Adaptive Behavior Modeling for Securing Dedicated High Performance Clusters

Presented by Yoginder S. Dandass

NSF U.S.-Japan Workshop on Critical Information Infrastructure Protection
September 28-29, 2004
Organization

- Introduction and Motivation
- Architecture
- Sensors
- Integration of Sensor Data
- Reducing Overhead
- Conclusions & Future Work
Introduction

- Clusters are becoming ubiquitous
  - Corporate research and development
  - Academic research

- Embedded clusters are found in many critical applications
  - Avionics
  - Power plants
  - Manufacturing lines
  - Missile nose cones
Securing High Performance Clusters

- Typically, security in clusters is poorly implemented
  - Recent attacks on academic clusters at Stanford University and other institutions
  - Misuse of assets
- Prejudice against security mechanisms
  - Contributes to inconvenience
  - Impedes performance
- Insufficiently tested software updates introduce vulnerabilities
  - Normal software life-cycle updates
  - Emergency patches
Behavior Modeling for Intelligent Intrusion Detection (IID)

- Criticism of existing IID techniques
  - High overhead
  - High rate of “false positive” indications
  - Easy to evade

- Factors in Cluster environments that mitigate IID concerns
  - Well-known workloads
  - Potential for distributed, multi-level sensors
  - Potential for hierarchical decision engines
Organization

- Introduction and Motivation
- Architecture
- Sensors
- Integration of Sensor Data
- Reducing Overhead
- Conclusions & Future Work
Behavior Modeling and Intrusion Detection on Individual Nodes

Network Fabric for Application and Operating System Communication

Node 1

Network Monitoring

Decision Engine

Behavior Monitoring

Network

OS

CPU

Memory

I/O

Application

Coprocessor (FPGA)

Node n

Network Monitoring

Decision Engine

Behavior Monitoring

Network

OS

CPU

Memory

I/O

Application

Coprocessor (FPGA)

Network Fabric for Monitoring and Modeling activities

September 28-29, 2004
Hierarchical Behavior Modeling and Intrusion Detection
Organization

- Introduction and Motivation
- Architecture
- Sensors
- Integration of Sensor Data
- Reducing Overhead
- Conclusions & Future Work
Application Object Code Analysis

- Finite state automaton (SA)
  - Branches are modeled as $\varepsilon$-transitions
  - $\varepsilon$-transitions allow the (FSA) to accept impossible paths

- Push down automaton
  - Adds a stack to the FSA model in order to prevent acceptance of impossible paths
  - $\varepsilon$-transitions mean that several PDAs need to be maintained simultaneously for program branches
Hybrid PDA (HPDA)

- Static and dynamic analysis of object codes in order to learn address of function calls
- PDA that uses the system call stack rather than its own stack
  - Tracking return addresses in the HPDA eliminates non-determinism in the model
  - Eliminates the need for maintaining duplicate PDAs when branches occur
  - Reduces the likelihood of successful mimicry attacks
Early Results: *gzip* on a 100M File

Convergence rate while constructing the HPDA

Execution time

---

September 28-29, 2004

12
Profiles of System Events

- Function calls for system and library services
- Well-behaved embedded and high performance systems have predictable event sequences and event parameters
  - Sequence of calls
  - Function call parameters
  - Time between events
- Anomalies may indicate faults and intrusions
Neural Networks and Hidden Markov Models for Intrusion Detection
Artificial Neural Network (ANN) for Detecting Intrusions

- Train an ANN to detect anomalous sequence of system calls
- A sliding window is used to extract sub-sequences from a system call trace
- ANN is trained using normal and anomalous data
- Simplified sigmoid function for light-weight computational overhead

\[
y = \frac{x}{2 \left(1 + |x|\right)} + 0.5 \quad \text{and} \quad y' = \frac{1}{2\left(1 + |x|\right)^2}
\]
Modeling of Stochastic System Events

- Weak hidden Markov models (HMMs)
  - Baum-Welch algorithm

- Integration of distributed training
  - Estimating an HMM from multiple observations
    - Requires centralized processing
  - Create an ensemble HMM from individual HMMs
Comparing Accuracy of ANN and HMM-based IID Techniques

- Simulated intrusions and software faults using interposition libraries
  - Modify behavior of MPI calls (e.g., return false errors)
  - Spawn non-MPI processes
  - Copy file content when file is closed
  - Execute extra computations just before the MPI program ends
Monitoring Operating Systems and CPU Data Structures

- Global and local descriptor tables
- Interrupt vector table
- Page directory tables
- Page tables
- Process descriptor blocks
- Thread contexts
Organization

- Introduction and Motivation
- Architecture
- Sensors
- Integration of Sensor Data
- Reducing Overhead
- Conclusions & Future Work
Need for Multiple Sensors

- Sensors vary by data source, type of detection, Implementation, technique used, etc.
- Have their own strength and weaknesses
- No “perfect” or “one for all” sensor
- Suite of different sensors can corroborate/complement/challenge each other’s findings
  - Improve detection rate
  - Provide wider coverage within the system
- Large number of sensors increases security administrator’s workload
Need for Sensor Fusion

- Provide an overall, condensed, and intelligible view of the system
  - Eliminate the need for manual analysis of data
  - Compress and reduce the volume of data
  - Identify event context by associating data from different sensors
  - Improve diagnostic ability by automatically identifying the category, significance, relevance, priority, phase, and result of failure/attack
  - Reduction of false alerts by sensor corroboration

- Sensor fusion techniques must be scalable
- Prototype system uses fuzzy cognitive maps
Early Sensor Fusion Prototype System

Level [I]:

Level [II]:

Level [III]:

Level [IV]:

Top Level GUI Display
Adaptive Sensor Fusion

- Variations of differential Hebbian learning
- Models need to deal with unseen/novel and/or changing situations

![Diagram of sensor fusion with weights and alerts]
Organization

- Introduction and Motivation
- Architecture
- Sensors
- Integration of Sensor Data
- Reducing Overhead
- Conclusions & Future Work
Security Coprocessor Research

- Move behavior modeling and IID computations to an FPGA-based coprocessor on each node
- Determine how the coprocessor can reconfigure itself to optimize performance
- Maximize the transparency and autonomy of the coprocessor
- Reduce the coprocessor “footprint”
Organization

- Introduction and Motivation
- Architecture
- Sensors
- Integration of Sensor Data
- Reducing Overhead
- Conclusions & Future Work
Conclusions and Future Work

- Behavior modeling produces accurate intrusion indications in dedicated environments
- Overheads are relatively small
- NSF - Cyber Trust Program Grant
  - New adaptive sensors
  - Adaptive and hierarchical fusion of sensor data
  - Reconfigurable security coprocessors
Security Research at Mississippi State
The Center for Computer Security Research (CCSR) at Mississippi State

- Department of Computer Science and Engineering
  - NSA certified center of academic excellence

- Colleges of Engineering, Business, and Arts and Science

- Forensics and law enforcement

- Software engineering

- Graphics and visualization

http://www.cse.msstate.edu/~security
Questions?

Selected Bibliography


Learning Formal Models of Programs

```c
{    id = getuid();
2    id += x;
3    return id;
}
int f(int x)
{    if(x) id = getuid();
5        else geteuid();
6        return id;
}

h()
{
7        fd = open("foo", O_RDONLY);
8        if(fd)
9                read(fd, buf, 255, 1);
10        read(fd, buf, 255,1);
11        if(fd) h(0);
12        else f(0);
13        close(fd);
14        f(1);
15        exit(0);
}
Mahalingam Ramkumar

- Security infrastructure for pervasive devices
- Information gathering in sensor networks
TV, Refrigerator
Microwave, Coffee Maker
VCR/DVD Players
Cable / Sat TV boxes
Mobile / Desktop Computers
Mobile Phone, PDA
Printers, Scanners
Wrist watches
Security Cameras
Locks, Garage door openers
Wireless Base-stations
Bluetooth, 802.11 devices

What do all these devices have in common?
1. Need to inter-operate
2. Communicate (securely) with each other

Need a **SECURITY INFRASTRUCTURE** for
**PERVASIVE DEVICES**
Pervasive Computing Research Topics

- Key distribution schemes for ad hoc and sensor networks
- Digital rights management
- Data hiding and steganalysis
TJ Jankun-Kelly  
Visualization for Network and Computer Security

- Augment traditional automated approaches to system log file analysis
  - Human visual system extremely effective at detecting patterns

- Three part process
  - Identify data characteristics of log file
  - Create visual metaphor to depict log file information
  - Explore log file data interactively via visualization
Log file of system use → Interactive Visualization → Knowledge and Insights

Origin AS changes from BGP message logs → Found router misconfigurations

BGP AS path update messages → Discovered lack of AS path damping, router connection instability

Network connections from TCPdump logs → Detected intrusions, attack patterns
How does one specify security requirements in software development specifications?
How does one prove that the specifications are indeed correctly implemented?