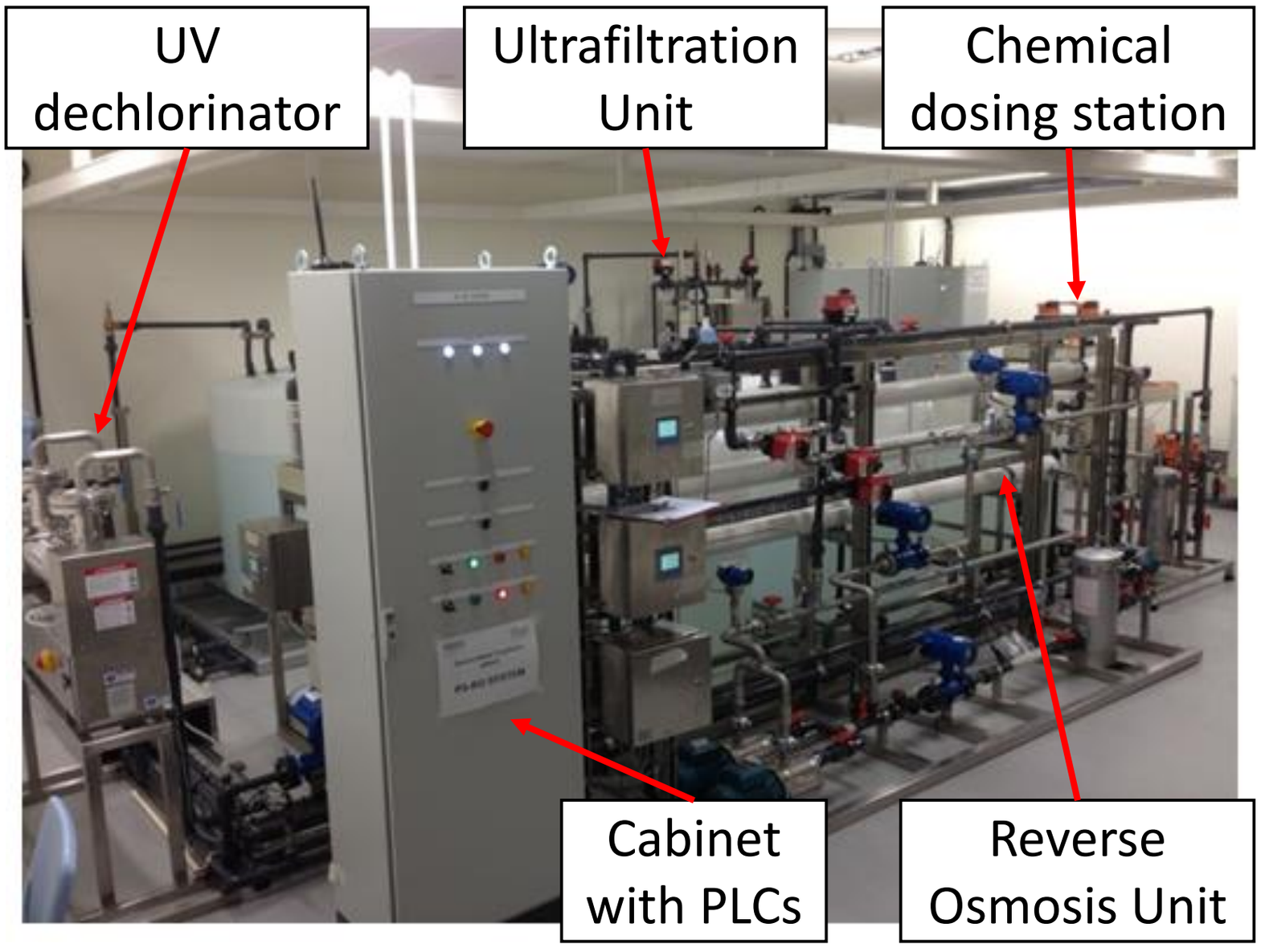
Systematic Attack Generation for Industrial Control Systems

Eunsuk Kang

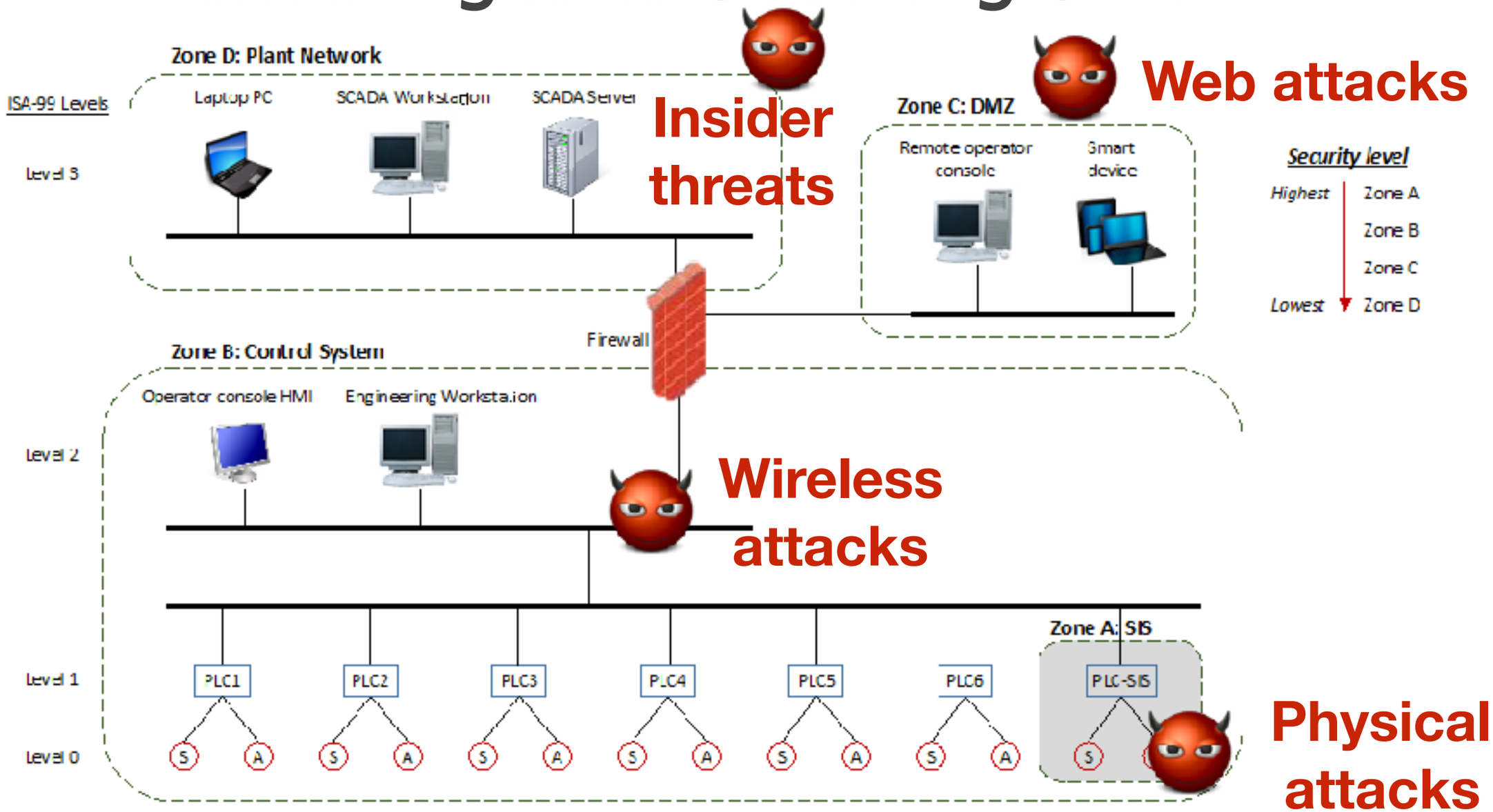
Sept 24, 2020

Secure Water Treatment Plant (SWaT)



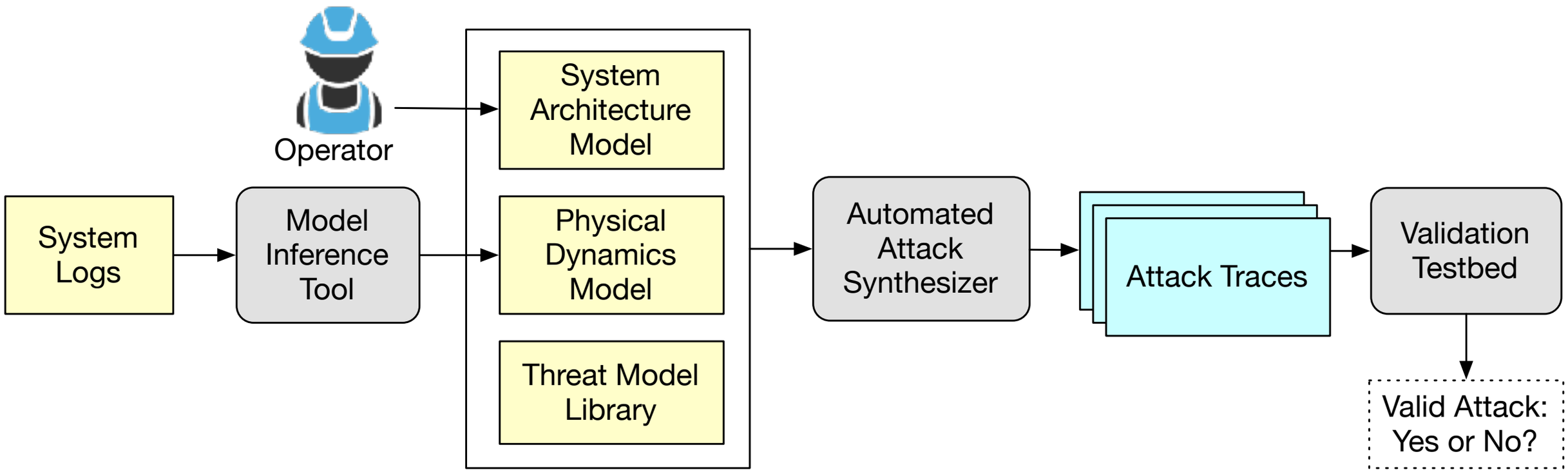
- Fully functional water treatment plant, developed at Singapore University of Technology & Design (SUTD)
- 6-stage distributed control system; 62 sensors & actuators
- Wireless communication to programmable logic controllers (PLCs)

Challenges to Securing SWaT



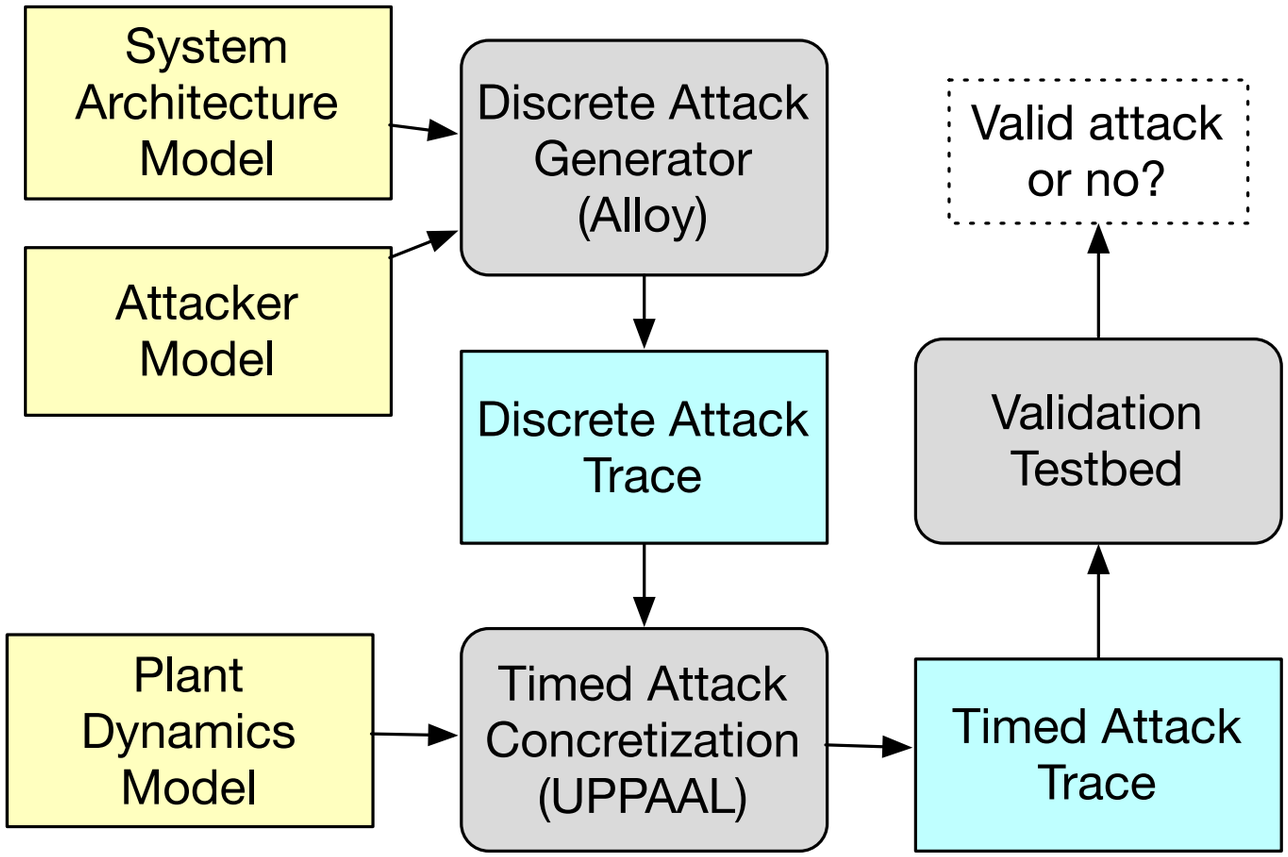
- **Legacy SCADA:** Little built-in security protection; limited use of crypto; connected to the Web (remote operator interface)
- **Heterogenous**: Network + software (PLC) + physical processes
- **Beyond security**: An attack can have **safety** implications (e.g., tank overflow, pump damage, water contamination)

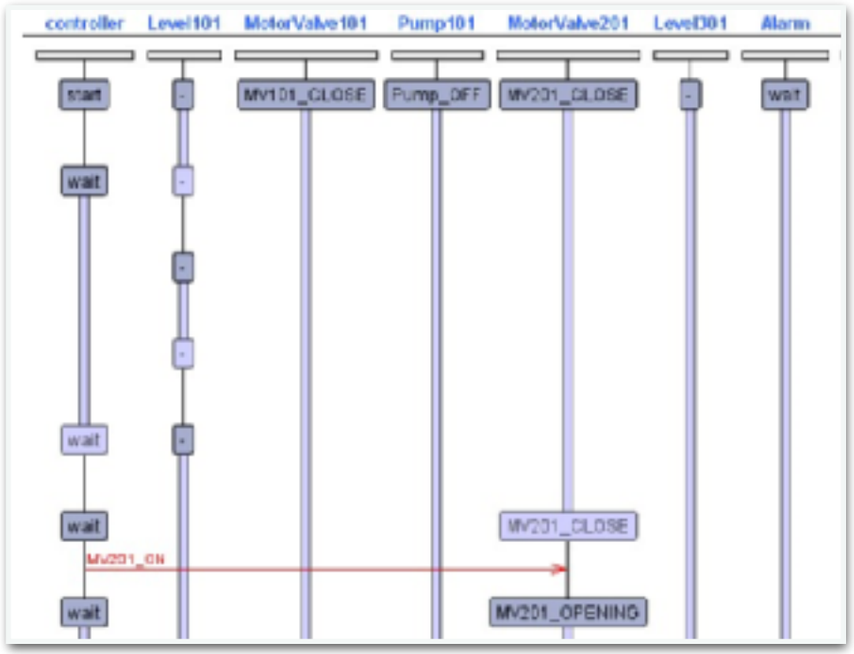
Automating Security Evaluation of ICS



- **Goal:** What are possible attacks on the system that could lead to safety failures? Can we synthesize & validate them automatically?
- **Benefits:** (1) Reduce the cost of security testing and (2) Identify potential security flaws before deployment
- **Research Thrusts**
  - Model-driven, automated attack synthesis using formal methods
  - Data-driven inference of physical dynamics model

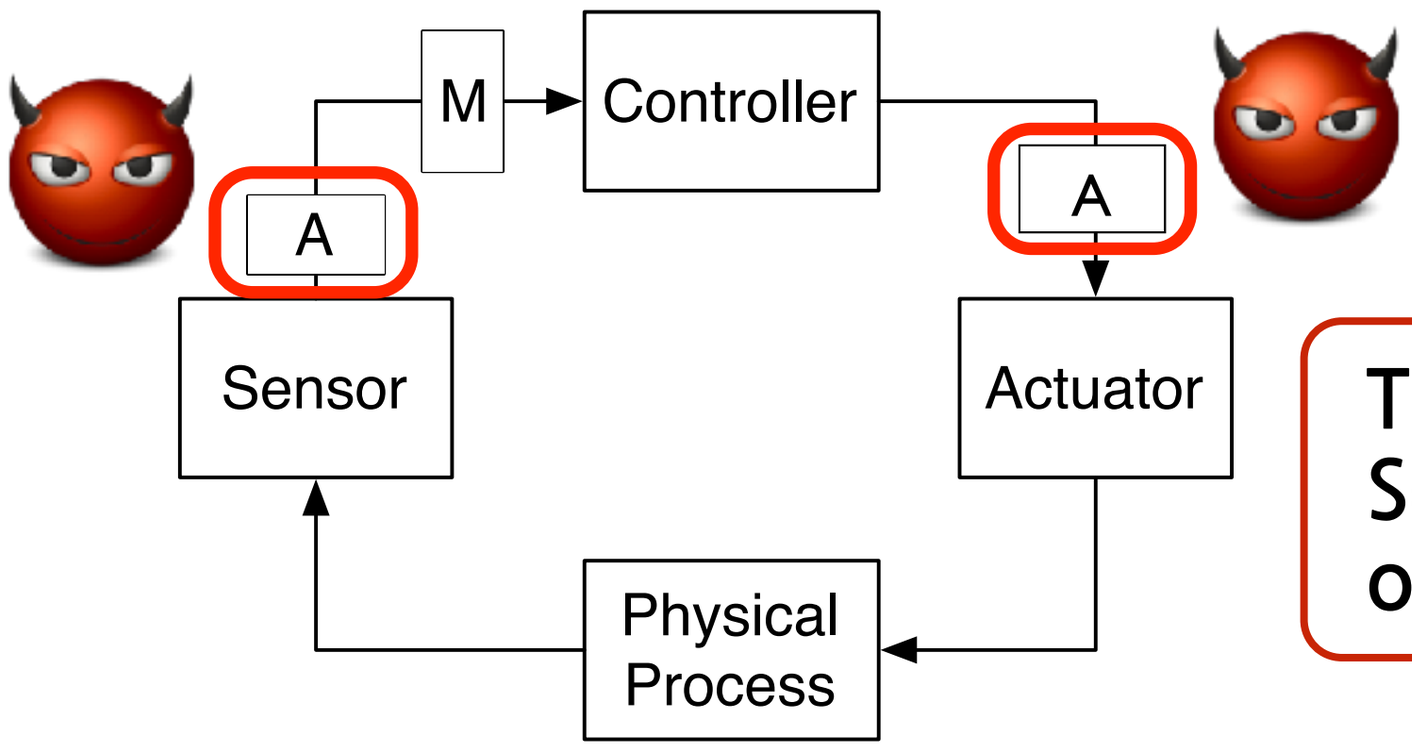
Thrust 1: Attack Synthesis for ICS





- **Goal**: Automatically synthesize **targeted** attacks offline for security testing
- **System model**: Connections between controllers, sensors and actuators; controller logic; built-in safety monitor
  - e.g., Monitor: “Raise an alert if the water tank is about to overflow”
- **Attacker model**: Manipulate sensor readings & actuator commands
- **Stealthy attack generation**: Generate sequences of attack actions that bypass the monitor & induce system into an unsafe state (e.g., overflow)

Modeling an Attacker



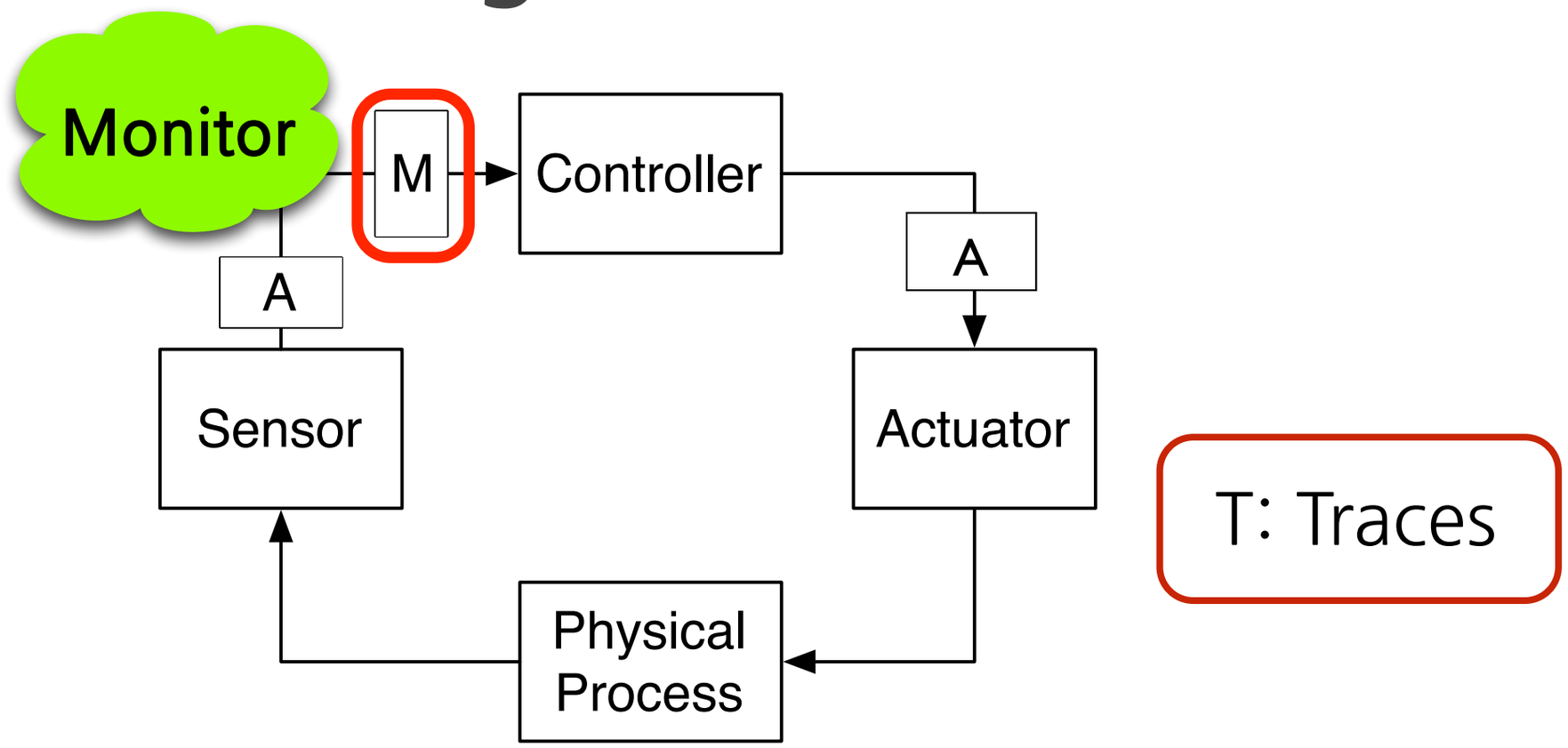
T: Traces = Sequences of observations

Attacker as an **edit function**

$$A: T \rightarrow T$$

Edit automata: Enforcement mechanisms for run-time security policies
Ligatti, Bauer, Walker (2005)

Modeling the Monitor



Attacker as an **edit function**

$$A: T \rightarrow T$$

Monitor as a **predicate on traces**

$$M: T \rightarrow \{true, false\}$$

where **M(t) = true** if system execution **t** satisfies its invariants

Targeted, Stealthy Attack Synthesis

Given a particular monitor (**M**), is there
an unsafe trace that remains undetected by M?

$$\exists\, t,t' \in T \mid \neg safe(t) \wedge t' = A(t) \wedge M(t') = true$$

original trace

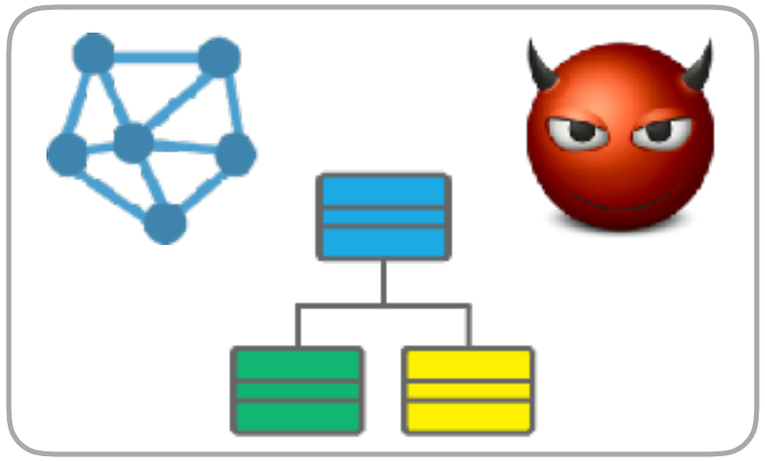
edited trace

Example

safe(t) = “Water level must remain below a max. threshold over trace t”
M(t’) = “Water level rises if and only if there is inflow into the tank”

Attack Synthesis as Constraint Solving

Models (system architecture, attacker) + safety requirement

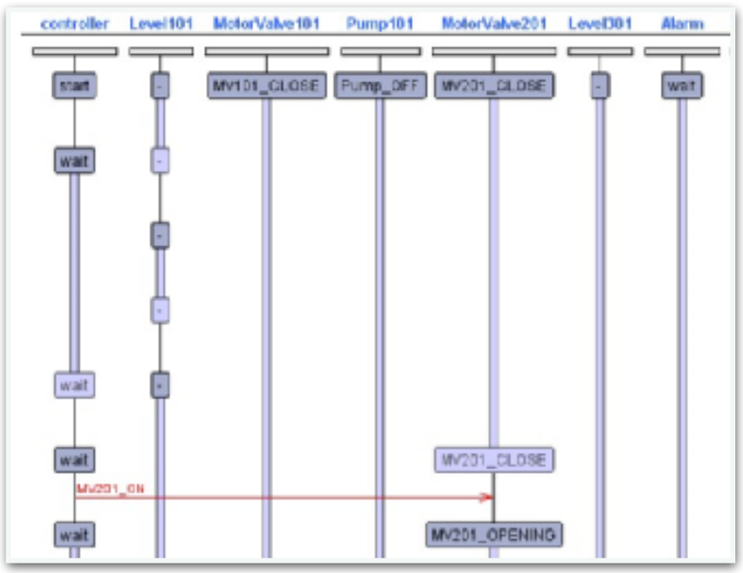


Logical constraints

$$...(b \vee (x + y \leq 0))$$
$$(\neg b \vee (x + z \leq 10))$$
$$\forall x \cdot (x - y \leq 0) \wedge$$
$$(z - x \leq -1)...$$

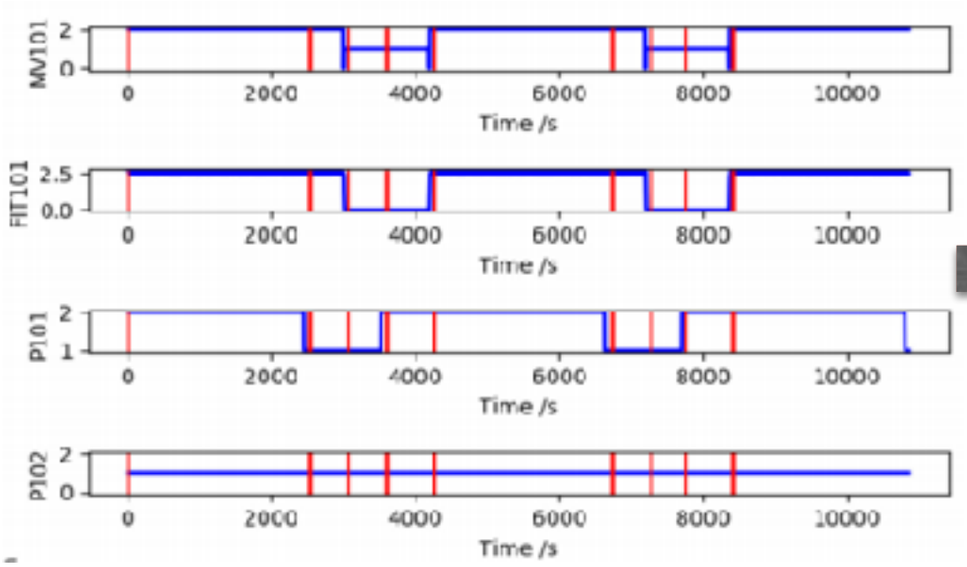
Constraint Solver (SMT)

"Is there a possible attack that results in a safety violation?"

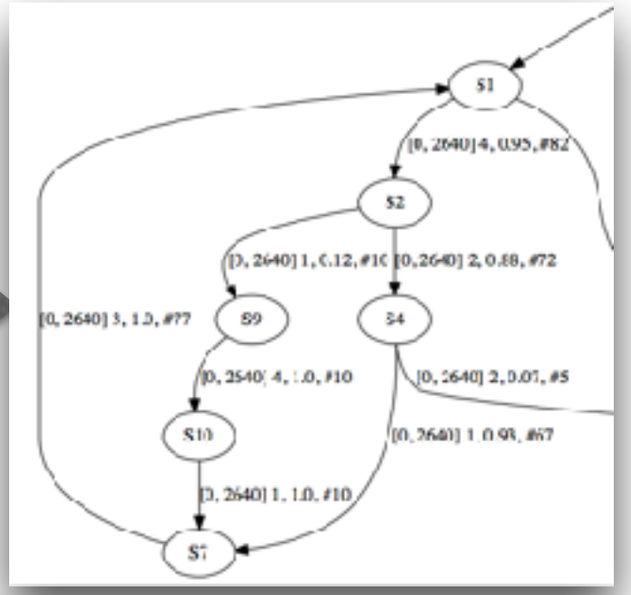


Satisfying instance as an attack trace

Thrust 2: Learning ICS Dynamics Model



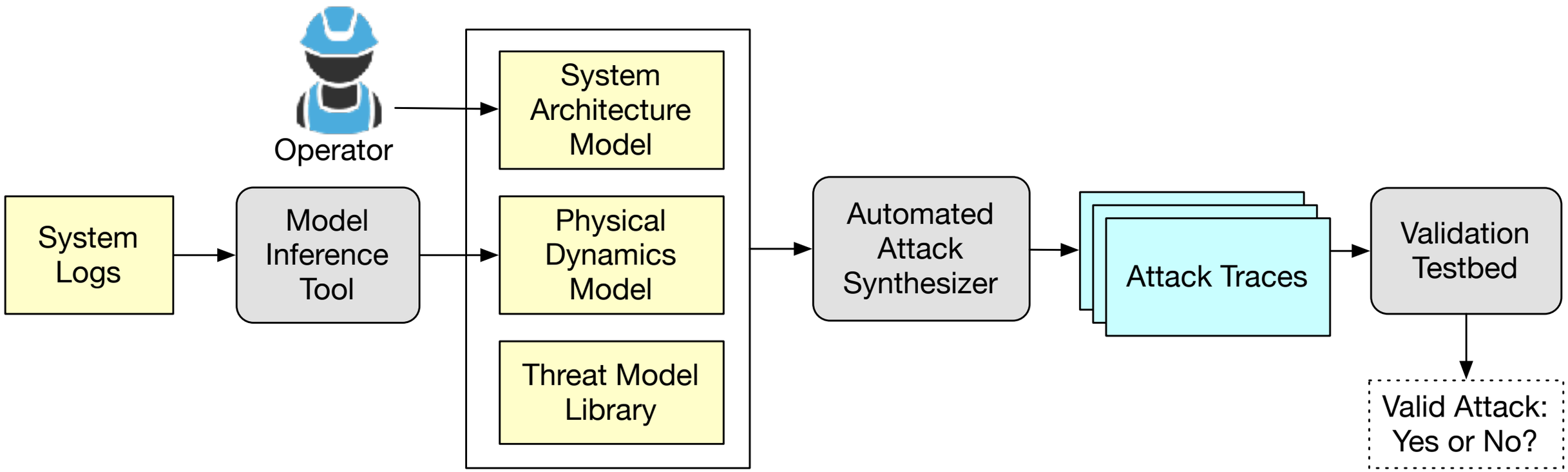
TABOR: Model Inference Tool



Sensor & actuator signals **Timed automata**

- **Goal**: Learn models capturing dynamics of sensors & actuators from plant operational logs
- Learn **interpretable** models that can be composed with system models for formal reasoning
- Underlying approach: Program synthesis with deep reinforcement learning & MCTS (joint work w/ Vincent Hellendoorn)

Automating Security Evaluation of ICS



- **Goal:** What are possible attacks on the system that could lead to safety failures? Can we synthesize & validate them automatically?
- **Benefits:** (1) Reduce the cost of security testing and (2) Identify potential security flaws before deployment
- **Research Thrusts**
  - Model-driven, automated attack synthesis using formal methods
  - Data-driven inference of physical dynamics model

Critical Infrastructure: Interconnection

Water Treatment

Water Distribution

Power generation, transmission, distribution

- Modeling & analysis of cascading attacks across multiple ICS
- Design methods to achieve resiliency against cascading attacks

Thank you!
Any questions?

eskang@cmu.edu