Achieving High Survivability in Distributed Systems through Automated Response

Saurabh Bagchi, Guy Lebanon, Gene Spafford
Gaspar Howard, YuSung Wu, Bingrui Foo
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

PURDUE UNIVERSITY

Survivable Systems and Intrusion Response

- Ways to make a system survivable
  - At design/implementation phase
    - Eliminate vulnerabilities
    - Policy / Access Control / Cryptography / Software Engineering
      - Challenge : “User Friendliness” (e.g. everybody likes User Access Control in Windows Vista or SELinux ?)
  - In production phase
    - Use IDS to identify misu
      - system logs checkin
        sniffing / virus scanning / V
    - Perform incident/intrusion response
      - Containment and Recovery
      - Stay transparent under normal operations
      - Intervene only when attacks are detected

Intrusion Response System

(focus of this work)

-based root kit detection..

Existing Automated Response System

- Traditional Anti-Virus (AV) Product
  - Scan / Quarantine virus-infected files
- Host-based Intrusion Prevention System (HIPS)
  - An integration of (host-based) firewall, system-level action control, vulnerability detection and sandboxing on top of a traditional AV product.
  - Monitor malicious activities
    - virus, probing from network, attempt to modify critical entries in system registry, visiting phishing websites…
  - Response actions
    - Block access to known phishing websites
    - Quarantine infected files
    - Lock-up internet connection
    - Request user permission to continue on with suspicious activities
  - Norton 360, McAfee Total Protection, TrendMicro Internet Security Pro…

Existing Automated Response System

- Network-based IPS (NIPS)
  - A purpose-built hardware/software to inspect network traffic
    - Content-based detection
      - worm infections / hacks…
    - Rate-based detection
      - for denial of service attack
    - Protocol-analysis
      - existence of large amount of data in the User-Agent field of an HTTP request,…
  - Constantly engaged proactive response actions
    - Rate-limiting, traffic sanitization, IP address / port-number black/whilte-listing
  - Reactive response actions
    - Drop connection, terminate session, update firewall rules
  - Cisco IPS 4200 Series, 3Com Unified Security Platforms, Juniper SSG, …

Existing Automated Response System: Shortcomings

- Stand-alone systems / Minimal collaboration among IDS/ IPS boxes.
  - Attacks against distributed systems cause correlated damages to multiple system components.
  - Correlation of alerts improves both the detection accuracy and the understanding of an attack in distributed systems

Existing Automated Response System: Shortcomings

- Static mapping between detector and response action
  - Example: If "/bin/sh" is detected in network traffic (potential attempt to create a shell), then "black-list the source IP".
    - What if the response is not effective? What if it's a false alarm? What if the created shell only has limited privilege and is not really harmful?
- Pure NIPS or pure HIPS strategy is often not desirable
  - NIPS alone at the perimeter of a system
    - Limited view of attack manifestations
    - False alarm can cause degradation of system performance
    - Some organizations are interested in letting attack propagate through the system till a point when significant damage is imminent
  - HIPS alone inside the system
    - Rely on host data for detection
    - More intrusive to applications
    - Last line of defense

System Model

- BASELINE Model of Automated Response in Distributed Systems
  - A collection of (detectors, response actions) pairs :
    - $\{(D_1,R_1), (D_2,R_2),\ldots, (D_k,R_k), \ldots, (D_N,R_N)\}$
  - For each pair, a mapping $f_k : D_k \rightarrow R_k$
  - f_k is designed based on expert knowledge
- Proposed Model of Automated Response in Distributed Systems
  - The set of all the detectors D and the set of all the response actions R

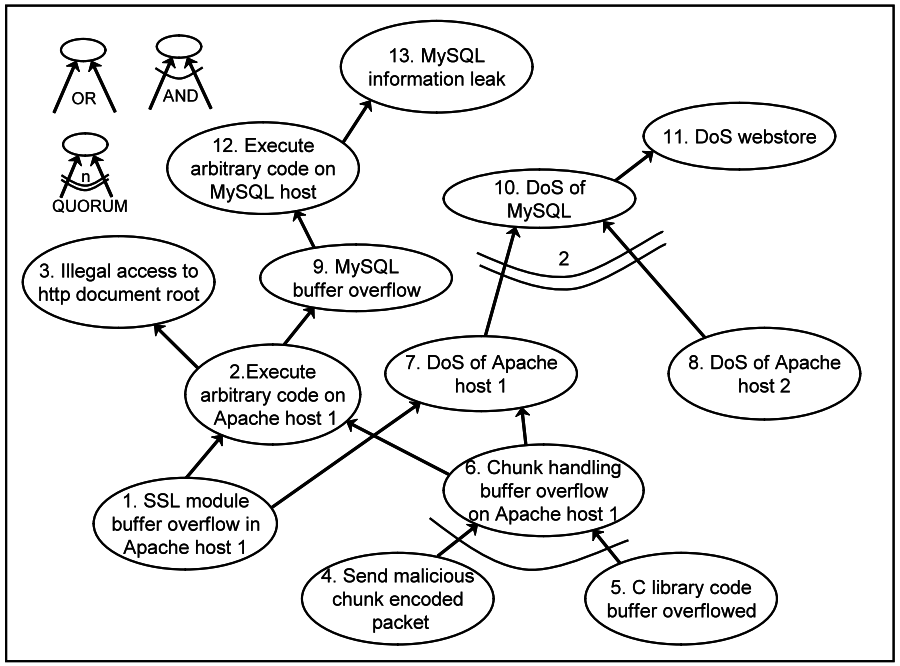
$$D=\bigcup_{k=1}^{N} D_k \quad , \quad R=\bigcup_{k=1}^{N} R_k$$

  - History of past attacks H
  - A mapping $f : (D,H) \rightarrow R$
  - f is designed to maximize expected system survivability based on the information accumulated in H and detectors D
  - f is designed to tolerate new types of attacks

Attack Model: Multi-stage Attack

- Attack originates outside the network
- Each step achieves certain privilege on a service
- Elevated privilege is used to compromise a connected service
- Ultimately some end goal is sought to be achieved

IDS Alert



PURDUE UNIVERSITY

Hypotheses

- The proposed model describes a set of responses, from which the expected system survivability is the upper bound of the expected system survivability from any set of responses generated from the BASELINE model.
- In a practical system, it is possible to identify cases when the proposed model yields a higher system survivability than the BASELINE model.
- It is possible that the use of history information in the proposed model can further improve system survivability.

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 $\in [0, \infty]$

$$\text{Iv} \quad v_1 \quad v_2 \quad \ldots \quad v_k \quad v_{k+1} \quad \ldots \quad v_m$$

Impact on system transactions Impact on system security goals

Optimality of Response Actions

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

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

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

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

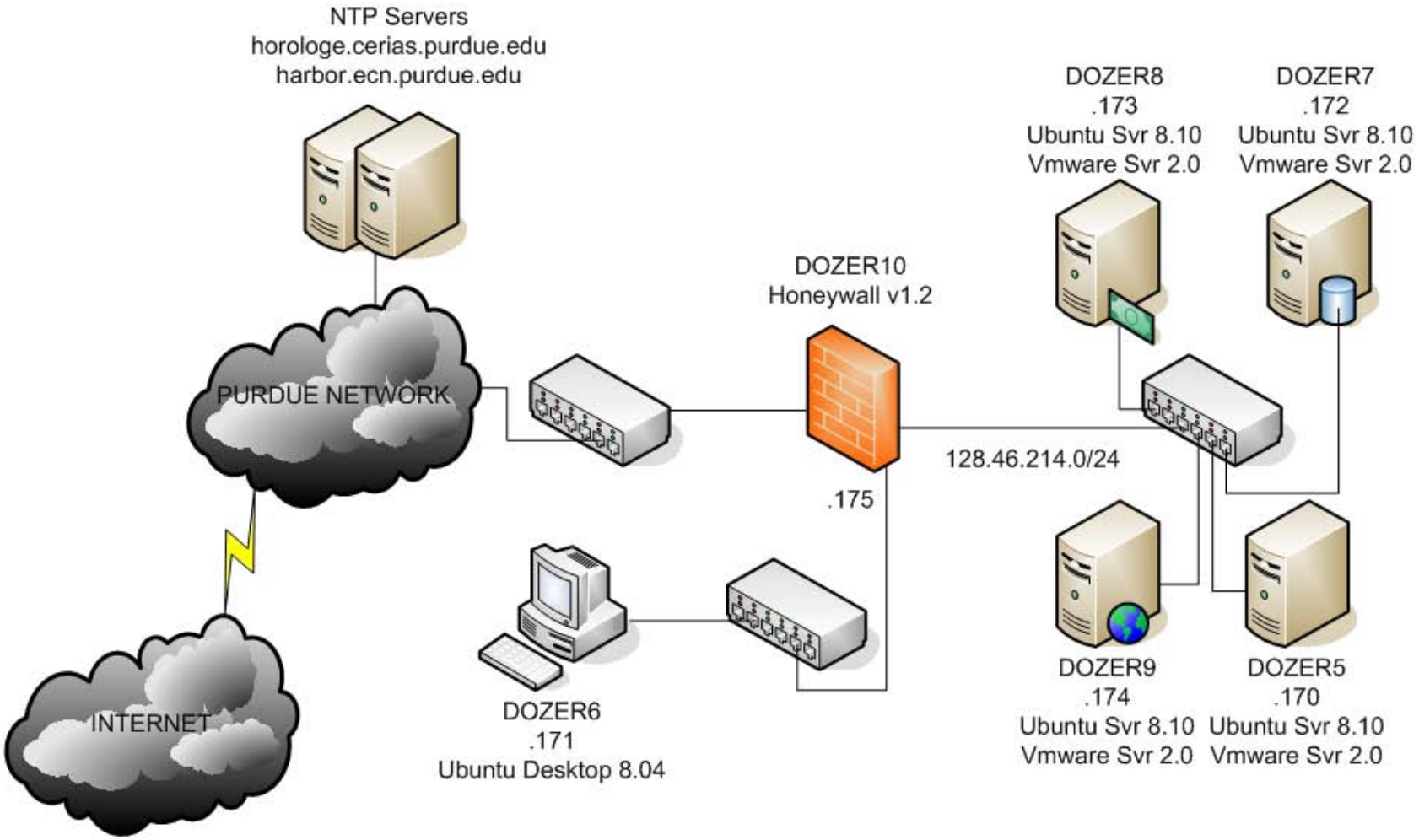
$Iv(r_k)$: Impact from deploying the response r_k

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

ADEPTS: An Unfolding Story

- Exact optimal response determination is NP-Hard
  - Dependencies between responses
- Local optimization of response actions [IJIS 08]
  - Consider the balance between disruptiveness of response and effectiveness of response
- Genetic algorithm for approximate search of globally optimal solution [SRDS 08]
  - Keep good solutions from previously seen attacks
- Current Work: Response combination for zero-day attacks
  - Multiple nodes in the attack graph are not known
  - Multiple edges in the attack graph are not known

How to Validate The Algorithms



PURDUE UNIVERSITY