Role of Secure Configuration and Intrusion Response in Secure Networks

Saurabh Bagchi
Dependable Computing Systems Lab (DCSL) &
The Center for Education and Research in Information Assurance and Security (CERIAS)
School of Electrical and Computer Engineering
Purdue University

Joint work with: Eugene Spafford, Guy Lebanon

Survivable Systems and Intrusion Response

- Modern life heavily depends on computer systems
- Intrusions/security breaches to these systems occur
- 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

Intrusion Response System

- The need for IRS
  - A survivable system needs to provide functionality through intrusions
  - Human intervention after IDS alert can be costly and slow
  - IRS takes reports from IDS (usually bundled together), processes it, and carries out actions to counter the intrusion
- Existing examples of IRS
  - Anti-virus software which disables access to worm executables or files infected with virus
  - Routers/firewalls which actively block worm traffic
- Characteristics of today's commercial IRS
  - Deal with single machines, not distributed applications
  - Static, signature based

Next Generation IRS

- 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
- One system that approaches this: ADEPTS

See our papers in CERIAS TR 2007-94, Computer Networks Journal 07, DSN 05

System Configuration and Security

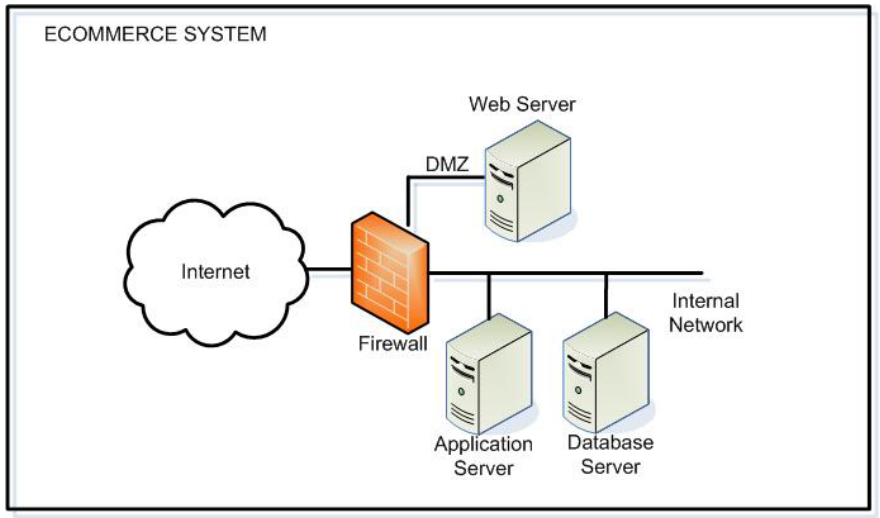
- Secure software but insecure configuration
  - Insecure configurations contribute a significant number of vulnerabilities in today's systems
  - *Example*: WinXP out of the box is insecure and time it takes to download and install SP2 is enough to compromise the machine
- Why do insecure configurations happen
  - Connected components in a system, with a transitive effect of insecure configuration in one component
  - Configuration changes happen regularly in response to new user demands or software upgrades
  - Expert system administrators for all the specialized components are becoming a precious resource

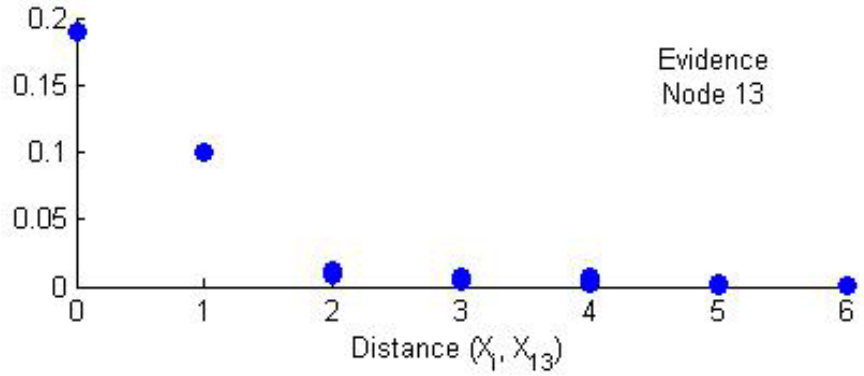
Tools for Secure Configuration Management

- Goals
  - 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
- Required characteristics
  - Incremental execution of the tool as configurations change
  - Risk assessment as a function of the configuration of the system, not just the individual components in the system
  - Takes system owner's input about importance of different system goals
  - Rigorous quantitative basis for the assessment, not just qualitative assessment

One Solution Approach: SMARTS

- Our work on a system called SMARTS:
  - Bayesian network used to model the causality in the network
  - Current security measures taken into account
  - Inference on the Bayesian network determines the conditional probability of certain system goals being violated





Y-axis is proportional to the confidence you will have in whether a system goal is achieved or not (higher is better)
X-axis is distance of the detector from the Bayesian network node