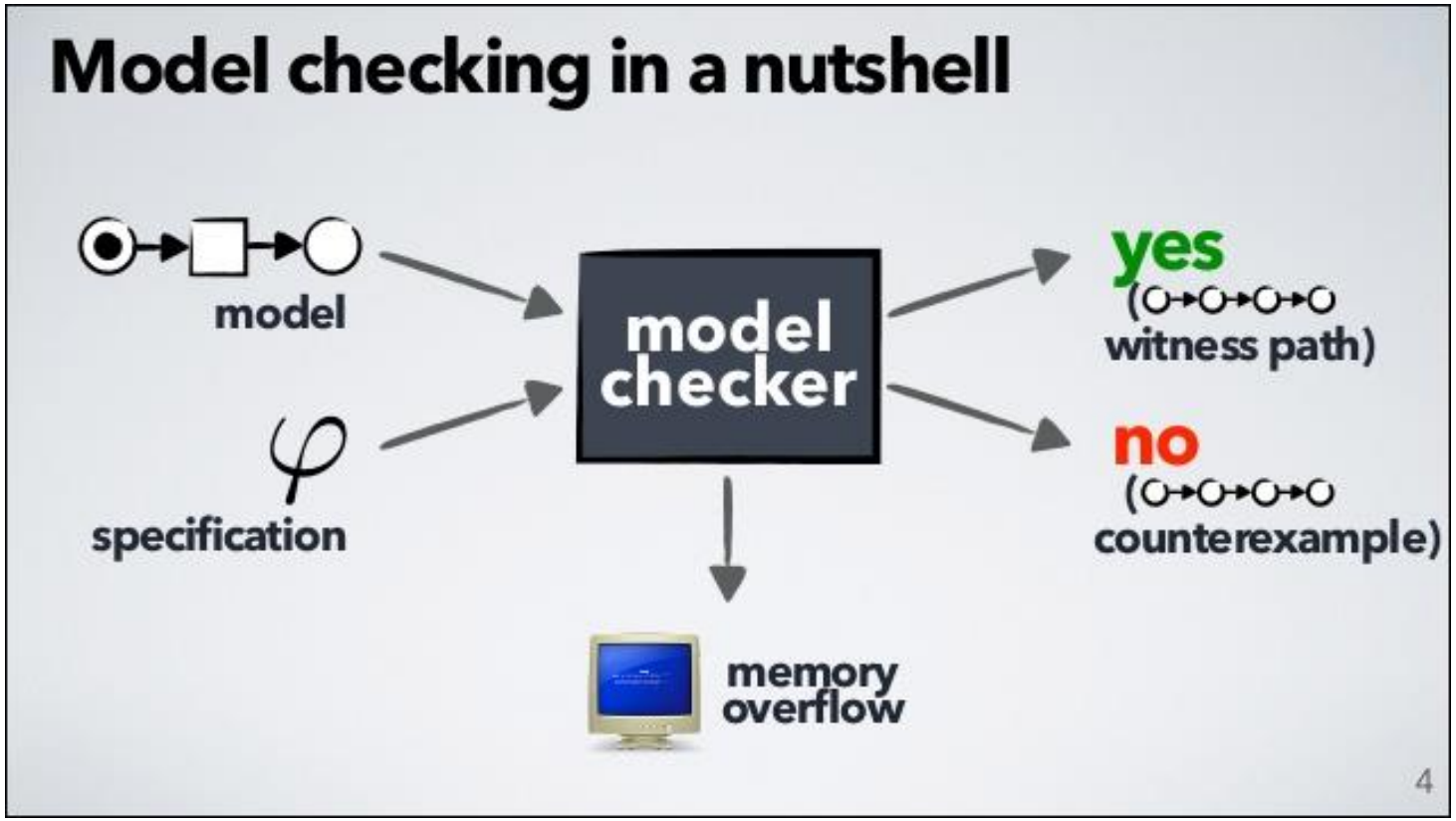
Constrained by size – state space explosion

Limited expressivity based on first order logic

- Usually enough for **safety** properties

Widespread tools available

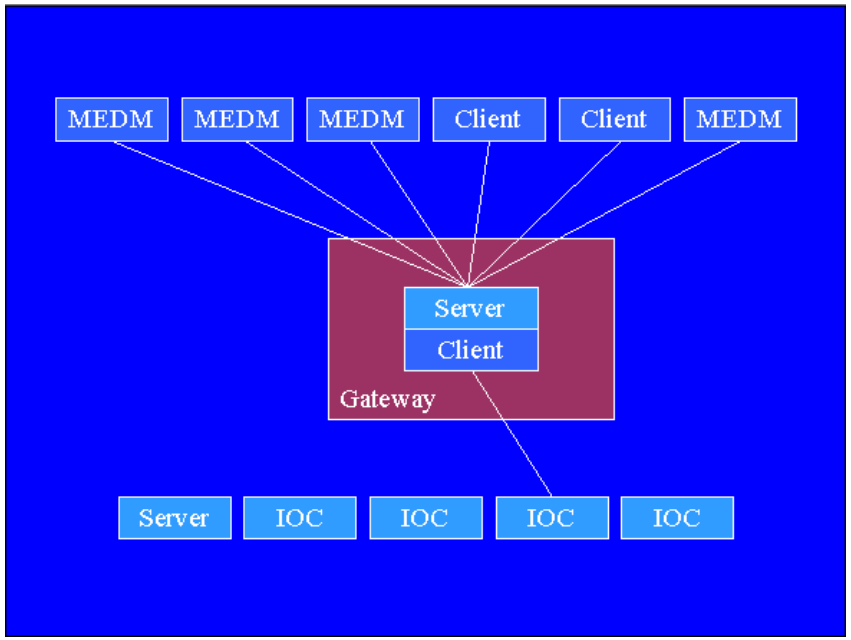
- SPIN, TLA+, UPAAL, PRISM, NuSMV, …



Our Approach

Use model checking to find bugs and concurrency problems

- Analyze PV Gateway using SPIN model checker



Why PV Gateway?

Widely used in EPICS community

Interesting (to us) Threat Model:

- Possibility of Dolev-Yao adversary
  - Network-level control (eavesdrop, add/drop messages)
- Written in C and all its glory
- Uses UDP → potential vulnerability to stateless attacks and UDP amplification attacks

Next Steps in Formal Methods

Finish initial modeling of PV Gateway

- Formulate some interesting properties to check if:
  - MEDM(client) can be restricted in their access to the PV
  - Gateway internal PV used for diagnostic purposes cannot be used as a backdoor?
  - PV_obj is destroyed after 120s unless it is connected?
  - PV_obj in inactive state for more than 7200s.
  - Transition between states are correct. For example, PV_obj in dead state should not be active in less than 120s.

Begin to model state space of CA/PVA protocols

Conclusion

Much work remaining to be done in these areas

Some next steps

- Security in open-source SDLC with signed commits, releases
  - Need to discuss with EPICS core and RTEMS maintainers
- Porting secure communications (dropbear ssh)
- Memory protection – Recurrent topic in RTEMS
  - Looking at how to create a model that fits across IOCs
- Secure Boot
  - ditto