Attack Detection, Isolation, and Reconfiguration of Control Systems Leveraging SDNs

DDDAS PI Meeting 2017

Alvaro Cardenas
Cy-Phy Security Lab
UT Dallas
September 7, 2017
Security and Control Experts Must Work Together

• Security is not only about keeping attackers out
• It is also about
  – Understanding risks
  – Mitigating
  – Detecting
  – Responding
to adversaries that have partial access to your system

• Computer scientists need to understand control
  – Cardenas, Amin, Sastry. Usenix Hotsec 2008
• Control engineers need to think adversarially
  – Cardenas, Amin, Sastry. ICDCS CPS workshop 2008
Defense in Depth

• Security is not only about keeping attackers out
• It is also about
  – Mitigating
  – Detecting (DDAS 2016)
  – Responding (DDDAS 2017)
• to adversaries that have partial access to your system
• We need to have multiple models/simulations of the physical system
• We obtain data from sensors
• Data from sensors and simulations need to be reconciled
• If anomalies are detected, then we need to dynamically change how data is obtained
  – e.g., replacing data with simulations or getting new data/new estimation results
DDDAS Anomaly Detection and Response (DDDAS-ADR)

- **Actuators** → $u_k$ → **Physical Process (Plant)** → $z_k$ → **Sensors** → $y_k$
- $y_{k-1} → $ → **Simulation** → $\hat{y}_k$ → **Anomaly Detection** (ignore bad sensors, reconfigure simulation)
- $u_k → \mathcal{T}_k → \text{Reconfigure Controller (account for bad actuators)} → \text{Reconcile Data} → \text{Dynamically Request More Data from Other Systems}
Detection = Simulation + Statistics
Response = SDN+NFV
Current Work: Prototype in Mininet

• Mininet: light virtualization environment tailored for software-defined networks (SDN)
• Industrial control protocol EtherNet/IP in Mininet
• Control algorithm: system of water tanks

• https://github.com/Cyphysecurity/ICS-SDN
Responding to Sensor Attacks

1. Notify IDS 101
2. Cloud Infrastructure
3. Notify SDN Controller: LIT101 Compromised
4. Lookup topology: LIT101 Port 4, Plant Estimator Port 5
5. Delete rule in_port: 4 in_port 4: drop in_port 5: forward
```python
POX Application Instructions | Switch Flowtable Result
---|---
if message['Variable'] == 'Switch_flow':
    self.compromised_sensor = True
    self.switch_flow()

self.switch_flow():
    msg = of.ofp_flow_mod(command=of.OFPFC_DELETE)

def _handle_PacketIn(self, event):
    if (in_port == 4) and (nw_src == "192.168.1.10") and (self.compromised_sensor):
        return

    port = self.macToPort[packet.dst]
    msg = of.ofp_flow_mod()
    msg.match = of.ofp_match.from_packet(packet, event.port)
    action = of.ofp_action_output(port=port)
```

| nw_src=192.168.1.10, in_port=4, actions=output:2 |
| nw_src=192.168.1.10, in_port=4 |
| nw_src=192.168.1.10, in_port=4 |
| nw_src=192.168.1.15, in_port=5, actions=output:2 |
Water Tanks Level Behavior With Attack and IDS

With SDN Defense
Without Defense
Normal Operation
Future Work: Applications to UAV and Autonomous Systems

• Leveraging our DDDAS-ADR to other autonomous cyber-physical systems
  • Coordination of UAVs/swarms to achieve mission objective under attacks
    • Malicious sensors
    • Malicious agents
    • Incorrect telemetry
Questions?

Alvaro A. Cardenas
alvaro.cardenas@utdallas.edu