Computer Forensics and the Insider Problem

Sean Peisert
UCSD/SDSC
October 19, 2004
Components of *Forensic Analysis*

- Logging
- Auditing
  - Gathering
  - Examining
- Analyzing
Important Definitions

- intent
- insider
- insider problem
Forensics Today

- Computer forensics on UNIX today relies largely on *syslog* and “TCT” to look at files and system state
- Complete lack of structure
- Signal-to-noise ratio for logged events
- Correlation
Assumptions in this Talk

• Assumptions:
  • Assume insiders are threats
  • Intent is irrelevant

• Questions:
  • How to log better? Log everything?
  • How can we re-create or understand easier?
Remaining Talk Overview

Case Studies of Intrusions

Current State of Forensics Research

Synthesizing Ideas From Other Disciplines

Summary and Future Directions
Lessons from Case Studies?

• *Cuckoo's Egg* (Stoll)
• “Evening with Berferd” (Cheswick)
• *Takedown* (Shimomura/Gross)
Case Study: Stakkato

• Initial solution was to shut down

• 1 week of syslogs = 28,634,491 lines (3 GB)

• “Logs show failed attempts, not successful ones”

• Summary: Forensic analysis too hard and too time-consuming. Efficiency is needed.
Forensics: What are the desired results?

- Desired end-result?
- Ideal process?
- Problem: Current solutions are lacking. Why?
Existing Forensic Research

• Bishop: *Goal-Oriented Auditing/Logging*

• Gross: *Active Confrontation*

• King/Chen: *Re-Virt, Backtracking Intrusions*

• Operational Tools: Coroner’s Toolkit, Sebek, SATAN, others

• Stallard: *Automated Analysis for Digital Forensic Evidence*
Bishop: Goal-Oriented Auditing/Logging

- What takes a system from a “good state” to a “bad state”?
- What is a “bad state”?
- *Example*: Van Doorn “nfs_shell” exploit
Gross: *Active Confrontation of Computer Intrusions*

- Most systems record too much or too little.
- Most systems are passive.
- *KAD* Package
- Most systems are “binary.”
- State-change analysis: *States, actions, and transitions*
CoVirt Group

- ReVirt (Dunlap and King)
  - Recreate running of a system exactly
  - Hypervisor approach
- Backtracking Intrusions (King and Chen)
  - Determine the origins of actions
Current Forensic Tools

- General application and kernel syslog data
- TCP Wrapper, Tripwire
- COPS, SATAN, nessus
- Coroner’s Toolkit, Sleuth Kit
- Honeynet’s Sebek
- Solaris SunSHIELD BSM
Stallard: Automated Analysis

- TCT + Expert System = ?
- \[\therefore\text{Same problems as TCT.}\]
Criticisms of current approaches

• U.S.-centric

• Efficiency vs. Effectiveness
  • Existing data ineffective
  • More data inefficient
  • Mutually exclusive?

• Not targeted at “real” systems
Forensics: Ideas for New Directions
Synthesizing Cross-Disciplinary Techniques

• Fault-tolerance
• Debugging
• Transactions
• Standard, Statistical, and Temporal Databases
• Intrusion Detection
Synthesizing Forensics and Fault Tolerance

- Checkpointing and replay
  - LTSS, CTSS, NLTSS
- Message-Logging
  - Bressoud/Schneider: *Hypervisor*
  - Zagorodnov/Marzullo: *FT-TCP*
- Distributed, Heterogeneous Redundancy
Synthesizing Forensics and Debugging

- Reading code
- Spafford: *Software forensics*
- Program verification
- Regression testing
Synthesizing Forensics and Transactions

- “Atomic actions”
- Correlating events
- WISE
Synthesizing Forensics and Databases

- WISE with Database accesses: *Multi-level security* (Baru)

- Standard Databases: Security without uselessness

- Statistical Databases: Defeating *Trackers* (D. Denning)

- Temporal Databases: Recreating Systems (Snodgrass)
Synthesizing Forensics and Intrusion Detection

- Sommer: IDS audit logs in legal proceedings
- Anomaly Detection
- Misuse Detection (and Specification Detection)
Synthesizing Forensics and Intrusion Detection: Anomaly Detection

• D. Denning: *Intrusion-Detection Model*

• Forrest/Hofmeyr/Somayaji: *Intrusion Detection Using Series of System Calls*
  
  • Biological immunology model
  
  • *s-tide, Primary Response, pH*
Synthesizing Forensics and Intrusion Detection: Misuse and Specification

- Kemmerer/Ilgun/Porras: $STAT, USTAT$
- Attack languages
  - Kemmerer: $STATL$
- Templeton/Levitt: $Requires/Provides$ model
Summary and Future

• Improving forensics improves the entire computer security cycle.
• The insider problem can be addressed.
• Future Research Required:
  • Recreation/Replay to improve Logging
  • Multi-Resolution Forensics to improve Auditing
  • Cross-Disciplinary Techniques
What’s Next?

• Do we have to record the entire state of the machine or can security policy limit this?
• Can we use a much lower-maintenance (and non-virtual) system?
• If ReVirt is the “ultimate” logging system, what is the corresponding “ultimate” auditing system?
• How can we take low-level, recorded data and translate it into corresponding high-level events that we can understand?