RESILIENCY SURVEY: CHALLENGES GOING FORWARD

Mladen Vouk, David Nicol, Kevin Sullivan, David Garlan Zbigniew Kalbarczyk, Ravi Iyer

University of Illinois at Urbana-Champaign
North Carolina State University
Carnegie Mellon University

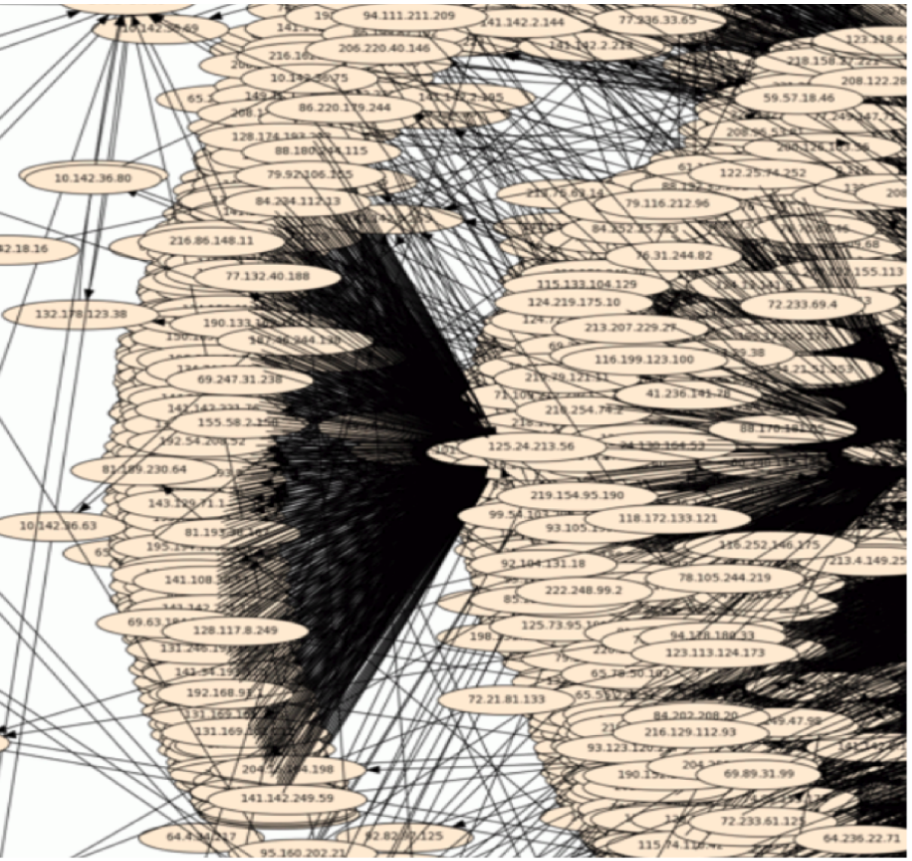
November 29, 2014

Resilience and Security

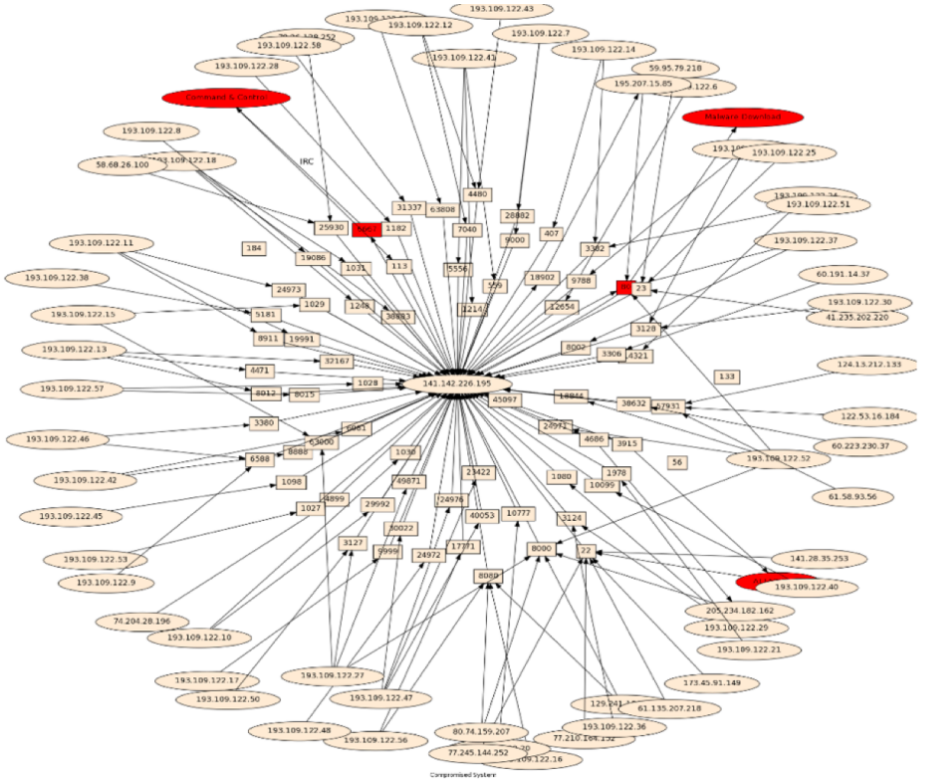
- **"Resiliency is the ability to sustain damage but ultimately succeed"**
- **"Resiliency is all about accepting that I will sustain a certain amount of damage."** NSA Director Admiral Mike Rogers:September 16, 2014

- The goal is to concurrently face the threats while maintaining critical functions essential for realizing system/application objectives
- Trust: System behaves predictably in contexts that are not anticipated
- Ability to operate in a multi-dimensional envelope

Magnitude of the Problem: Five-Minute Snapshot of In-and-Out Traffic within NCSA



(a)



(b)

Some challenges

- Repair rate, (being back within an hour, in real time …)
- What level and/or what criticality of service
- Other constraints: Situational awareness, change in system operating environments
- Multiple and rare events occurring together (e.g coordinated attacks)
- Assessment and validation
- Impact of policies cost model

How to Achieve Resiliency?

- By design to
  - auto-detect cyber-attacks;
  - isolate or interfere with the activities of a potential or actual attack;
  - recover a secure state and continue, or fail safely.
- By accounting for human in the loop
  - e.g., deceitful or malicious but entirely normal usage of the system
  - many current system and networks are generally complex cyber-physical-human systems

Learning from Reliable Systems?

Reliable/Dependable	Secure
Fault avoidance	Static analysis to identify and remove vulnerabilities and design flows
Errors escape even in the best designed and tested systems	
Runtime error detection	Continuous monitoring for identifying (diagnosing) abnormalities in system/application/user behavior
Systems/applications get compromised or fail	
Error Recovery	Repair actions in response to malicious attacks

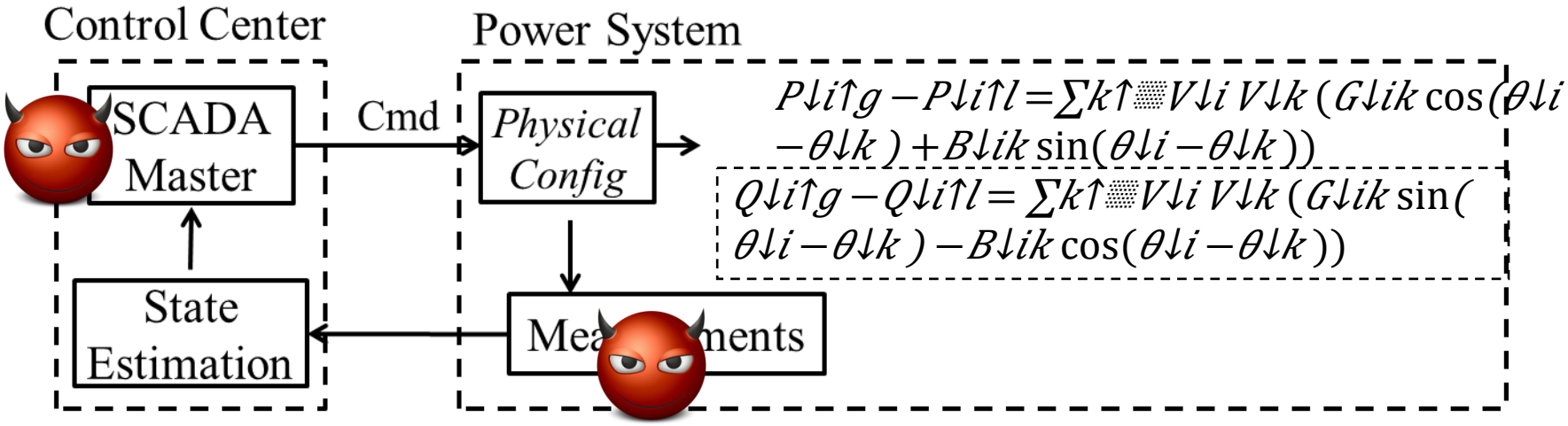
Predictive Security Metrics

- Develop security metrics and models capable of predicting whether or confirming that a cyber system preserves a given set of security properties in a given context
- Metrics for damage propagation, can you continue to provide the designed/planned/expected minimal level of service

- Hard because uncertain and variable nature of:
  - behavior of intelligent adversaries,
  - attractiveness of the target system,
  - impact of the architecture and design decision, and development process choices

Example: Challenge of Control-related Attacks in SCADA Systems

- ***Threat model***: *control commands*, if maliciously crafted, can directly change system's physical state
- ***Control-related attacks***: a sophisticated attacker can exploit system vulnerabilities and use a few maliciously crafted commands to put the system into insecure electrical states

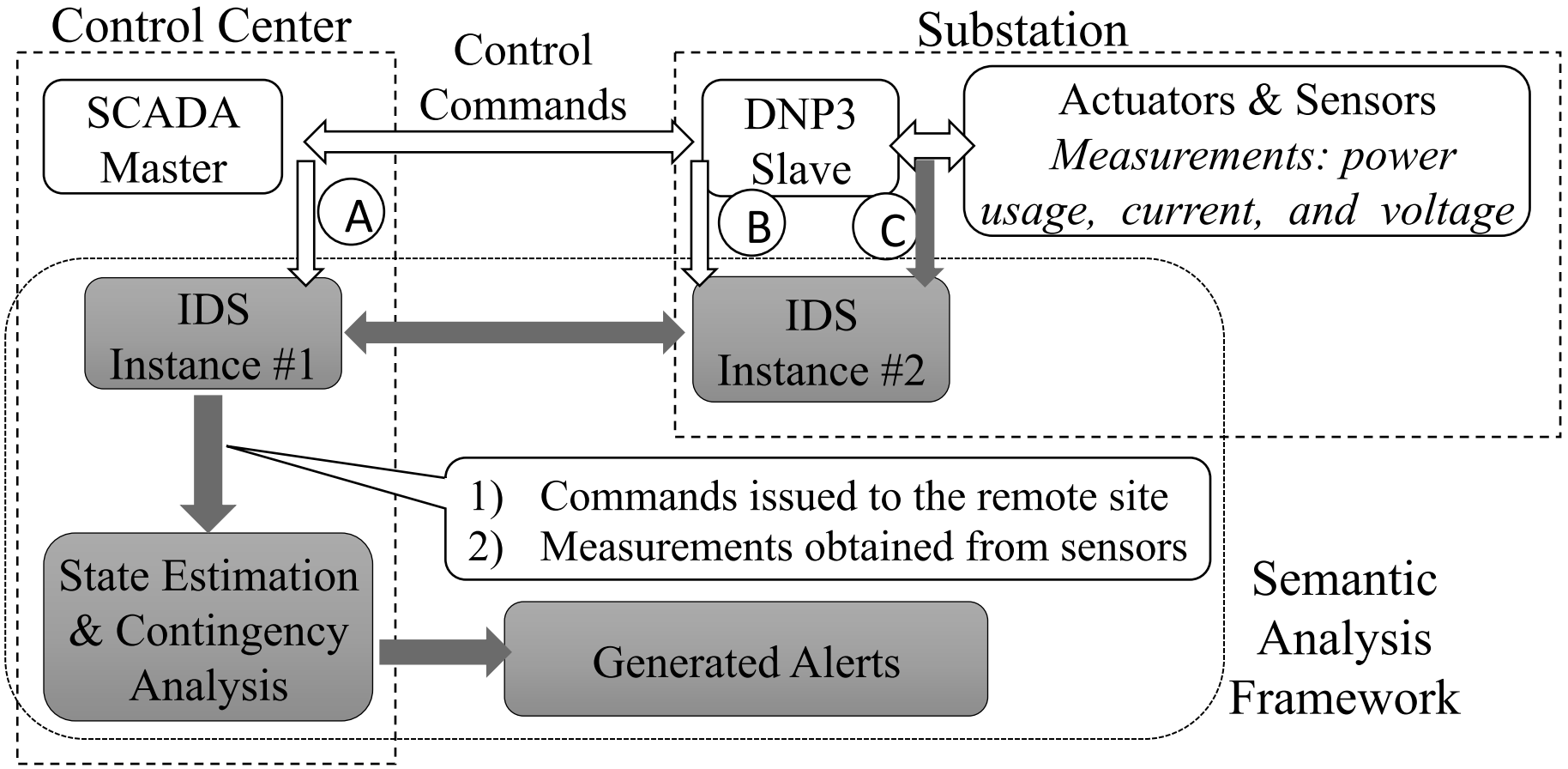


Why Is This Difficult?

- ***Hard to detect*** based solely on states of physical components
  - Classical state estimation and contingency analysis methods are performed periodically on small range of system changes
  - Measurements can be compromised during network communications
- ***Hard to detect*** based solely on network activities
  - Malicious commands may not generate a network anomaly

- **Need to understand semantics and interplay between the physical and cyber**

A Semantic Analysis Framework



Going Forward

- Focus Paper in lablet projects while addressing the broader issues
- Group meeting to firm up paper on or before the january meeting in NC
- Summer School on Resiliency?

Backups

A Real Multi-stage Security Incident at NCSA and Corresponding Factor Grap

Post-incident analysis of attacker actions:

1. Compromise a user account and log in from a remote location
2. Download, compile, and execute a privilege escalation exploit (CVE-2008-0600)
3. Inject credential collecting code (to harvest user credentials) into the node's SSHd server,
4. Restart the SSHd server

Our goals

Research methods for ***preemptive detection of attacks*** before the system misuse

Challenges

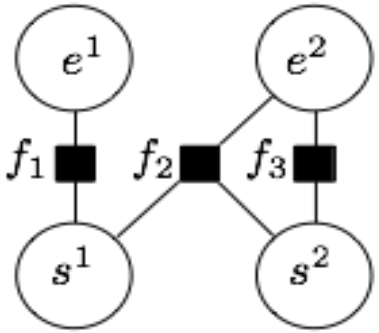
Attackers may enter the target system using stolen credentials
Defenders only operate on a partial knowledge on the attack
Defenders must rely on semantics of event logs: difficult to correlate with attacker's actions
Examining an event in isolation may not be sufficient to make decisions

Factor Graph Representation of an Example Incident at NCSA

Variable nodes
(defined based on the data from security/system logs)

e^1: download sensitive
e^2: restart system service

s^1: user state when observing e^1
s^2: user state when observing e^2



Factor functions/nodes
(defined based on the data from security/system, knowledge of the system, security experts opinion)

$$f_1 = \begin{cases} 1 & \text{if } e^1 = \textit{download sensitive} \\ & \& \; s^1 = \textit{suspicious} \\ 0 & \textit{otherwise} \end{cases}$$

$$f_2 = \begin{cases} 1 & \text{if } e^2 = \textit{restart service} \\ & \& \; s^1 = \textit{suspicious} \\ & \& \; s^2 = \textit{malicious} \\ 0 & \textit{otherwise} \end{cases}$$

$$f_3 = \begin{cases} 1 & \text{if } e^2 = \textit{restart sys service} \\ & \& \; s^2 = \textit{benign} \\ 0 & \textit{otherwise} \end{cases}$$

The factor function f_2 can improve detection accuracy by incorporating prior information