# Responses to Cyber Attacks in Distributed Systems

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

Supported by:
NSF, Lockheed Martin, NEHRP

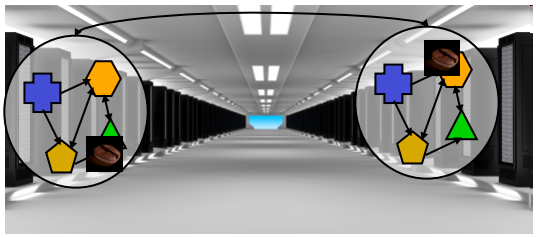
Joint work with: Eugene H. Spafford, Guy Lebanon



## Outline

- **Problem Statement**
- Solution Directions
- Some Promising Solutions
- Ongoing Challenges

## Defending Distributed Systems

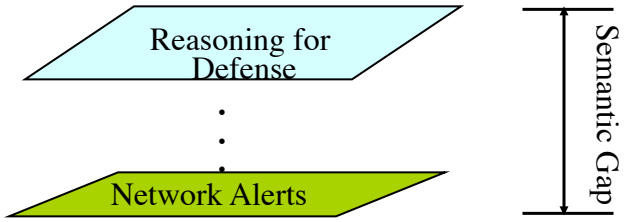


- Large-scale distributed systems to defend
  - Heterogeneous third-party services
- Lots of points for attacks
  - Lots of points to introduce cybersecurity mechanisms
- Interactions between the services allow for attack escalation

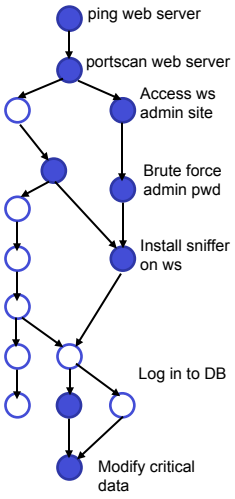


## Drowning in a Sea of Alerts

- Large distributed systems get tons of alerts
  - Up to 20,000 per day
- Many of these are false alarms



# Fast Moving Attacks

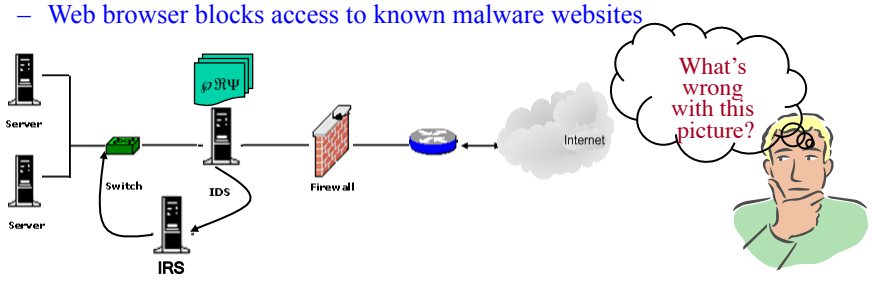



- Multi-stage attacks
  - Compromise outward facing services
  - Use transitive trust and privilege escalation
  - Compromise internal services
  - Access crown jewels
- Attack progresses in machine time, rather than human time
- Examples: Worms and other self-propagating malware



# Signature-based Responses

- Intrusion Response Systems (IRS) take reports from IDS and carry out actions to counter the intrusion
- Many examples of IRS
  - Anti-virus software 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 websites



## Dealing with Zero-Day Attacks

- Zero-day attacks are difficult to deal with through signature-based mechanisms
  - They exploit unknown vulnerabilities
  - Their path of attack spread is not known *a priori*
- Challenges for zero-day attacks
  - Exact matching of mechanics of attack step does not work
  - A reactive approach to security allows devastating zero-day attacks to get through
  - Learning-based approaches are predicated on exact matches and therefore do not work well



## Outline

- Problem Statement
- **Solution Directions**
- Some Promising Solutions
- Ongoing Challenges

## Solution Directions

- We want to perform secure configuration and intrusion response in the face of threats that are fast-changing and therefore unknown

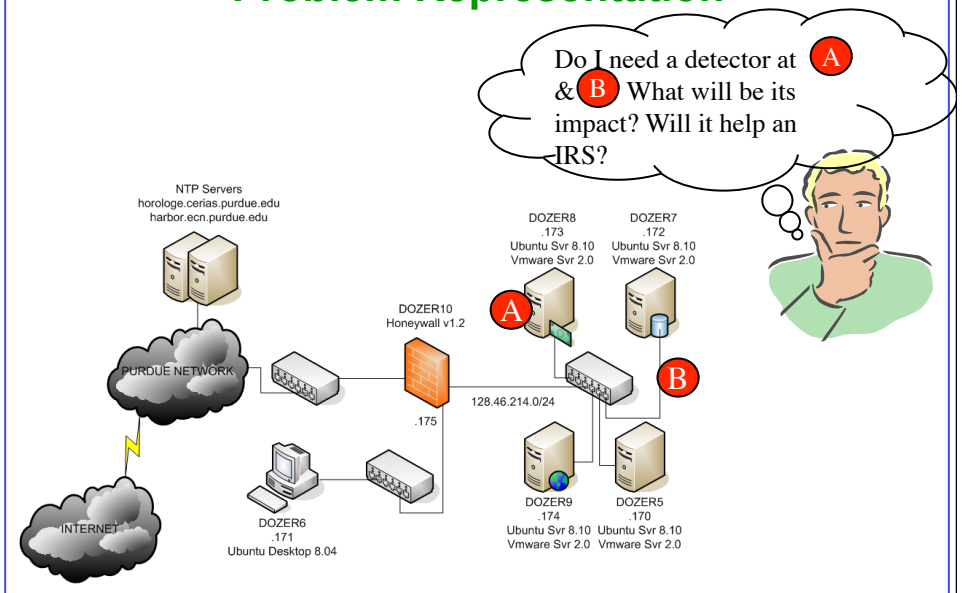
1. We want to learn from past behavior
   - But not overlearn
2. We want to grow our knowledge structures with runtime information
   - But not learn untruths
3. We want to perform the learning at runtime
   - This implies expensive batch mode processing is out
4. We do not want to rely only on signature-based security
   - Abstractions of attack steps are useful



## Outline

- Problem Statement
- Solution Directions
- **Some Promising Solutions**
- Ongoing Challenges

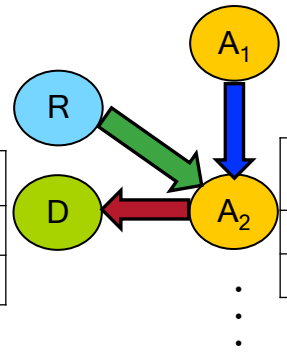
# Problem Representation






# Our Solution Approach: Detector Placement (SMARTS)

- Bayesian network used to model the causality in the network

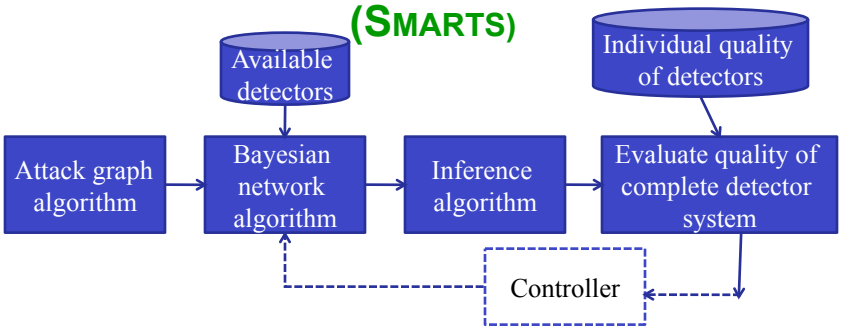



| | $A_2$=T | $A_2$=F |
|---|---|---|
| D=T | 0.9 | 0.2 |
| D=F | 0.1 | 0.8 |

| | 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)




- 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
- Controller to adjust detector setting when network changes



## Adaptive to Current Threat Environment

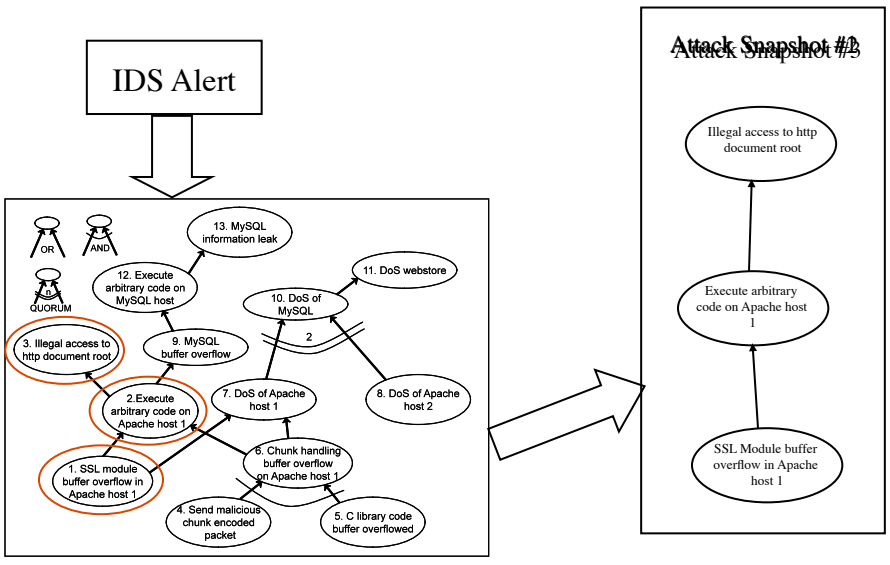
- It is expensive to turn all sensor rules all the time
  - *Example:* Snort default rule set has > 9,000 single step attack rules, in 73 categories and takes > 5 sec to match all of them
- Approach:
  - Perform damage assessment – currently through Bayesian inferencing
  - Damage assessment indicates
    - Which components are likely compromised but needs further evidence to determine with high confidence
    - Based on attack spread, which components are likely to be compromised
  - Sensor rules are activated based on results of damage assessment
- Responsive to changes in system
  - Incremental inferencing when some parts of system change

# 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

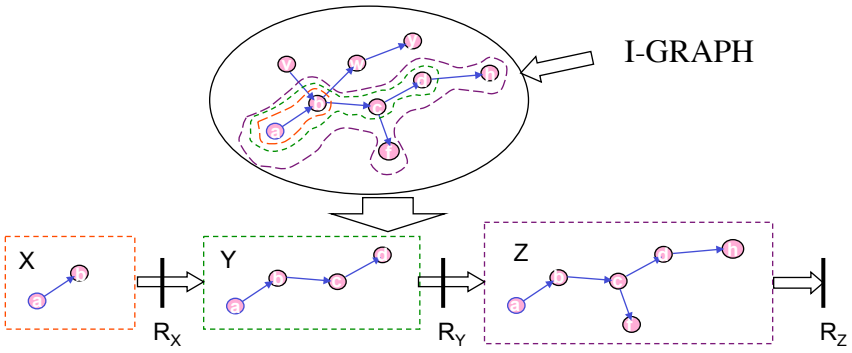


# Attack Snapshot



# Dynamics between attack and responses

- Successive attack snapshots created for incoming IDS alerts

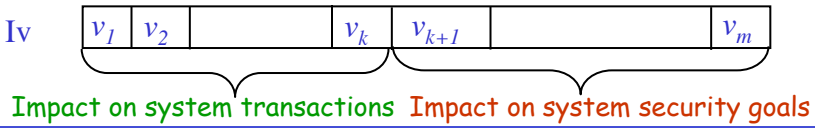



- Assuming an attack includes three "snapshots" X, Y, and Z
- Each snapshot includes I-GRAPH nodes which have been achieved as part of the attack thus far
- Following each snapshot $k$, SWIFT determines a response combination $R_k$ (a set of response actions) to deter the escalation



# Impact Vector

- A system has transaction goals and security goals that it needs to meet through the time of operation
  - Example: provide authentication service & preserve privacy of sensitive data
- Attacks are meant to impact some of these goals
- Deployed responses also impact some of these goals
  - For example, by temporarily disabling some functionality for legitimate users as well
- Assume the impact can be quantified through a vector Iv
  - Each element in the Iv corresponds to the impact on each transaction/ security goal $\in [0, \infty]$



## Optimality of Response Actions

- We formally define the cost for a response combination (a set of response actions) $RC_i$ as:

$$Cost(RC_i) = \sum_{n_k \in \text{I-GRAPH}} Iv(n_k) \cdot \Pr(n_k) + \sum_{r_k \in RC_i} Iv(r_k)$$

$Iv(n_k)$ : Impact from reaching an attack step node $n_k$

$\Pr(n_k)$: Probability of reaching node $n_k$

$Iv(r_k)$ : Impact from deploying the response $r_k$

- The response combination $RC_i$ is said to be optimal for a given attack if it achieves the minimal $\text{Cost}(RC_i)$
  - In ADEPTS, optimality achieved "per node and per out-going edge"

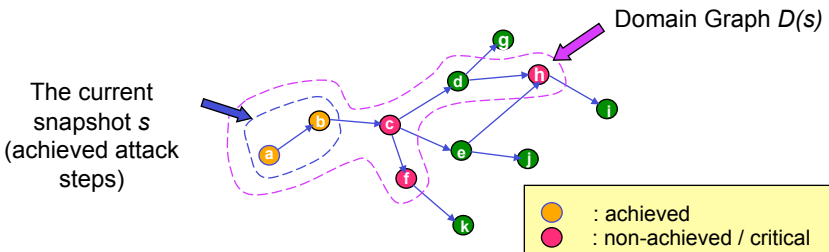




## Domain Graph

- Limit the response search space for a snapshot $s$ to a subset of I-GRAPH, namely the **Domain Graph** $D(s)$
- $D(s)$ includes critical nodes from I-GRAPH
  - A node $n$ is critical if $|\text{Prob}(n)*\text{Iv}(n)|$ is greater than a given threshold
  - Also include nodes on the path leading to critical nodes

$$Cost(RC_i) = \sum_{n_k \in D(s)} Iv(n_k) \cdot \Pr(n_k) + \sum_{r_k \in RC_i} Iv(r_k) \text{ is minimized.}$$

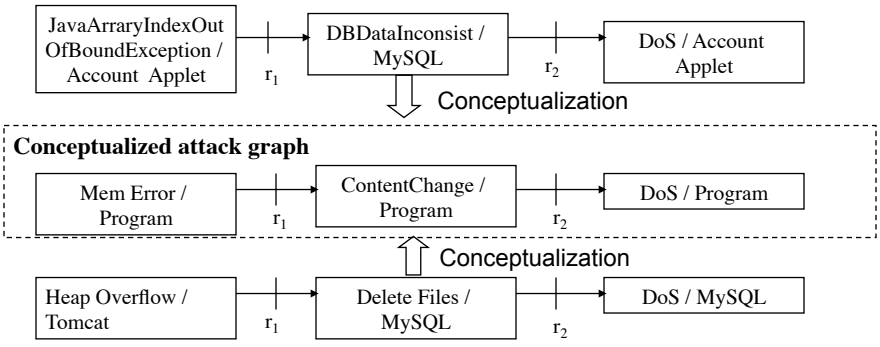


## Responding to the Unknown

- 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

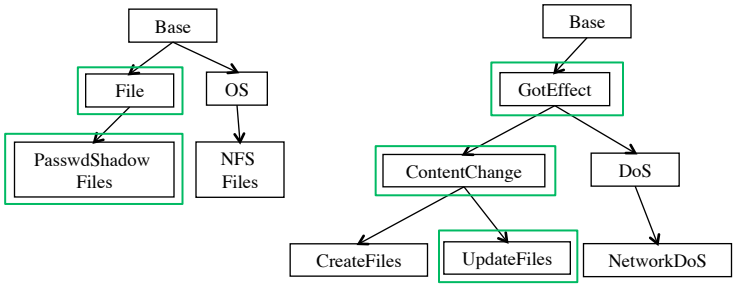



Responses: $r_1$: Disable connection from tomcat/applet to MySQL; $r_2$: Rollback to last data files checkpoint

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

## Utilize History from Similar Attack

- How to calculate similarity between two attacks?
  - Inheritance hierarchy for components, detector alerts, and connections




  - Calculate distance for each node and each connection
- Compute graph edit distance
  - Conceptually, the sequence of steps to convert one graph to another
  - Through addition, deletion, or modification of nodes and connections

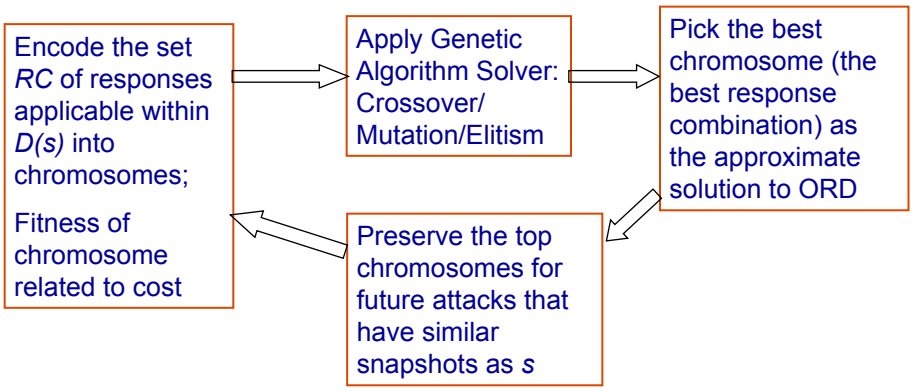


## Utilize History from Similar Attack

- Acquire from the similar attack
  - Effectiveness Index (EI) values of responses
  - Edge Propagation Factor (EPF) values of edges
  - Effective Response Combinations
- Efficient search through space of prior attacks
  - Attack similarity is defined to follow metric space conditions: $d(x, x) = 0; d(x, y) = d(y, x); d(x, y)+d(y, z) \geq d(x, z)$
  - Prior work allows for efficient storage and search through attack template library
  - Disjoint parts of multiple attacks can be used in responding to the current attack

## Approximate O.R.D. with Genetic Algorithm

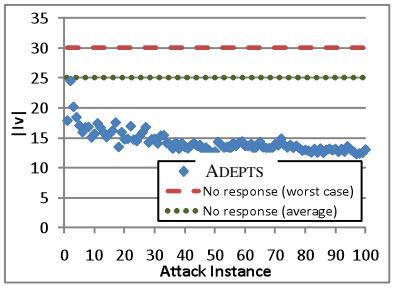
- We proved Optimal Response Determination (O.R.D.) to be NP-hard by mapping the Set Covering Problem to it



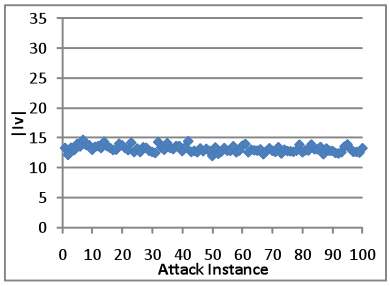



## Sample Result from ADEPTS

- Two distinct attacks: LLDoS and MalExec




LLDoS with no history




ADEPTS: LLDoS with prior history from MalExec

- With history even from a seemingly distinct attack, performance at the first attack instance is better
- ADEPTS learns from prior instances of the attack

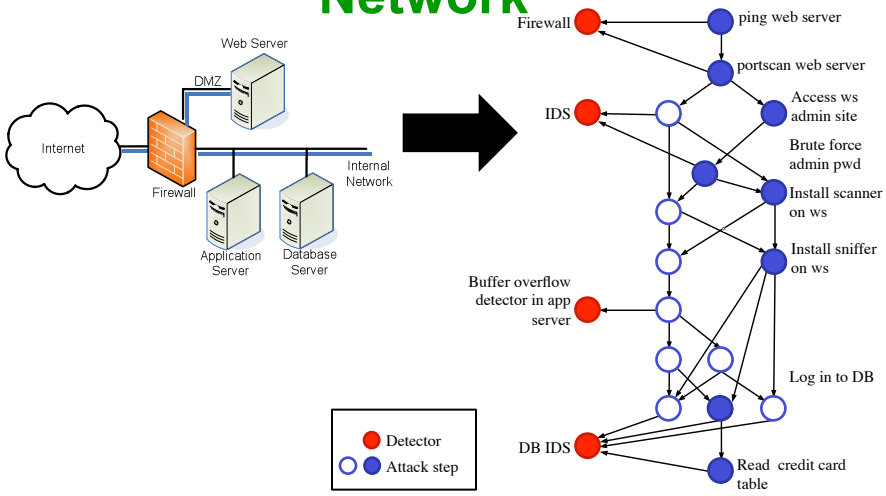
## Goals of Ongoing Work

- 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



## Backup Slides

# 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