

# DERBI: Diagnosis, Explanation and Recovery from Break-Ins



*Douglas B. Moran*

*Mabry Tyson*

*Pauline Berry*

*David Blei*

*Jim Carpenter*

*Ruth Lang*

**Artificial Intelligence Center**

**SRI International**

**333 Ravenswood Avenue**

**Menlo Park CA 94025**

**<http://www.ai.sri.com/~derbi>**



# Project Rationale

- 1 Issue: limited expertise at most sites
- 1 Large commonalties in many intrusions
  - possible variations on shared recipes
  - reusing tools, techniques, tactics
- 1 Goals:
  - Allow non-expert SysAdmin to understand nature, extent, and recovery of break-in.
  - Faster recovery for typical system
  - Improved reporting



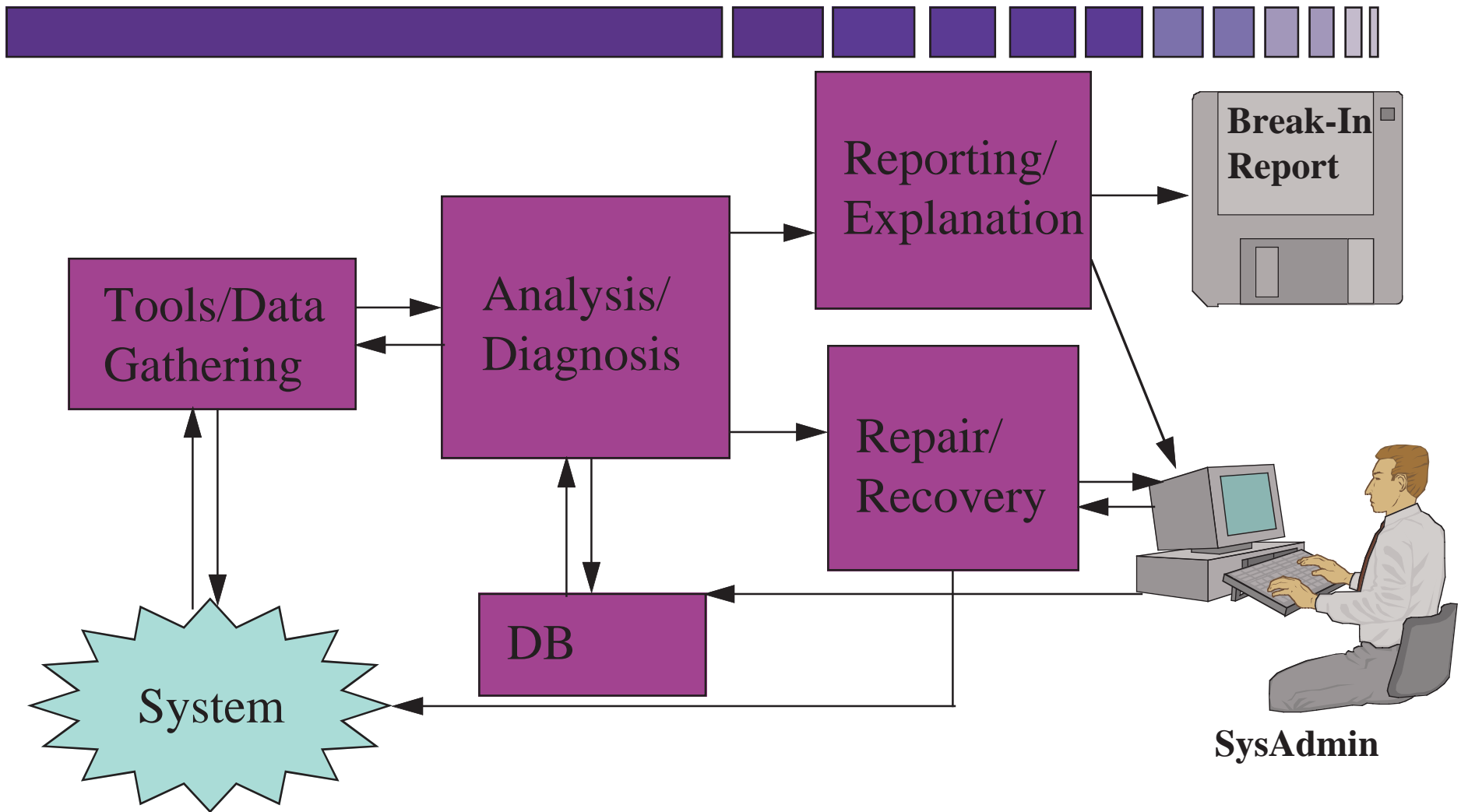
# Overview/Review



- 1 Use AI technology to seek out and analyze data and to direct recovery effort
- 1 DERBI uses its knowledge of:
  - intrusion scenarios (and components)
  - data gathering tools
  - vulnerabilities
- 1 DERBI determines the nature of the attack, explains extent, and advises on recovery



# DERBI Architecture





# Design Goals

- 1 Integrate and respond to information from multiple independent sources
- 1 Easily expandable: new tools, new scenarios, new vulnerabilities
- 1 Minimize impact of on-going use of computer; phased response to suspicious events
  - low-level monitoring of prominent indicators
  - invoked by other system (ID) or SysAdmin



# Knowledge Source: Scenarios

- 1 Scenarios provide components that can be detected or deduced
  - as DERBI finds components of a scenario, it prioritizes looking for remaining components
  - scenario identification: side-effect, not goal
- 1 DERBI recognizes attacks despite new scenarios or new variants
  - tendency to reuse some known tools and tactics



# Example: Rootkit

- 1 Check for known files: /dev/ttyp, /dev/ptyp, ...
- 1 Check for substituted commands
  - checksum (original & any patches)
  - experiment
    - » PS: misbehaves on unusual argument (“ps -/”)
    - » PS: compare to alternatives (e.g., “TOP”)
    - » LS: create known hidden files and test
- 1 Ethernet in promiscuous mode? Possibly legit
- 1 Holes in log files (wtmp)?



# Knowledge Source: Tool Behavior



- 1 Which tools can return which data
  - multiple sources of same information
  - multiple ways to answer same question
- 1 Additional characteristics
  - efficacy
  - cost (time, computing resources)
    - » defer expensive operations (e.g., restore from tape)
  - impact, side-effects
    - » ordering of rules



# Knowledge Source: Vulnerabilities



- 1 Focus on exploited, not existing
  - priority for repair: ~ intruder's repertoire
  - vulnerabilities often only component of exploit
  - SysAdmin may not want to fix:
    - » interfere with needed functionality
    - » expensive to fix (installation, testing, rebooting)
- 1 Determining exploited ones:
  - often little/no direct evidence
  - indirect: signature of known scenario



# Knowledge Source: SysAdmin

- 1 SysAdmin can enter facts into system
  - another source of data
    - » Example: smith was not on computer yesterday
    - » DERBI could query SysAdmin
  - guide/prime the system
    - » Example: ethernet sniffer may be running (deduced from report of passwords discovered on bboard)
- 1 Current interface
  - Developmental prototype: rudimentary UI



# Procedural Reasoning System

- 1 Mature technology
- 1 Integrates goal- and event-based activities (top-down and bottom-up)
- 1 Reactive: responds quickly to new data
- 1 Supports distributed problem-solving through multiple, communicating agents
- 1 Metalevel reasoning for defining complex control strategies



# Status: Diagnosis



1 PRS integration: implemented

1 Tools

- Attack evidence tools

  - » DERBI-developed: implemented + ongoing

  - » 3rd party ID system (emulated): start summer

- 3rd party vulnerability tools: ongoing, June 1998

1 Scenarios

- Rootkit (3 variants): 80% complete

- Other scenarios: Pending access to data

- “leave behinds” & byproducts: July 1998



# Status: Explanation/Reporting



- 1 Level of suspicion of scenarios
  - Gauge with cumulative level: implemented
- 1 Textual output of suspicious conditions
  - Rough, developer level: implemented
- 1 Organized explanation: September 1998
  - Organized by scenarios
  - Explanation of evidence for each component
  - Extraneous suspicious items enumerated
- 1 Report in template suitable for IRST: Dec. 1998



# Status: Recovery/Repair



- 1 Repairs prioritized by suspected exploit
  - Diagnosed scenario guides both explanation and recovery
  - Individual repair steps are cookbook
    - » automated, incident-customized security manual
- 1 Scheduled for FY'99



# Outside Requirements

- 1 Intrusion scenarios, scenario fragments
  - “leave behinds”
  - hiding, camouflage, cleanup
  - side-effects and other signatures
- 1 Info on plans for automated incident reports
  - submission templates
  - querying database
  - dynamic updating of scenarios (and fragments)