Restart and Secure: An Approach to Enhance Security in Real-Time Cyber-Physical Systems

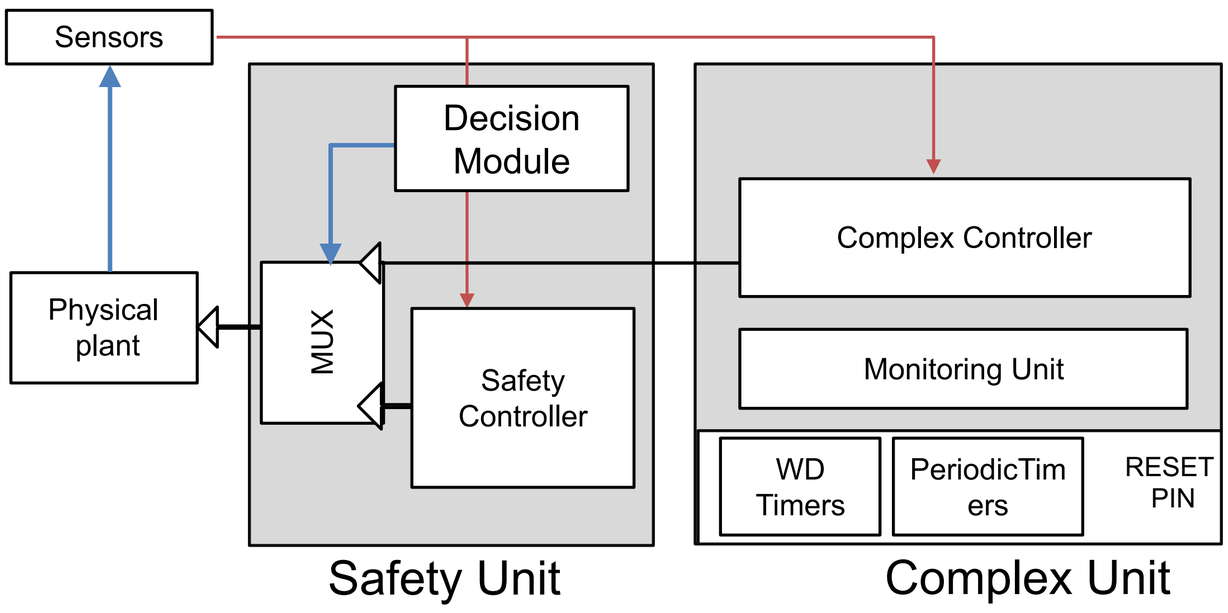
Disha Agarwala[1], Monowar Hasan[2], Fardin Abdi Taghi Abad[2] and Sibin Mohan[2,3]
[1]Dept. of Computer Science and Engineering, PES University, India
[2]Dept. of Computer Science, [3]Information Trust Institute, University of Illinois at Urbana-Champaign, USA

ENGINEERING AT ILLINOIS
INFORMATIONTRUST INSTITUTE

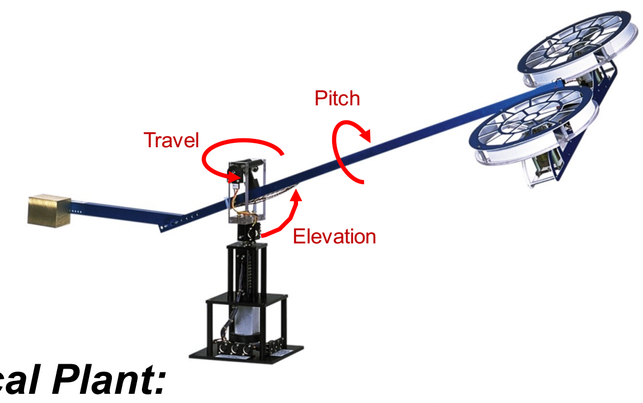
Introduction

- Evidence of recent attacks on safety-critical real-time cyber physical systems (RT-CPS) necessitates a unified protocol to ensure both *safety* and *security*
- We propose a restart-based architecture to ensure security in RT-CPS
  - Designed on top of *System-level Simplex* [1], a variant of *Simplex* [2]
- The postposed framework allows the designers to integrate own security mechanisms without sacrificing safety and real-time constraints

Architecture



Safety unit guarantees safety and temporal constraints when complex unit is not available (e.g., during restart)



Physical Plant:

- Quanser 3-DOF helicopter
- HIL Simulations

Safety Unit:

- x86 PC (Intel Core i7 processor, 8GB RAM)
- Ubuntu Linux 14.04

Complex Unit:

- BeagleBone Black development board (ARM Cortex-A8 1GHz processor, 512MB RAM)
- Debian Linux with RT-PREEMPT patch (kernel version 4.4.12-ti-rt-r30)

Triggering Restart

- *Monitoring Unit*
  - ✓ Checks the current kernel modules with the original modules from the `proc/modules` file
  - ✓ Triggers a reboot if malicious entity is found
- *Watchdog Timer*
  - ✓ Controller tasks update a watchdog timer at every execution loop
  - ✓ Failing to update watchdog will trigger a restart
- *Periodic Timer*
  - ✓ Restates the system periodically

Approach and Workflow

- The safety unit is a multi-threaded program
  - Communicates to the complex unit using *sockets*
- The safety controller, the complex controller and the decision module are *periodic* asynchronous threads
- The commands of safety and complex controller are fed to the decision module
  - If the complex controller doesn't send a command within 10ms control switched to safety controller
- The complex controller updates the watchdog timer at each execution loop
- The periodic timer restarts the system after every 10 seconds

Experience and Evaluation

❑ *Recovery by Monitoring Unit and Periodic Timer*

- We inject malware that silently lodges in the system and extracts information
- The system is recovered from the attack within:
  - 2 seconds If the restart is triggered by monitoring unit
  - 23 seconds when system reboots due to periodic timeout

❑ *Recovery by Watchdog Timeout*

- Launch DoS attack (fork bomb) by injecting malicious code within highest priority controller tasks
- A restart event is triggered since the watchdog is not updated
- The system recovers in 14 seconds

Conclusion

- The framework:
  - Prevents attackers from causing harm to the physical systems, even if the intruder gain control over the controller tasks
- Performance-security tradeoff
  - Use of dedicated computing module and unavailability of complex controller during restart
  - Might be *acceptable* for security-critical CPS

References

[1] Bak, S., Chivukula, D.K., Adekunle, O., Sun, M., Caccamo, M. and Sha, L., 2009. The system-level Simplex architecture for improved real-time embedded system safety. In *IEEE RTAS,* pp. 99-107
[2] Sha, L., 2001. Using simplicity to control complexity. *IEEE Software*, *18*(4), pp. 20-28