CPS-SPC 15 @ Denver CO

MiniCPS: A toolkit for security research on CPS Networks

Daniele Antonioli (SUTD)  Nils Ole Tippenhauer (SUTD)
Hi!

• Personal:
  ▶ DANIELE ANTONIOLI
  ▶ SUTD’s ISTD PhD (Prof N.O. TIPPMENHAUER)
• SCy-Phy group:
  ▶ Applied CPS security research
Why MiniCPS: Cyber-Physical Systems

• CPS are:
  ▶ Complex
  ▶ Critical
  ▶ Connected
Why MiniCPS: Cyber-Physical Systems

- CPS are:
  - Complex
  - Critical
  - Connected

- CPS information may be difficult to:
  - Obtain
  - Prove
  - Share
Why MiniCPS: Cyber-Physical Systems

- CPS are:
  - Complex
  - Critical
  - Connected

- CPS information may be difficult to:
  - Obtain
  - Prove
  - Share

- CPS research requires different expertises:
  - Electronics, Automation
  - Networking, Computer Science
  - Physics...
Why MiniCPS: SWaT testbed

- Pure Water: 5 US gallons/min, 6.0 – 7.0 pH, minimum conductivity of 10 $\mu$S/cm$^3$
- Recovered Water: 70% processed water, 50% dirty recirculation
Why MiniCPS: SWaT network

- Wired and Wireless links.
- Ethernet/IP, Common Industrial Protocol.
MiniCPS: Vision

• Research Environment:
  ▶ Reproducible
  ▶ Extensible
  ▶ Shareable
MiniCPS: Vision

• Research Environment:
  ▶ Reproducible
  ▶ Extensible
  ▶ Shareable

• Targeted to Cyber-Physical Systems:
  ▶ Network communications
  ▶ Control logic
  ▶ Physical layer interaction

Don’t reinvent the wheels. . .

But: “Stand on the Shoulders of Giants”

E.g: linux, python, mininet, git
MiniCPS: Vision

• Research Environment:
  ▶ Reproducible
  ▶ Extensible
  ▶ Shareable

• Targeted to Cyber-Physical Systems:
  ▶ Network communications
  ▶ Control logic
  ▶ Physical layer interaction

• Don’t reinvent the wheels... 
  ▶ But: "Stand on the Shoulders of Giants"
  ▶ Eg: linux, python, mininet, git
MiniCPS: Diagram

- (C)yber → Network Emulator
- (P)hysical → Process Simulation, State API
- (S)ytem → Control Logic Simulation
• Network-in-a-box emulator:
  ▶ Reproduce (complex) topologies
  ▶ Generating real packets using real protocols
MiniCPS: What is Mininet

- Network-in-a-box emulator:
  - Reproduce (complex) topologies
  - Generating real packets using real protocols
- One Linux kernel, multiple devices:
  - Lightweight virtualization
  - Each device is a container
MiniCPS: What is Mininet

- Network-in-a-box emulator:
  - Reproduce (complex) topologies
  - Generating real packets using real protocols
- One Linux kernel, multiple devices:
  - Lightweight virtualization
  - Each device is a container
- SDN/OpenFlow development
MiniCPS: Physical Layer API

- Database to represent the (physical) state:
  - Abstract low-level details (SQL query)
  - Use high level semantic functions: `get`, `set`
• Database to represent the (physical) state:
  ▶ Abstract low-level details (SQL query)
  ▶ Use high level semantic functions: get, set

• Compatibility layer:
  ▶ Programming Language agnostic
  ▶ Support different storage back-ends
MiniCPS: SWaT example

- **Control strategy:**
  - Sensors: level (LIT), flow (FIT)
  - Actuators: motorized valve (MV) and pump (P)
  - PLC1 takes decision with the aid of PLC2 and PLC3
  - Physical process simulation updates the state

- **Network:**
  - Realistic addresses (CIDR, MAC, ports)
  - Replicate services: web-servers, ENIP client/server
  - Optional Attacker and SDN Controller
• *Passive* and *Active* ARP poisoning MITM attacks
• *SDN Controller* for ARP poisoning Detection and Mitigation
• MiniCPS is a CPS research platform:
  ▶ Reproducible
  ▶ Extensible
  ▶ Shareable

• MiniCPS is used to investigate issues in real testbeds:
  ▶ MITM attacks (ettercap)
  ▶ Ethernet/IP reverse-engineering (scapy)
  ▶ SDN controllers development (pox)
MiniCPS: Conclusions

• MiniCPS is a CPS research platform:
  ▶ Reproducible
  ▶ Extensible
  ▶ Shareable

• MiniCPS is used to investigate issues in real testbeds:
  ▶ MITM attacks (ettercap)
  ▶ Ethernet/IP reverse-engineering (scapy)
  ▶ SDN controllers development (pox)

• Contribute:
  ▶ http://scy-phy.github.io/index.html
  ▶ https://github.com/scy-phy/minicps

• Thank You!

Q & A