SEMANTIC SECURITY ANALYSIS OF SCADA NETWORKS TO DETECT MALICIOUS CONTROL COMMANDS IN POWER GRID

ZBIGNIEW KALBARCZYK

EMAIL: KALBARCZ@ILLINOIS.EDU

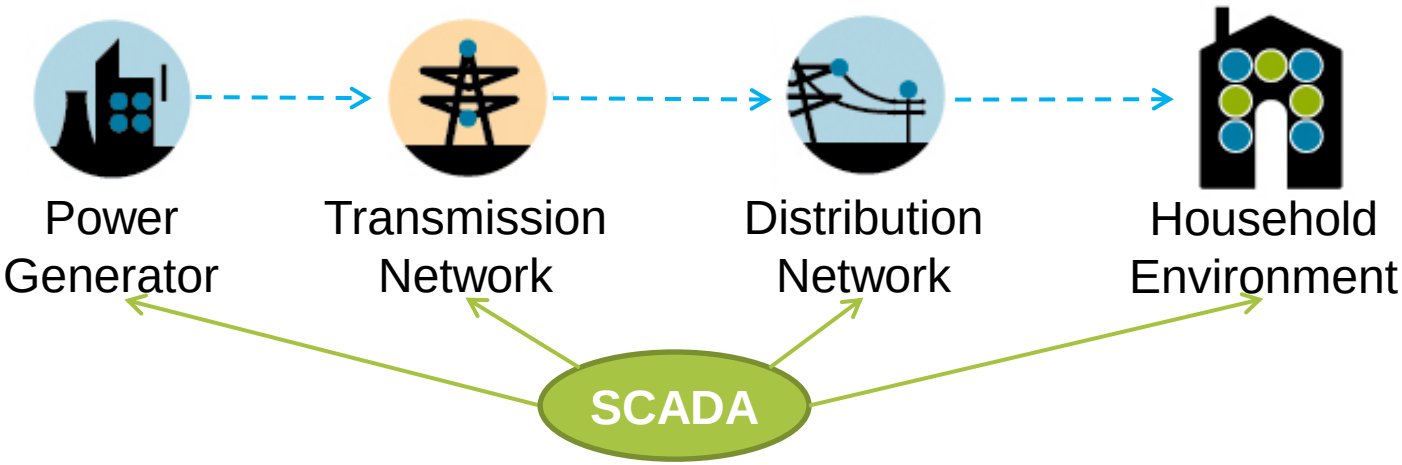
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

JANUARY 2014

Outline

- Problem
- Attack Model
- Attack Scenario
- Semantic Security Analysis Framework
- Evaluation
- Conclusions

Power Grid Operations

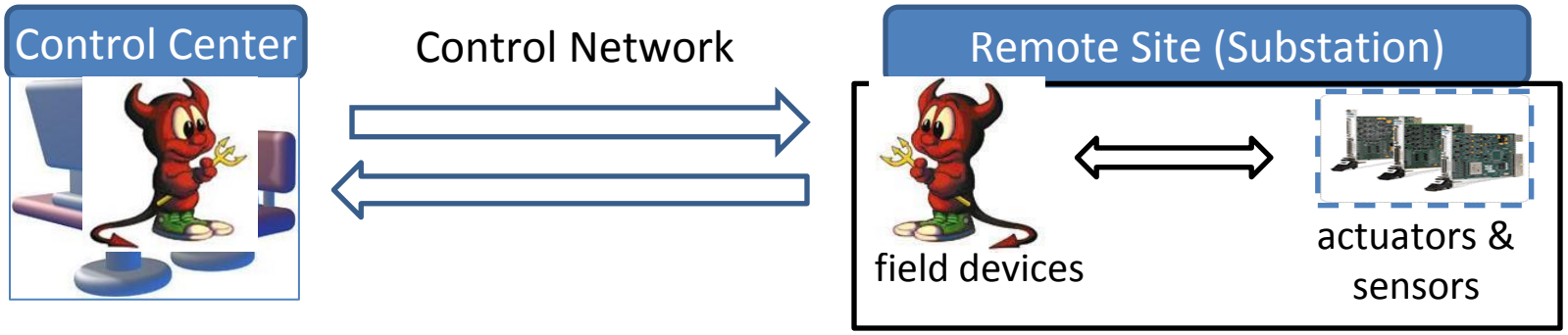


- ***S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition (SCADA)* system
  - Monitor and control geographically distributed assets in industrial control environment, e.g., power grid or gas pipeline
- To boost control efficiency, SCADA systems integrate proprietary protocols into IP-based network infrastructure

Challenges of Control-related Attacks

- Control-related attacks: a sophisticated attacker can exploit system vulnerabilities and use a single maliciously crafted control command to bring system in insecure/unsafe state
  - ***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

Attack Model

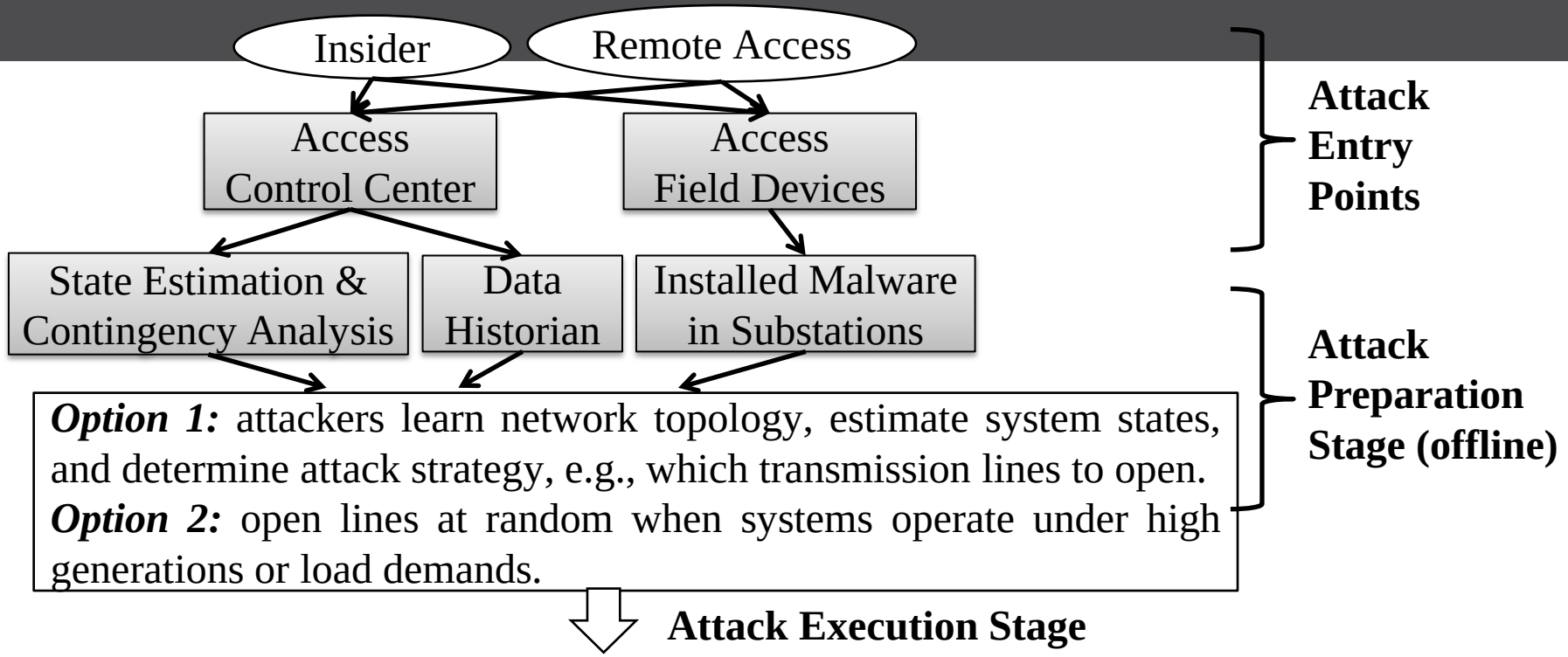


- We *DO NOT TRUST* "intelligent" devices
  - Computing devices in the control center
  - Intelligence field devices in substations
  - Control network
- We *TRUST* measurements of power usage, current, and voltage directly obtained from sensing devices in substations
  - Concurrent physical accesses to and tampering with a large number of distributed sensors is hard to achieve in practice

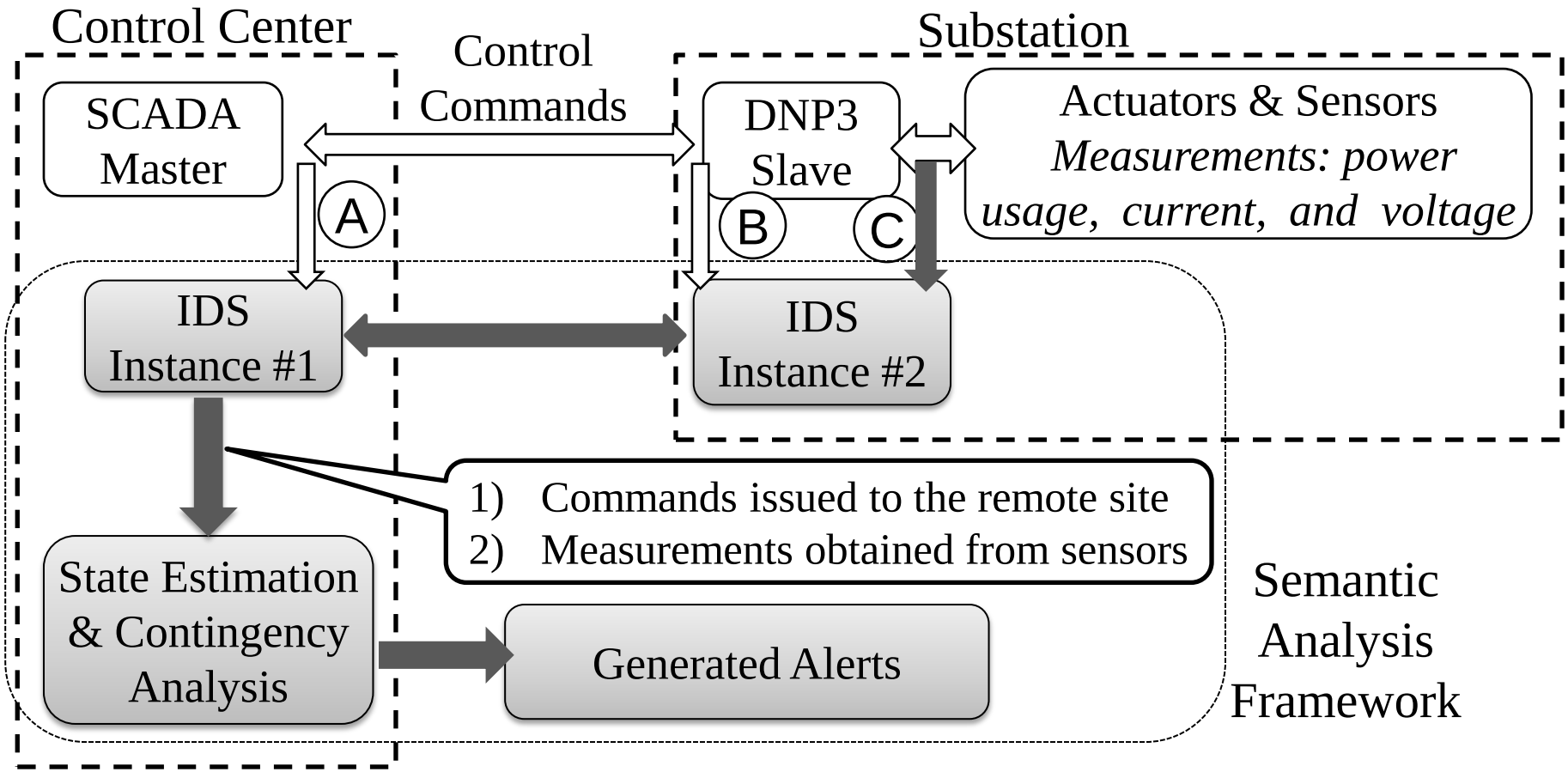
Attack Scenario Assumptions

- An attacker can penetrate the intelligent components in the power system
- An attacker can issue maliciously crafted control commands that can put the power system into an insecure state

Attack Scenario Stages



Semantic Analysis Framework



Semantic Analysis Procedure

- Extract parameters of control commands from SCADA network packets
- Obtain trusted measurements from sensors in substations
- Trigger contingency analysis to estimate consequences of executing the commands carried by the network packets
- *Response to detected intrusions*

- The semantic analysis framework do not impact the normal functioning of SCADA system
  - no additional delay introduced in the communication between the SCADA and substations

Monitor Control Commands

- Bro intrusion detection system (IDS) is adapted to analyze network packets transmitted using the DNP3 protocols
- Network IDS distinguishes critical commands from non-critical ones
  - *Critical commands:* commands that can operate physical devices and potentially change the system state

Command Type	Description
Read	Retrieve measurements from remote substations, e.g., read binary outputs
Write (Critical)	Configure intelligent field devices, e.g., open, edit, and close a configuration file
Execute (Critical)	Operate actuators or sensors, e.g., open or close a breaker connected to a relay

Evaluation Testbed Setup

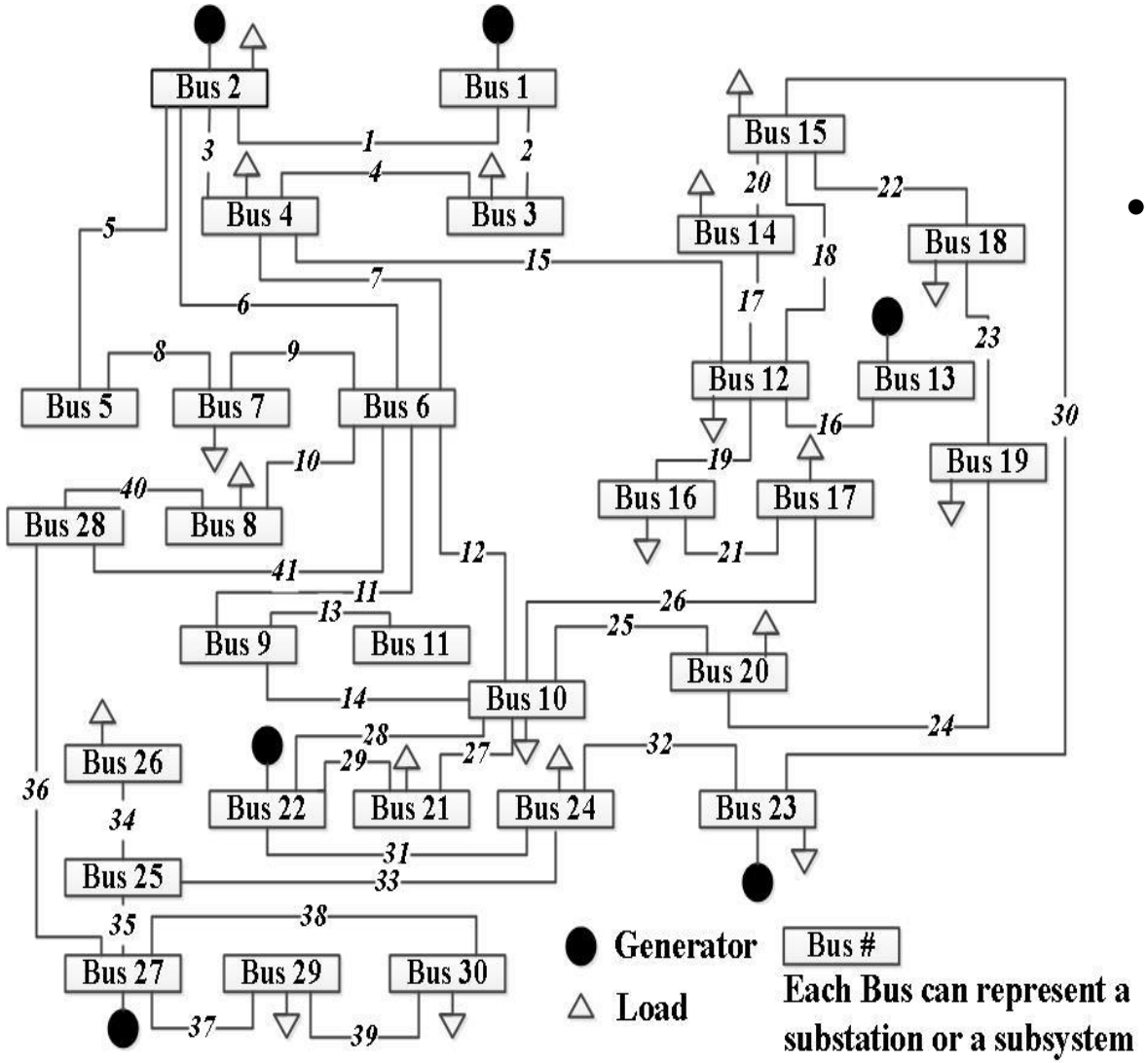
- Hardware and system software
  - An Intel i3 (3.07 GHz) quad-core; 4 GB RAM, running Linux OS
- Application software
  - *SCADA master* and *DNP3 slave* implemented using open source DNP3 library
  - Produce synthetic DNP3 network traffic
- Intrusion detection system
  - *Bro IDS* with integrated *DNP3 analyzer* to monitor network traffic
  - *Matpower*, an open source Matlab toolbox for power flow analysis

Effects of Malicious System Changes

- SCADA master issues DNP3 network packets to change power system states
  - The traffic includes network packets, representing *read*, *write*, and *execute* commands
  - Include the maliciously crafted commands
  - IEEE 30-bus system analyzed

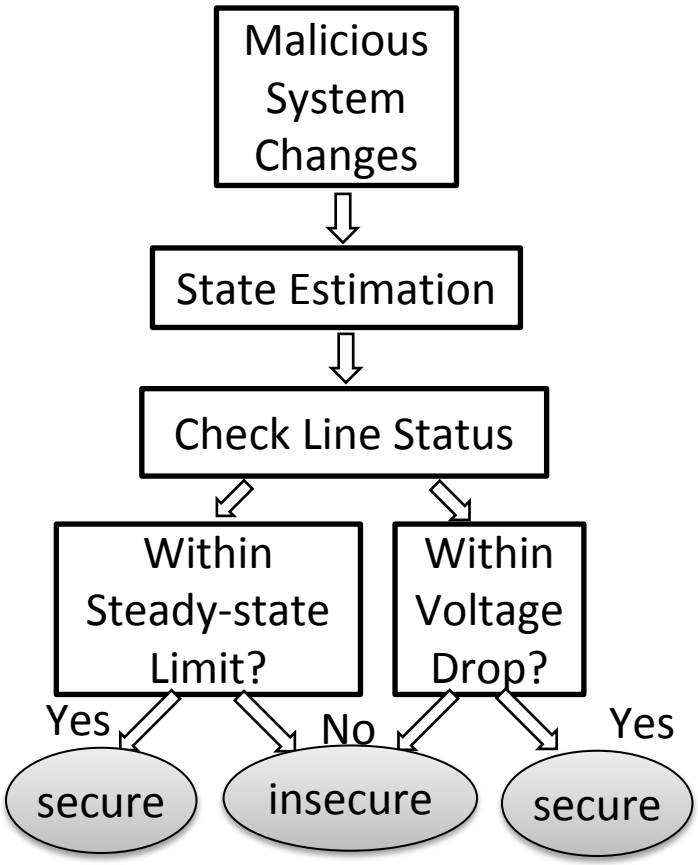
Cmd Type	Description	Event Pattern
Read	Request to read (i) static data and (ii) event data from relays	Periodic event with interval of 1 second
Write	Request to (i) update the static configuration file and (ii) open/close an application in a relay	Poisson process with average command arrival interval of 50 seconds
Execute	Request to open/close a breaker of a relay	Poisson process with average command arrival interval of 100 seconds

IEEE 30-bus System



- ***Malicious changes***
  - Increase generation (at bus 2, 13, 22, 23, and 27) by 50%
  - Increase load demand by 50%
  - Open 3 transmission lines at random
  - All changes simultaneously

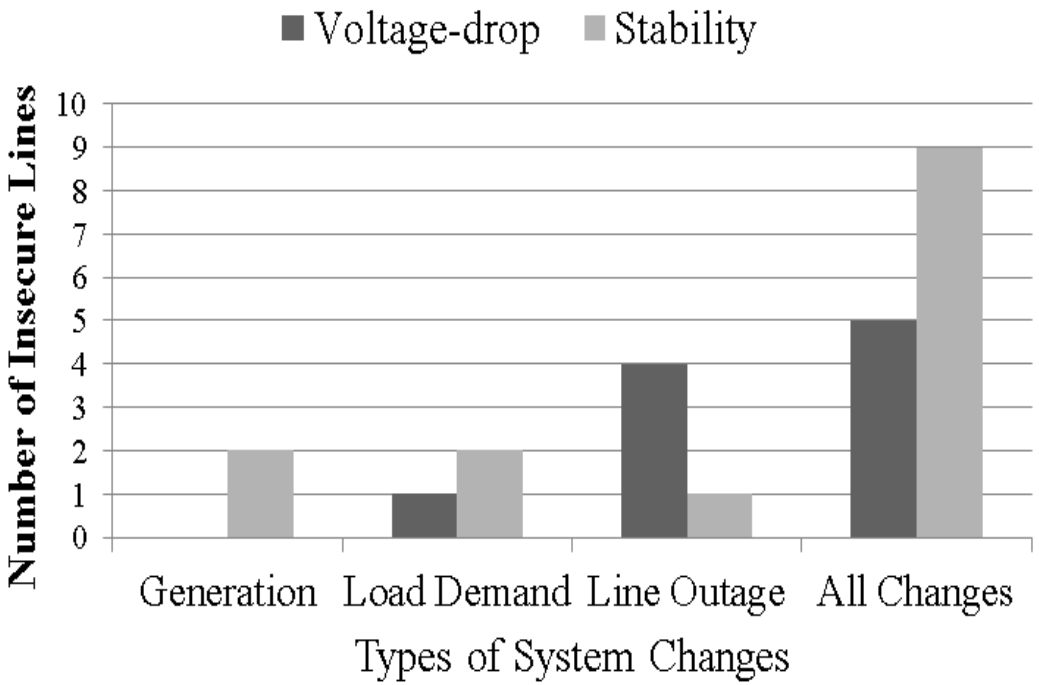
Procedure to Check System State



- ***Check line status***
  - ***Voltage drop limit*** – the voltage at the receiving end (VR) and at the sending end (VS) of a single transmission line should satisfy the operational condition VR / VS ≥ 0.95
  - ***Steady-state stability limit*** – the maximum power that a line can carry.
- ***Security Metric***
  - Number of insecure lines

Effect of System Changes

- Coordinated system changes (i.e., combination of increase in generation and load demand, and line outage) put up to 9 additional lines in insecure conditions



- To escape detection an attacker may want to avoid making changes to many physical components
  - attack when the system is most vulnerable, e.g., in presence of already high load demand
  - opening a few transmission lines may be sufficient to create a blackout

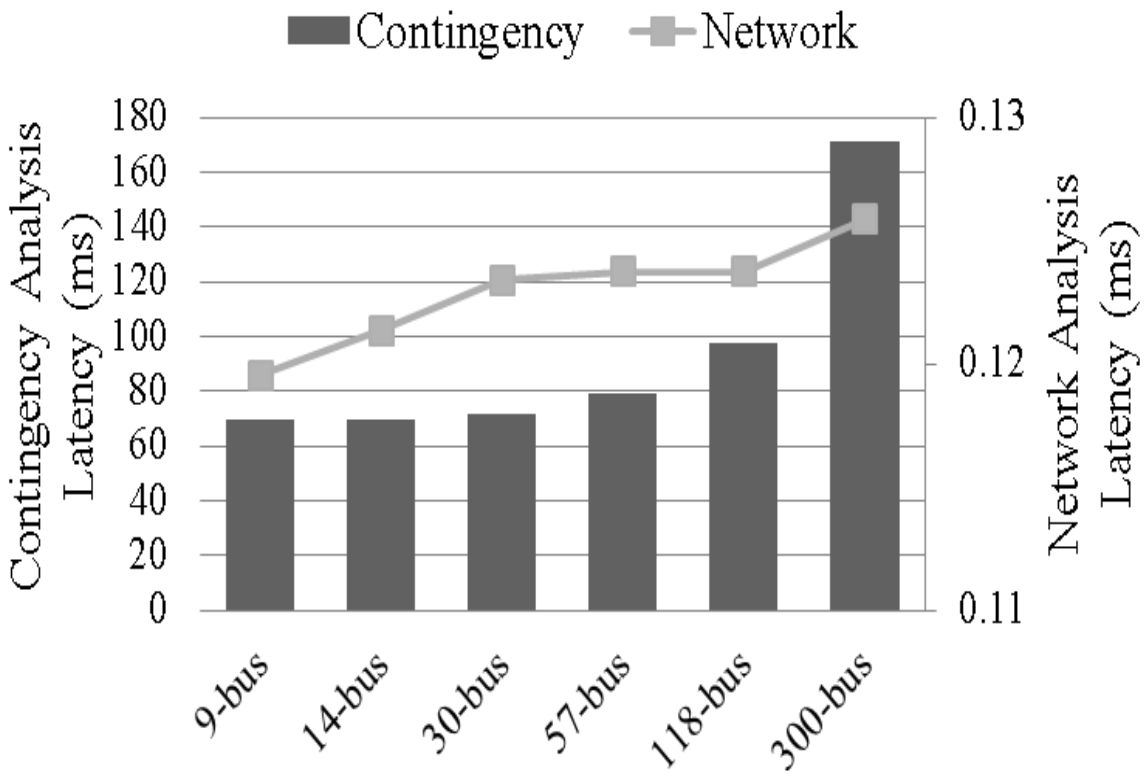
Performance Evaluation: Setup

- SCADA master is configured to simulate 24 hours of operations
  - 77,000 read commands
  - 1,800 write commands
  - 900 execute commands

- Measurements
  - the average execution time of network monitoring, e.g., filtering out noncritical commands and extracting parameters of critical ones
  - the time to carry on contingency analysis for different size test systems

Performance Evaluation: Results

Execute the contingency analysis to estimate consequences of executing a command



monitor critical commands, extract their parameters and deliver to contingency analysis software

The time to estimate consequence of executing a command (*~100ms*) is almost three orders of magnitude higher than the time of the network monitoring (*~0.1 ms*)

Does Measured Performance Allow Timely Semantic Analysis on Critical Commands?

- *Yes !*
- Network traffic involved to carry critical commands in power systems is still low
  - many critical commands to operate substation devices are issued manually
  - the interval between control commands are on the order of seconds (or minutes)
- There is a limited number of types of critical commands
  - ignore uncritical commands to reduce the frequency of the semantic analysis

Conclusions

- Show that in the Power Grid SCADA, an attacker can use legitimate, but maliciously crafted, commands to put the power system in insecure state
- Propose a semantic analysis framework based on an IDS extended with
  - network packet analyzer
  - power flow assessment tools
  - to (preemptively) estimate the execution consequence of a command and prevent the system damage
- Evaluated the approach on the IEEE 30-bus system
  - the semantic analysis provides reliable detection of malicious commands with a small overhead

Future Work

- Improve performance of the state estimation
  - consider different strategies as to *how* and *when* to re-compute the system state
- Investigate response to a detected intrusion
  - e.g., postpone a command

Acknowledgments

- Hui Lin
- Adam Slagell
- Peter Sauer
- Ravishankar Iyer

- Our sponsors: DOE, DHS, and NSF

Current Status of the Software

- The DNP3 analyzer is already included in the Bro IDS official branch which you can download at: http://www.bro.org/download/index.html
  - The source code of the analyzer can be found at: bro/src/analyzer/protocol/dnp3