Security Architectures for Dealing with the Unknown

Saurabh Bagchi
The Center for Education and Research in Information Assurance and Security (CERIAS)
School of Electrical and Computer Engineering
Purdue University


Joint work with: Eugene H. Spafford, Guy Lebanon

Problem Statement

- Ways to make a system survivable
  - At design/implementation phase
    - Eliminate vulnerabilities
    - Policy/Access Control/Cryptography/Formal Verification
  - In production phase
    - Use IDS (system logs checking/network packet sniffing/virus, worms scanning, detecting files modifications…) to identify misuses/anomalies
    - Perform incident/intrusion response (IRS) to detected misuses/anomalies
- Focus Areas:
  - Configuration of security sensors so that prompt situation recognition is achieved
  - Automated intrusion/incident response based on the sensor inputs

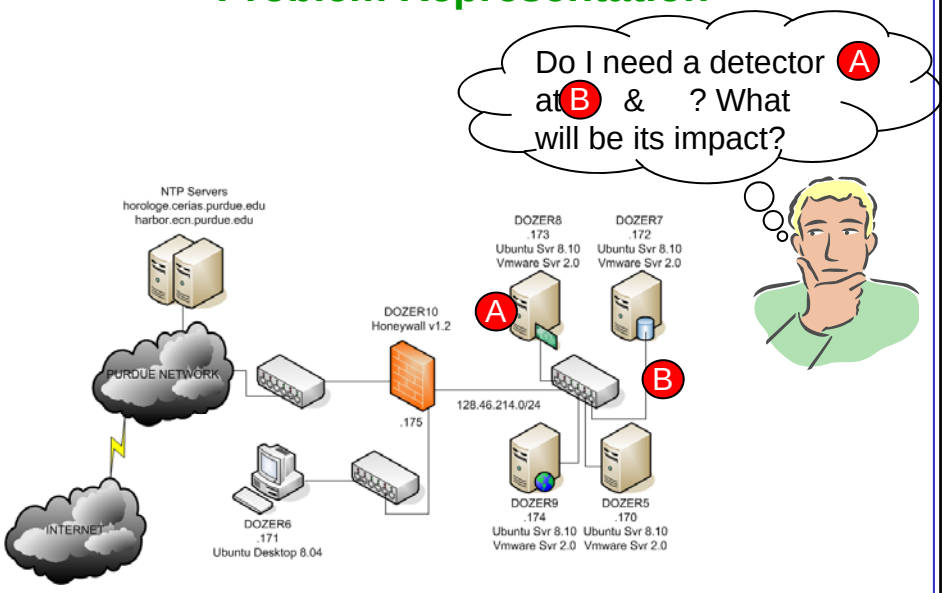
Problem Statement

- We want to perform secure configuration and intrusion response in the face of fast-changing and therefore unknown threats
- We want to learn from past behavior
  - But not overlearn
- We want to grow our knowledge structures with runtime information
  - But not learn untruths
- We want to perform the learning at runtime
  - This implies expensive batch mode processing is out

Motivation: Secure Configuration

- Placement and choice of detectors are more an art than a science, relying on expert knowledge
- Impact of choice is significant on the accuracy and precision of overall detection function
- More (detectors) is not always better
  - Economic cost
  - Administration cost
  - Performance cost

Problem Representation

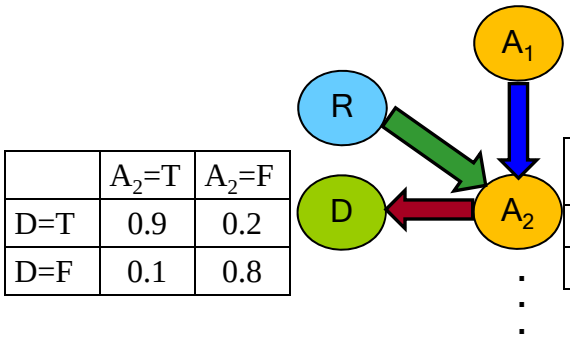


Current State of the Art: Placement of detection sensors

- Individual detector quality (false alarm-missed alarm) is considered
- Dependencies/overlap between multiple detectors not considered
- Cost analysis (maintenance, acquisition, performance, etc. costs) done qualitatively

Our Solution Approach: Detector Placement (SMARTS)

- Bayesian network used to model the causality in the network



	A_2=T	A_2=F
D=T	0.9	0.2
D=F	0.1	0.8

	R=F, A_1=T	R=T, A_1=T	R=F, A_1=F	R=T, A_1=F
A_2=T	0.90	0.15	0.25	0.05
A_2=F	0.10	0.85	0.75	0.95

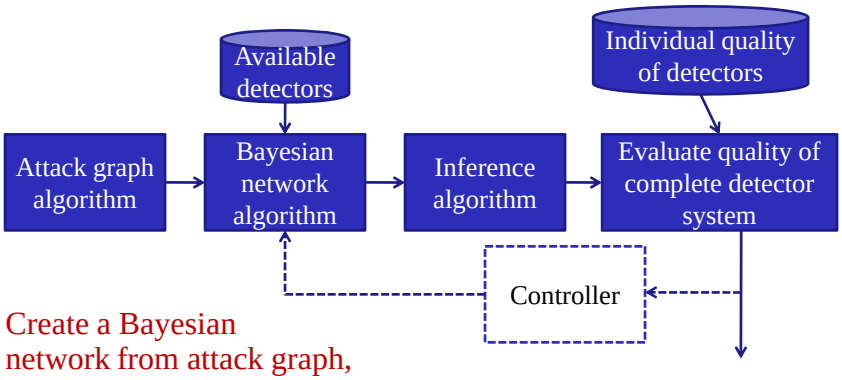
A Attack Step D Detector R Response

Our Solution Approach: Detector Placement (SMARTS)

- System goals that are important to the owner are quantified
  - Need to determine with confidence whether specific system security guarantees have been violated
  - Need to determine with confidence whether specific adversary goals have been achieved
- Inference on the Bayesian network performed through different choice and placements of detectors
- Heuristic-driven choice of one detector and its placement at a time
- Heuristic depends on individual detector quality and overlap with previously chosen detectors

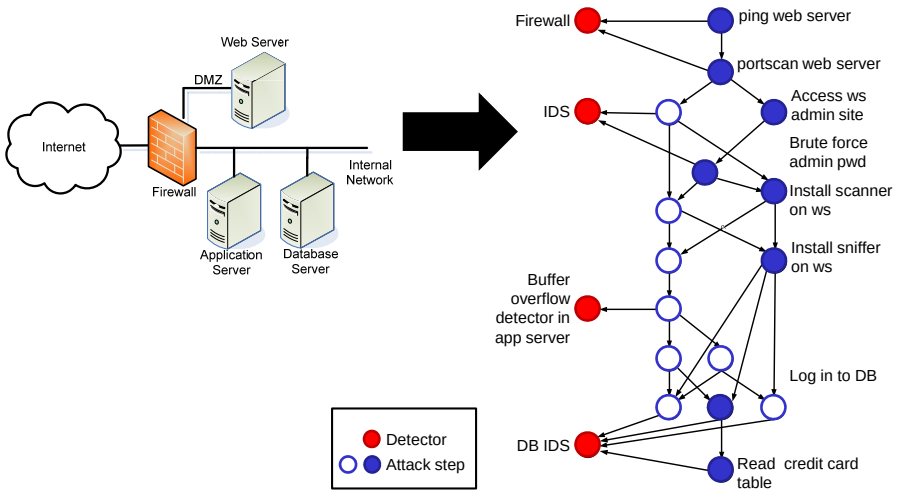
See our paper in RAID 08

Overall Framework



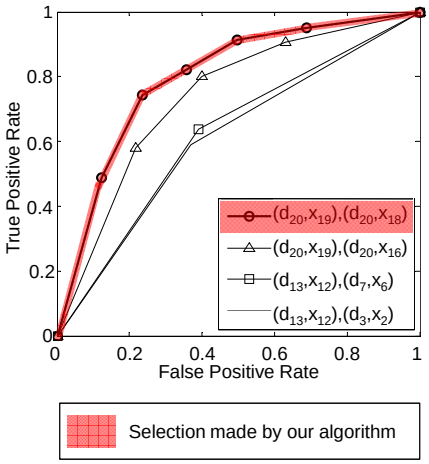
- Create a Bayesian network from attack graph, adding detectors and CPTs
- Run inference algorithm to compute possible combination of detectors, given an ultimate goal (from attacker perspective)
- Controller to adjust detector setting when network changes

Sample Network and Corresponding Bayesian Network



Impact on Choice and Placement of Detectors

- System: Three-tier web-based online service
- Objective: determine impact of selecting detectors and corresponding locations
- Performance of detector pair (selected from algorithm) is compared against randomly selected pairs



Why Intrusion Response System (IRS)?

- Intrusions/security breaches to computing systems occur
- A survivable system needs to provide functionality through intrusions
- Human intervention after IDS alert can be costly and slow
- IRS needed to take reports from IDS and carry out actions to counter the intrusion
- Existing examples of IRS
  - Anti-virus software which disables access to worm executables or files infected with virus
  - Iptables which terminates a session on matching a malware signature
  - Web browser blocks access to known malware / phishing websites

State-of-the-Art: Intrusion Response System

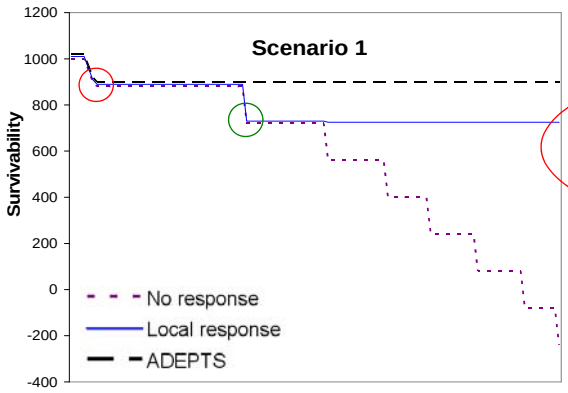
- Summary on existing IRS
  - Most of them are stand-alone and target one machine box only
  - They are tied through static mapping to specific IDS alerts
- IRS for Distributed Systems
  - An environment of multiple interconnected boxes with heterogeneous and cooperating services
  - Few general-purpose IRS solutions exist for distributed systems
  - The most common way is to use the stand-alone solutions separately and independently on the boxes
    - E.g., Have McAfee anti-virus software installed on the workstation boxes, and CISCO IPS on the edge routers

Our Solution Approach: Intrusion Response (ADEPTS)

- Short-term as well as long-term goals
  - Contain the current attack
  - Recover affected services to a functional state
  - Proactive defenses for future attacks
- Leverage distributed system's characteristics
  - Determine if the alert is false
  - Determine if the impact is worth responding to
- Learn from thy observations and mistakes
  - Calibrate prior responses
  - Learn characteristics of interactions in the system through past attacks
  - Quick customized responses to polymorphs of prior attacks

See our papers in SRDS 08, Computer Networks Journal 07, DSN 05

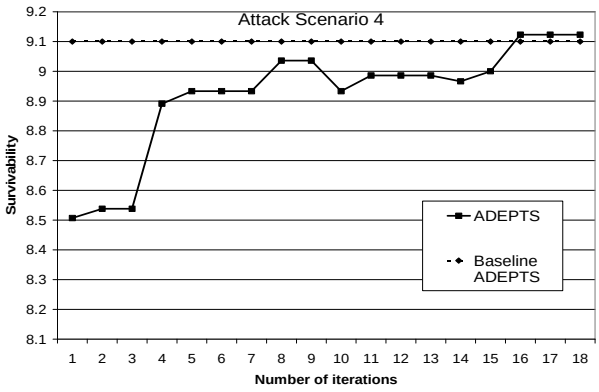
Resilience to multi-stage attack due to a distributed IRS



Scenario 1
Use php_mime_split (CVE-2002-0081) buffer overflow to insert malicious code into Apache.
'ls' to list webstore document root and identify the script code informing the warehouse to do shipments.
Send shipping request to warehouse and craft the request form so that a warehouse side buffer overrrun bug fills the form with a victim's credit card number.
Unauthorized orders are made.

Experiment – Incorrect Initial Conditions

- ADEPTS where initial settings (effectiveness of responses, IV values, etc.) are incorrect, say due to inexperienced sysadmin

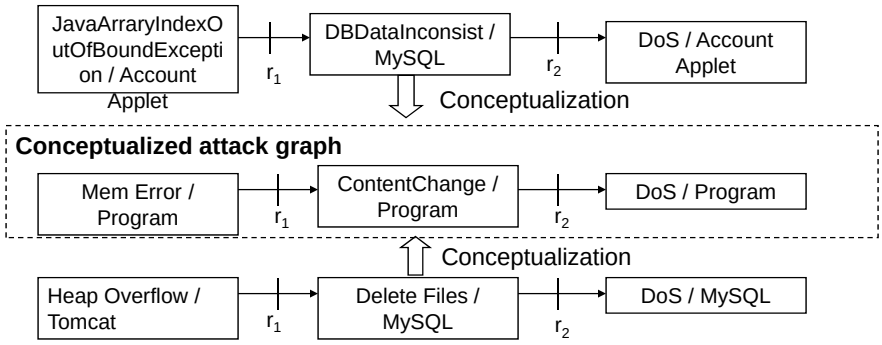


After 16 iterations of the attack, the effect of incorrect initial parameters disappears

Responding to the Unknown

- Zero-day attack
  - Knowledge of the steps in the attack does not exist in the IRS
- Current solution: Take a drastic response, such as disconnecting the service
- Problem:
  - May be reacting to spurious alarms
  - Cannot learn from the spread of the attack
- Our solution approach:
  - Abstract the specifics of the attack
  - At a higher level of abstraction, map the attack to a previously seen attack
  - Use the learning on the previous attack to guide the responses for the current zero-day attack

Responding to the Unknown: Example



Responses: r_1: Disable connection from tomcat/applet to MySQL; r_2: Rollback to last data files checkpoint

- Challenges: (1) High similarity does not necessarily give you the best response; (2) To what level should the graph be conceptualized

Secure Configuration and Response: Unified

- Consider that online containment and recovery are critical
- Then the value of the detector also depends on the availability of a response
- The choice of detectors is therefore driven by
  - Availability of a response
  - Effectiveness of the response
- Our solution
  - Use a Decision Network to model attack steps, detectors, and responses
  - Use Bayesian inferencing to calculate a utility function
  - Greedy selection of detectors and within that greedy selection of responses

Goals of Ongoing Work

- Secure Configuration Management
  - Detector placement is a specific example of security configuration
  - Tool should detect (when insecure configuration is introduced) and diagnose (which component has been mis-configured)
  - Tradeoffs exist between security of configuration and usability
  - Tool must not make arbitrary decisions on this spectrum
- Automated Intrusion Response
  - Resilience to zero-day attacks through more effective responses (i.e., less drastic than rebooting the servers)
  - Correlation of multiple detectors to increase confidence that an attack is underway before responding