Task 14 Cyberattack Tolerance SubBFT ++

Mar 21, 2023

Dr. Sukarno Mertoguno
Dr. Bo Feng
Muhammad Faraz Karim

Georgia Institute of Technology

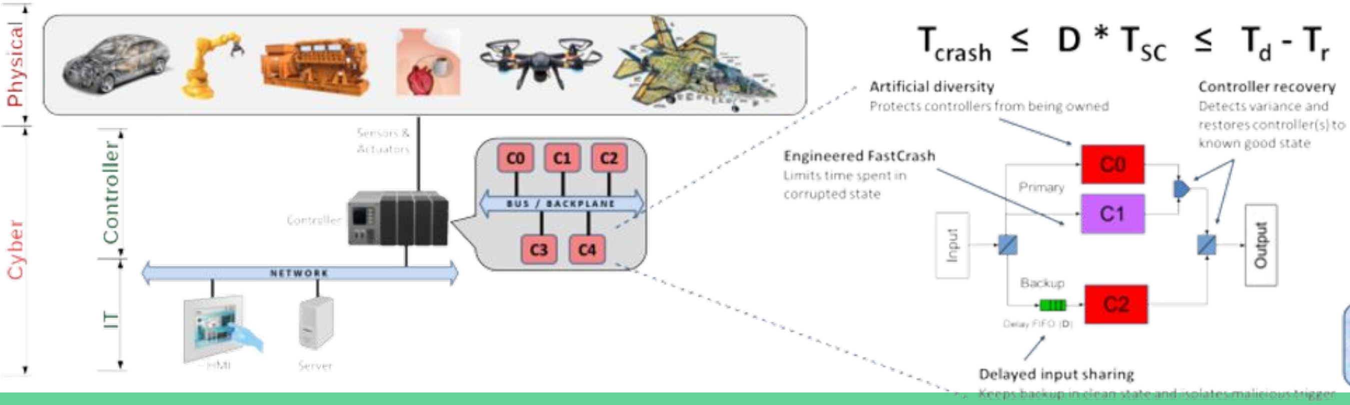
Cyber-Attack Resilience for CPS

Goal:

- Continuous, uninterrupted operation under direct cyber-attack campaign
- Develop an economical and robust cyber attack resilience at controller level (level 0 & 1), relying on the physical properties of the controlled physical systems.

Past Methods:

- BFT++ is a new approach to resiliency, leveraging established Fault Tolerant systems.
- Proposed SubProcess BFT++ (SubBFT++) will reduce the deployment cost for BFT++ cyber-attack-resilience.



$$T_{crash} \le D * T_{SC} \le T_d - T_r$$

Successful attack requires:

1. Success on derailing targeted program → targeted program loses control
2. Success on capturing control → attacker controls program execution

Existing BFT++ variants

Vanilla variant

- Multiple replicated devices with artificial software diversity to detect attacks
- A device replica with delayed input to promptly recover from attacks
- **More robust security guarantee**
- **Less service disruptions**
- **Higher overhead (due to device replica)**

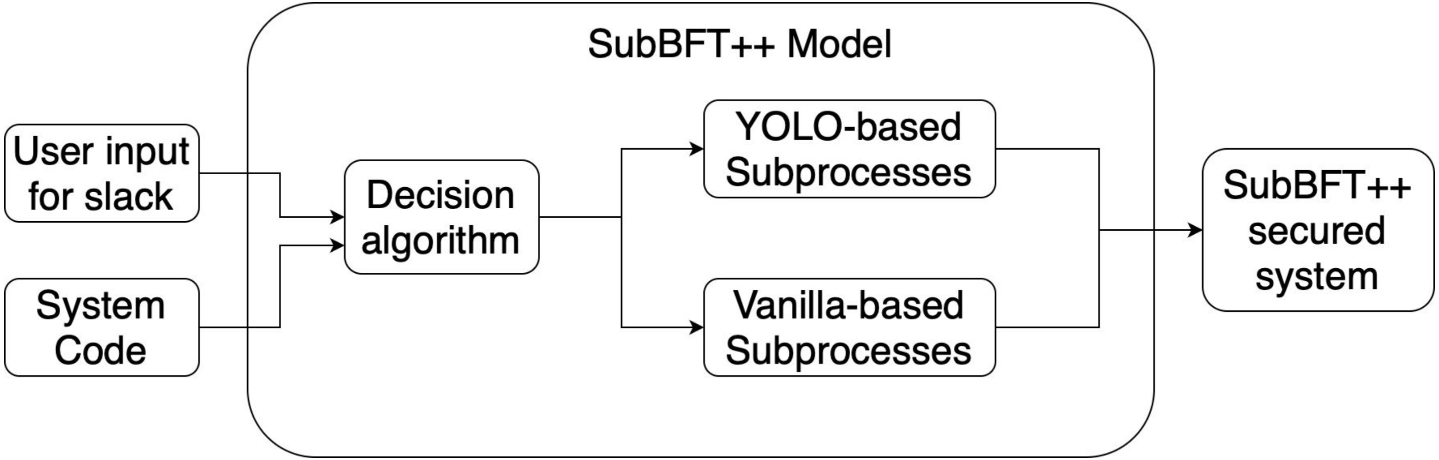
YOLO variant

- Firmware diversification (to probabilistically prevent and detect attacks)
- Frequent reset (to recover from attacks)
- **Lower overhead**
- **Probabilistic security guarantee**
- **More disruptions (due to frequent reset)**

Our solution: SubBFT++

- Goal:
    - **Robust defense**: as robust as the Vanilla variant
    - **Low cost**: comparable to the YOLO variant
- SubBFT++ stands for **Subprocess BFT++**
    - Operate on the subprocess level (the previous variants of BFT++ operate on the whole program)
    - For each subprocess, determine to duplicate (similar to vanilla variant) or randomize (similar to YOLO variant)
        - Utilize static program analysis techniques
        - Selection criteria: available slack in the system (idle time), minimum tolerable slack (instructed by the user), stateful or stateless, importance of the function

SubBFT++ Workflow



Control flow and decision making

- The Decision algorithm traverse the system code function by function and labels them according to the better fitting security technique based on function features, whether or not the function keeps state, is affected by resetting, how important/vulnerable the function is, etc.
- As for the implementation, there are two viable routes
  a. Passing a labeled control flow graph alongside the code to the diversification compiler and the compiler checks the label for each function before it proceeds to implement either technique
  b. Diversification and duplication happen for each function, but at run time, the scheduler links relevant copies of the functions based on their labels.

Diversification Tools

1. Multicompiler (Secure Systems Lab, UCI)
   LLVM-based compiler to create artificial software diversity to protect software from code-reuse attacks. This tool works on source code.

2. Embrittle (Galois, Inc.)
   A binary diversification and Fault Encouragement tool to harden a system against attacks. It works on binary input to create artificial binary diversity with minimal overhead. It significantly increases the bar of attacks.

Evaluation of diversification tools

- Multicompiler is our first choice as it works on source code
- However, it works only for x86 targets. Very few PLCs are based on x86
- We are porting it to ARM, a popular architecture for PLCs

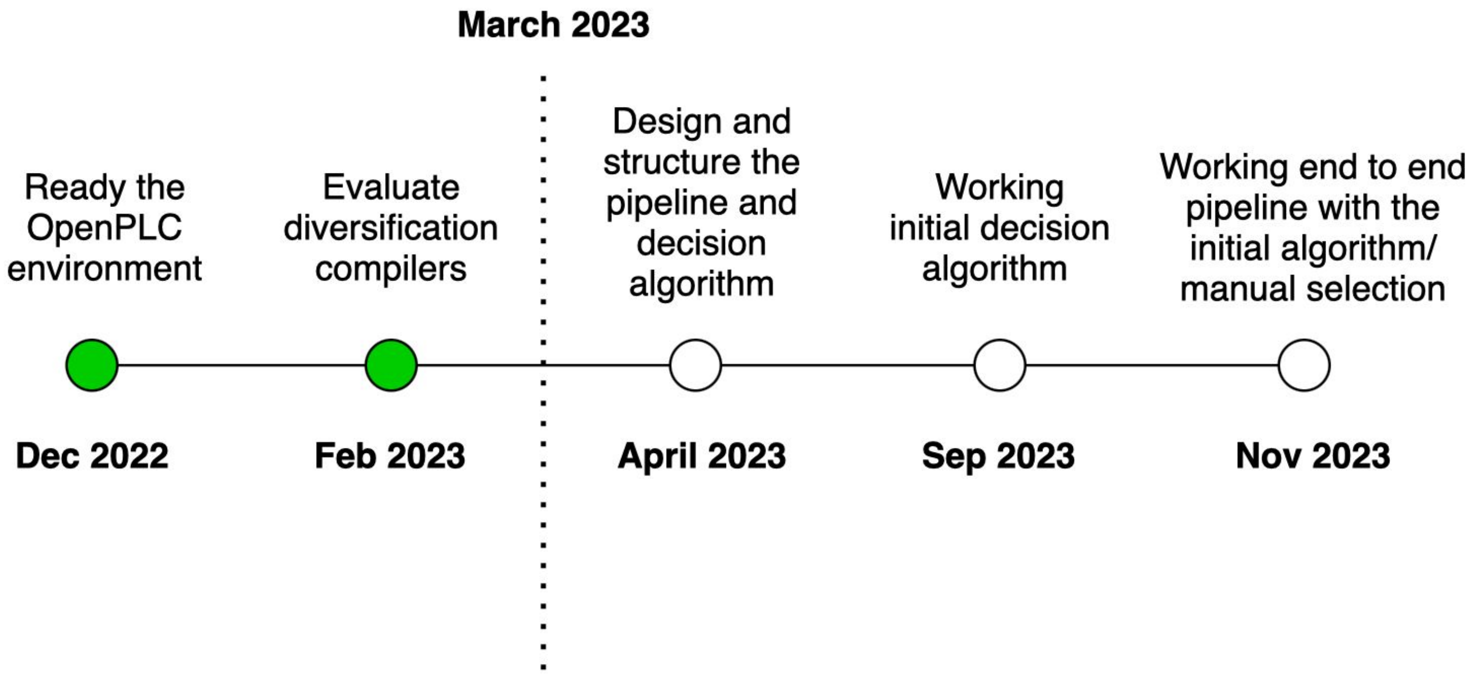
Integration with OpenPLC

- OpenPLC is an open-source Programmable Logic Controller used for industrial and home automation and internet of things. It allows for PLC programs to run on a spectrum of hardware, from raspberry pis to could servers

- Very practical for automating legacy systems since it can run on a variety of hardware, and does not require great processing power.

- Our decision algorithm runs when OpenPLC compiles the firmware and decides which part of the system is protected by which methodology

Timeline and milestones



March 2023

Ready the OpenPLC environment

Dec 2022

Evaluate diversification compilers

Feb 2023

Design and structure the pipeline and decision algorithm

April 2023

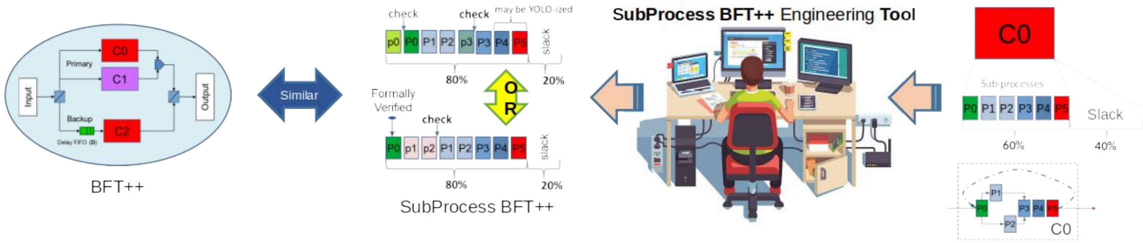
Working initial decision algorithm

Sep 2023

Working end to end pipeline with the initial algorithm/ manual selection

Nov 2023

Cyber-Attack Resilience for CPS – Part B



We plan to integrate the SubBTF++ engineering tool into OpenPLC design tools and environment.

Impact:

- Providing cyber attack resilience for application which cannot afford device redundancy, alleviate the need for redundant device in BFT++
- Significantly widen the applicability of BFT++ and resilience against direct cyber-attack
- Automated isolation of offending data, which can be communicated to other system components, e.g. SCATOPSY, RAM2., to prevent repeat attack. BFT++ team needs a a protocol & format for submitting malicious sample to both SCATOPSY & RAM2
- Integration into OpenPLC design environment for ease of deployment and dissemination.
- Discussion with Fathom5 (https://www.fathom5.co/) for potential integration of SubBFT++ into their toolset. Fathom5 runs Hack-The-Machine for the Navy, and is currently working w/ partners to commercialize the original BFT++

Thank you