Protocol Fingerprinting and Casuality-Aware Intrusion Detection in Industrial Control Systems

Team Members

Epifanio Sarinana
Efrain Gonzalez
Luis Gaytan
Diego Garcia

Goals

- Develop a two-layer IDS for ICS:
    - Layer 1: Protocol Fingerprinting Module for lightweight detection of deviations in traffic patterns (Modbus, DNP3, Profinet)
    - Layer 2: Casuality Tracking Module graph reconstruction across domains (network -> PLC -> physical process), optimized with reduced precision or approximate methods for faster computation
- Demonstrate how combining protocol-level and causality-level defenses improves anomaly detection while remaining efficient

Objectives

1. Identify key ICS protocol fingerprinting features (packet length, function codes, timing, payload entropy)
2. Build a lightweight tool for capturing traffic and extracting protocol fingerprints
3. Reconstruct causality graphs linking network traffic, control logic, and process state
    a. Implement reduced-precision/approximate graph construction methods
4. Evaluate system performance on ICS datasets/testbed, comparing:
    a. Fingerprinting only
    b. Casualty Only
    c. Combined two-layer approach

Timelines

Week 1-2: Project setup & Literature Review

- Study ICS protocols (Modbus, DNP3, Profinet) and ICSTracker
- Select datasets/testbed (SWaT, BATADAL, etc)
- Define protocol features (packet size, function code, timing, entropy)

Week 3: Fingerprinting Design

- Build packet capture + parsing pipeline (Scapy, PyShark)
- Implement feature extraction for ICS protocol (eg. Modbus)
- Define protocol features (packet size, function code, timing, entropy)

Week 4: Fingerprinting Implementation

- Extend the parser to the rest of the ICS protocol
- Develop a lightweight ML detector
- Test on sample traffic

Week 5-6: Baseline Causality Tracking

- Implement a simple graph builder linking packet -> PLC command -> sensor/actuator response
- Use existing logs to validate causal chains
- Establish baseline performance (accuracy, overhead)

Week 7: Reduced-Precision Optimization

- Integrate reduced precision or approximate graph metrics
- Measure computation speed-up vs. accuracy loss

Week 8: Advanced Causality

- Add dependency recovery
- Compare against the ICSTracker baseline

Week 9: Layer integration

- Combine fingerprinting + causality

Week 10: System Testing

- Evaluate the combined IDS on the dataset/testbed

Week 11: Refinement

- Tune

Week 12

- Finalize project report and presentation

Roles & Task Assignment

Epifanio Sarinana

- Implement the causality graph
- Integrate reduced precision/approximate computing
- Benchmark

Efrain Gonzalez

- Build fingerprinting
- Optimize lightweight detection
- Test and validate

Luis Gaytan

- Define fingerprint features

- Implement packet capture & parsing
- Document fingerprinting methodology

Diego Garcia

- Integrate fingerprinting + causality
- Run system testing on the dataset/testbed
- Collect metrics

Milestones

Week 1-2: Literature review + feature extraction (everyone)
Week 3: Fingerprint design
Week 4: Fingerprinting implementation
Week 5-6: Baseline causality graph constructions
Week 7: Reduced precision to graph
Week 8: Advanced graph
Week 9: Integrate fingerprinting + graph
Week 10: System testing
Week 11: Refinement
Week 12: Final report