

Automated Analysis for Computer Forensics

Tye Stallard

University of California, Davis
Department of Computer Science
Computer Security Lab

stallard@cs.ucdavis.edu

www.csif.cs.ucdavis.edu/~stallard

Outline

- Background
- Motivating scenario
- General approach
- Prototype
- Future work

Background

- Definition of “forensics”
 - “The application of science and engineering to the law.”
- Computer crime
 - Instrumentation of crime
 - Target of crime

Forensics organizations

- Law enforcement
- Practicing attorneys
- Academic
 - Computer science
 - Tulsa, CMU, UCF, Santa Clara
 - Accounting
 - Management of Information Systems
 - Law
- Health care

Law enforcement forensics labs

- FBI: CART – by end of 2003
 - 500 terabytes of evidence
 - ~300 cases
- FBI Regional Computer Forensics Lab (RCFL)
- DoD Computer Forensics Lab (DCFL)
- High Tech Crime Task Force (HTCTF)

“First, they do an on-line search”



Digital crime-scene procedure

- “An Examination of Digital Forensic Models” -
Reith, Carr, Gunsch
 - 1) Identification
 - 2) Preparation
 - 3) Approach strategy
 - 4) Preservation
 - 5) Collection
 - 6) Examination
 - 7) Analysis
 - 8) Presentation
 - 9) Returning evidence

Tools

- Disk based
- Hardware
 - Digital Intelligence – F. R. E. D.
 - D. I. B. S. - PERU
- Software
 - **EnCase**, FTK, NetWolves, SafeBack, SnapBack, ILook, ASIS, SMART

Automated analysis techniques

- Thumbnails of images on disk
- Comparison of file extension to file type
- “strings” & “grep” (keyword indexing)
- Cryptographic checksum library
 - Hashkeeper Database/NIST NSRL
- Tools for computer experts
 - TCT/TCTUTILS/TASK

Investigative Goals

- How was the computer penetrated?
- Justify a trap & trace order
- Why did the attacker choose this target?
- Narrow list of suspects
- Document damage

Scenario:

Corporate perspective

- The phone call
- Confirmation
- Course of action:
 - Wipe and reinstall
 - Full investigation

Investigation

- If you do:
 - Local expertise isn't being productive
 - Time intensive
 - System downtime
 - Corporate liability
- If you don't:
 - Lessons will never be learned
 - Will become victim to the same vulnerabilities
 - Experience can't be shared
 - Corporate liability

Questions to answer

- Operational
 - How did they get in?
 - How long have they been in?
 - What are they doing now? (What did they get?)
 - How do we get them out?
- Legal
 - Who are they? (Insider vs outsider)
 - Intentional vs accidental
 - Where's the smoking gun?
 - What's the value of the damage?

Evidence

- Novice attacker
 - Obvious evidence – e.g. syslog, /dev/???
- Experienced attacker
 - Obvious evidence is gone e.g. Kernel module
 - But unintended side effects are available
- Perfect attacker
 - The state of the system *appears* unaltered

General problem

- Given: A policy violation exists
- For each potential cause (hypothesis):
 - What is the supporting evidence?
 - What evidence refutes the hypothesis?
- The transformation of data into knowledge

Debugging and investigation

- Debugging
 - What sequence of events could have occurred for this to happen?
- Automated evidence analysis
 - What evidence is available that would show my hypothesis of the attack is correct?

Work with what's available

- Prepared sysadmins
 - Audit trails/firewall logs/IDS/application logs/accounting records
 - Separate security/administrative domains
 - Independent sources
- Unprepared sysadmins
 - Gather data from what's available
 - Default logging config/file system/file formats

General approach

- 1) Define invariant relationships between redundant digital objects (specification)
 - Expert knowledge
- 2) Automated interpretation of evidence
 - A “standard” format e.g. XML
- 3) Search for violations of data model
 - “Based upon expert knowledge, 'X' shouldn't occur”

An expert's expertise

- Physical world
 - An automobile collision has occurred
 - Where are the potential clues?
 - Does the evidence match the story?
- Virtual world
 - A successful computer attack has occurred
 - Where are the potential clues?
 - Does the evidence match the story?

Invariant relationships

- “A file is only modified when the owner is logged in”
 - Modification time vs owner's login session
- File access time vs owner's login time
 - .login always accessed on login
- A log file is shorter than the backup copy
 - Shimomura and Mitnick
- Compare output of system binaries to that of trusted copies (DERBI)

Gather evidence

- More is better
- Fresher is better
- Different levels of granularity
 - High and low level
- Separate security domains
- Interpret data into objects experts think about

Search for violations of data model

- Based on the knowledge base, an expert system searches all available data for semantic incongruities
 - “The password file was changed, but no one was logged in.”
 - “The security log is shorter than the backup copy.”
- By eliminating the majority of “normal data” and expert can focus on “suspicious data”

Backward chaining expert system

- Pose a query and look for answers
- Backward chaining algorithm
 - Search for facts that support the query
 - Search for implications of the facts
 - Search for implications of the implications...
- Example:
 - “Tye was logged in Friday, from 8:00 to 5:00”
 - “Tye wasn't logged in Friday, 12-8 & 5-12”
 - “There should be no evidence of activity by Tye on Friday between 12-8 or 5-12”

Prototype

- Automated data collection
 - The Coroner's Toolkit
- Automated data aggregation
 - Perl to normalize TCT output into XML
- Automated data analysis
 - JESS to reason with the data
 - Rete algorithm

Prototype

- Example of data

- TCT “body”: b900118e126cfa9a846af902547bffab|/sbin/arp|770|160457| 33261|-
rwxr-xr-x|1|0|0|0|39388|1015273049|986743051| 1008773349 |4096|80
- lastlog: “stallard pts/3 :0 Fri Sep 6 13:54 - 14:52 (00:57)”

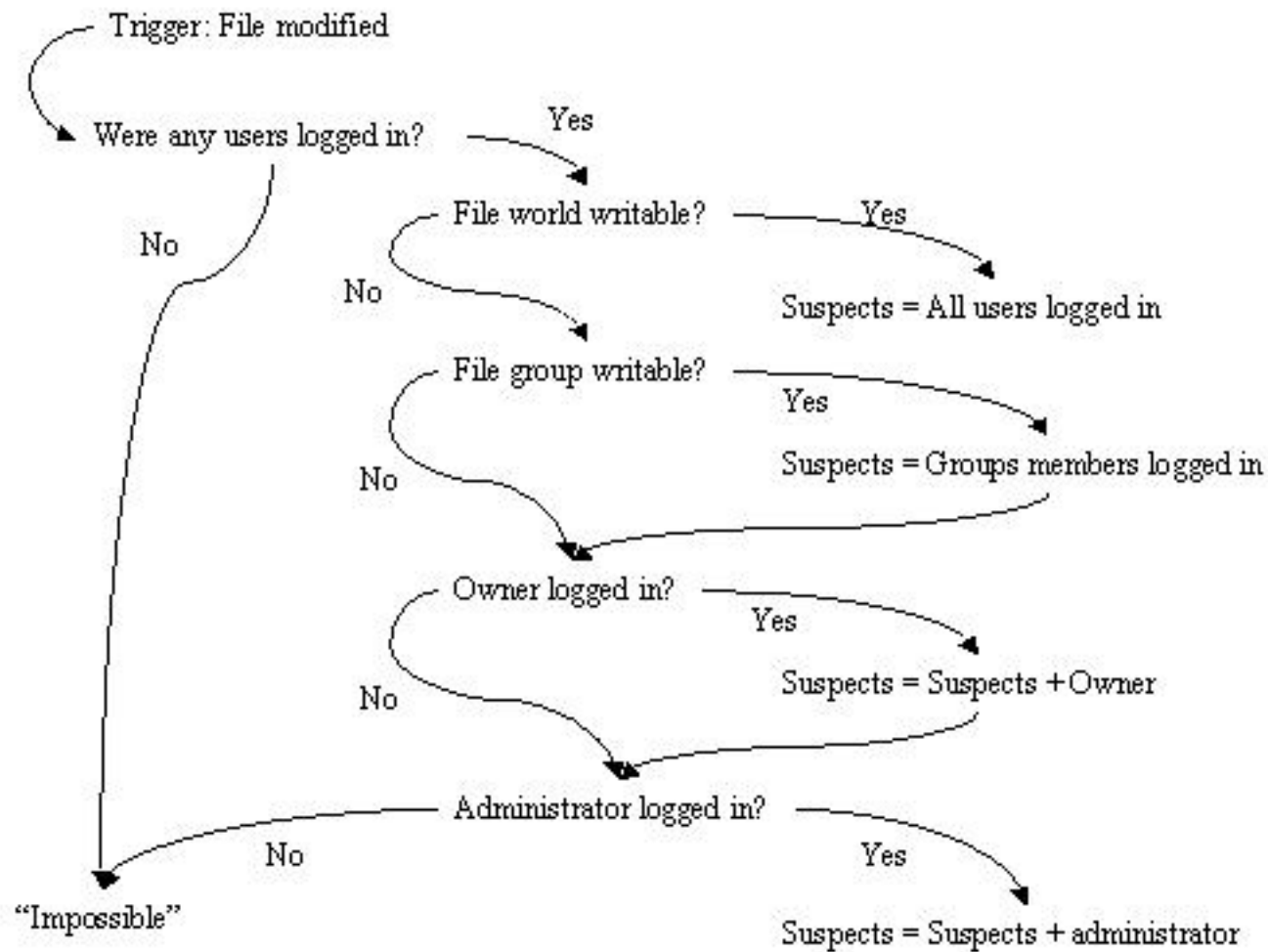
- Example of rules

- Finds all users logged in at the specified time
- “(session (uid ?u) (login ?i&:(>= ?t ?i)) (logout ?o&:(>= ?o ?t)))”

- Example of output

- “File /usr/bin/at owner 0 modified 986418017”
- “No users logged in at time 986418017”

Prototype Decision Tree



Related work

- Artificial intelligence
 - SRI project: “Diagnosis, Explanation and Recovery from Break-Ins” (DERBI)
 - Procedural Reasoning System (PRS)
 - MITRE “Automated diagnosis”
 - Elsaesser & Tanner (plan recognition)
- Audit trail analysis
- Integrity checking
 - Constrained data items in databases
 - Tripwire

Shortcomings

- Garbage in/garbage out (Defiler's toolkit)
- Invariant relationships are a form of specification of known good behavior
 - How does one know there are no exceptions?
 - Incomplete knowledge of a system's components may lead to false deductions (But diagnosis may be justified/verified)
- Implementation would not eliminate the need for experts

Future work

- Expand ontology
 - Add sources of data: Do the “stories” corroborate?
- Response
 - Automated and/or intermediate
- The bridge between evidence and understanding
 - What useful questions are easily answered?
 - What useful questions cannot be answered?

Summary

- An application of an expert system to incident investigation
 - Evidence (in)
 - Explanation (out)
- Prototype implemented
- To automate evidence analysis, a standard description for evidence is important

Automated Analysis for Computer Forensics

Tye Stallard

University of California, Davis
Department of Computer Science
Computer Security Lab

stallard@cs.ucdavis.edu

www.csif.cs.ucdavis.edu/~stallard