Motivation for File Integrity

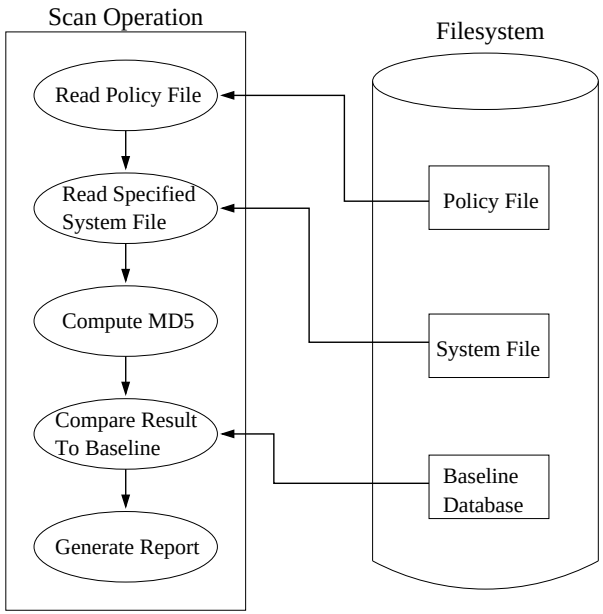
Intrusion Detection System (IDS) goals include the ability to:

- characterize and identify computer system attacks,
- identify occurrences of security breaches, and
- trigger actions to protect computing resources.

Compelling motives for distributed file integrity monitoring using mobile agents include:

- it is a popular detection approach,
- detection of unforeseen intrusions,
- and intruders often open a backdoor to facilitate future access.

File Integrity Operation



- A *policy* file defines the monitored resources
- *Baseline* data is computed for entries in the policy file
- Computation uses *hash function* such as SHA-1 or MD5
- Scan results are compared to the baseline reference values
- A discrepancy results in alarm notification

Motivation for Mobile Agents

On-demand deployment of mobile agents can address several challenges present in client/server environments:

- mitigate network latency,
- reduce network load,
- execute asynchronously and autonomously,
- dynamically adapt to changes in network resources,
- are naturally heterogeneous, and
- provide robust and fault-tolerant behavior.

Goals

Two Projects:

- *Mobile File Integrity and Consistency Analyzer – M-FICA.*
- *Collaborative Object Notification Framework for Insider Defense using Autonomous Network Transactions – CONFIDANT*

Two Goals:

Goal-1: Reduce single point-of-failure exposures in existing IDS frameworks and

Goal-2: Increase barriers against insider tampering pathways.

Distributed processing approach using mobile agents to reach these goals.

Commercial Tools

Tripwire:

- is a popular integrity-checking tool
- that uses a baseline database of signatures
- with Manager acting as a central console and reporter.

Disadvantages include:

- extensive network usage sending data to a central processing site and
- a single point-of-failure also due to centralized processing.

Comparison of related frameworks

File Integrity Analyzer / IDS	Execution Model	Agent Form	Agent-to-Agent Interaction	Single Point of Failure	Safeguards Against Insider Tampering
Tripwire, AIDE, Veracity, integrit	Client-Server	N/A	N/A	Yes	No
AAFID	Deployed Agent	Key-Value Pair	None	Yes	No
Bernardes-Moreira	Mobile Agent	Aglet	Create, Halt	Yes	No
TACH	Deployed Agent	Aglet	None	Yes	No
M-FICA	Mobile Agent	Concordia Java Object	Communicate	Yes	No
CONFIDANT	Mobile Agent	Concordia Java Object	Communicate, Create	No	Yes

M-FICA Operation

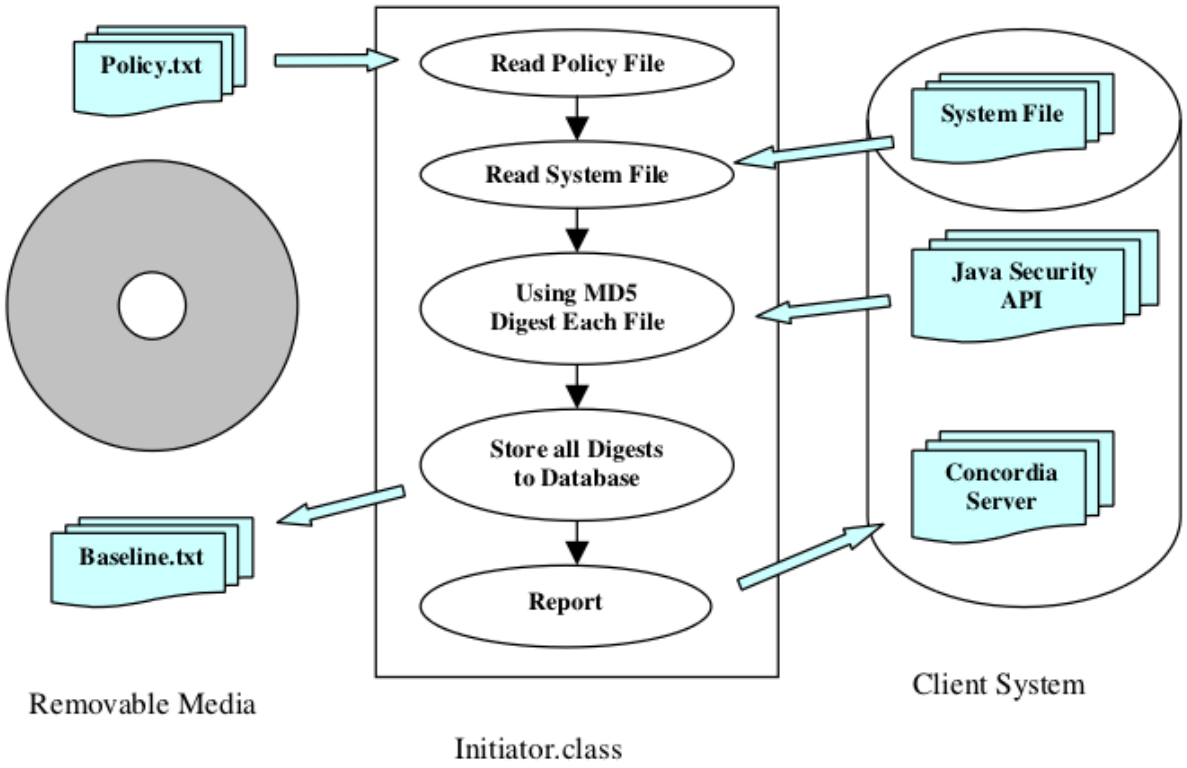
M-FICA agents travel between network hosts to inspect for file changes using local resources then return to report results.

Functions: Create a baseline database then compare results on subsequent scans.

Environment: Implemented in Java using the Concordia mobile agent API.

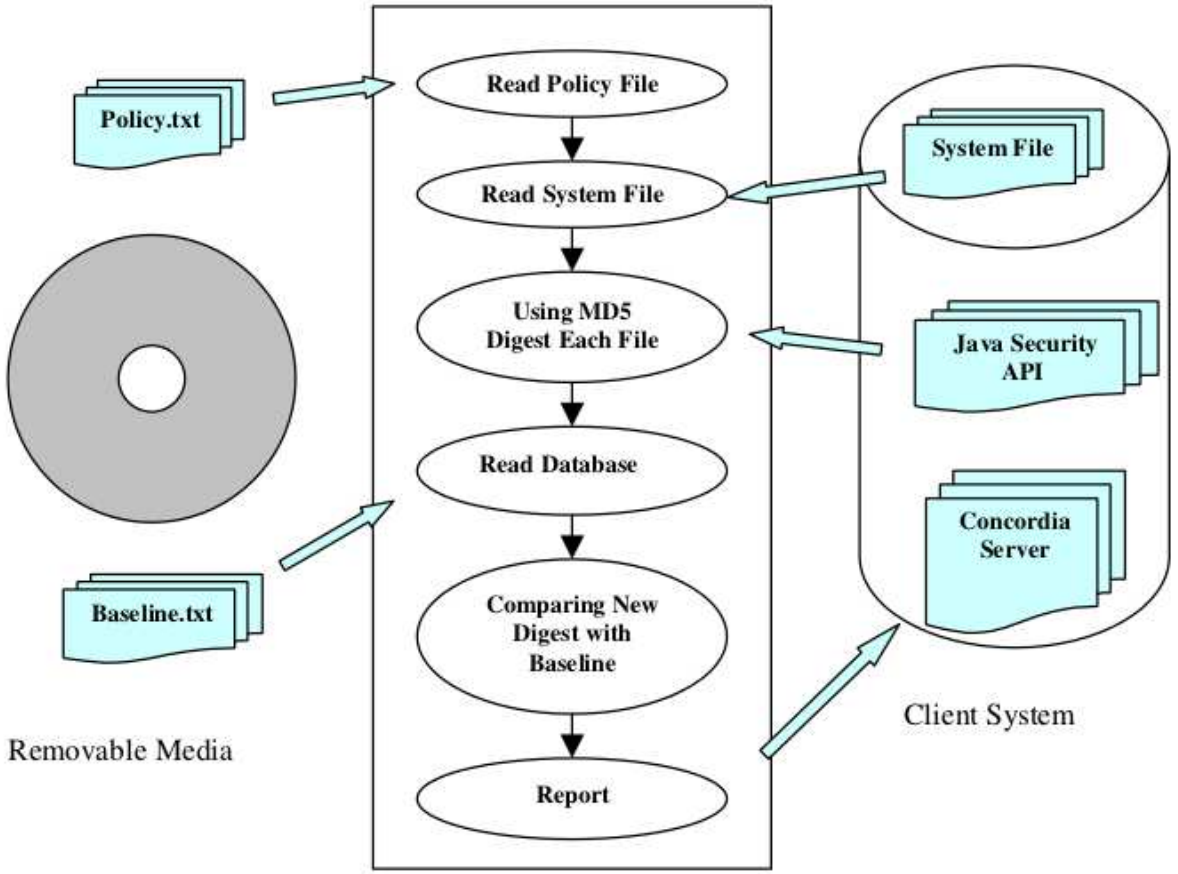
M-FICA Initiator Agent

Initiator agent reads the policy file and creates a baseline for each entry.



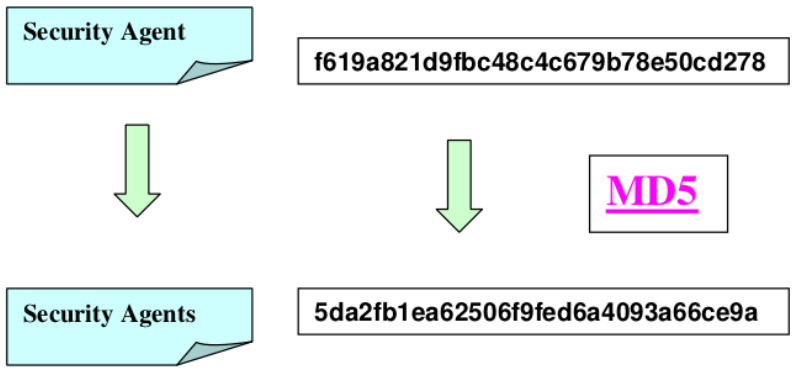
M-FICA Examiner Agent

Examiner agent reads the policy and system files, compares the current MD5 with the baseline database, and reports the result.



M-FICA Experimental Results

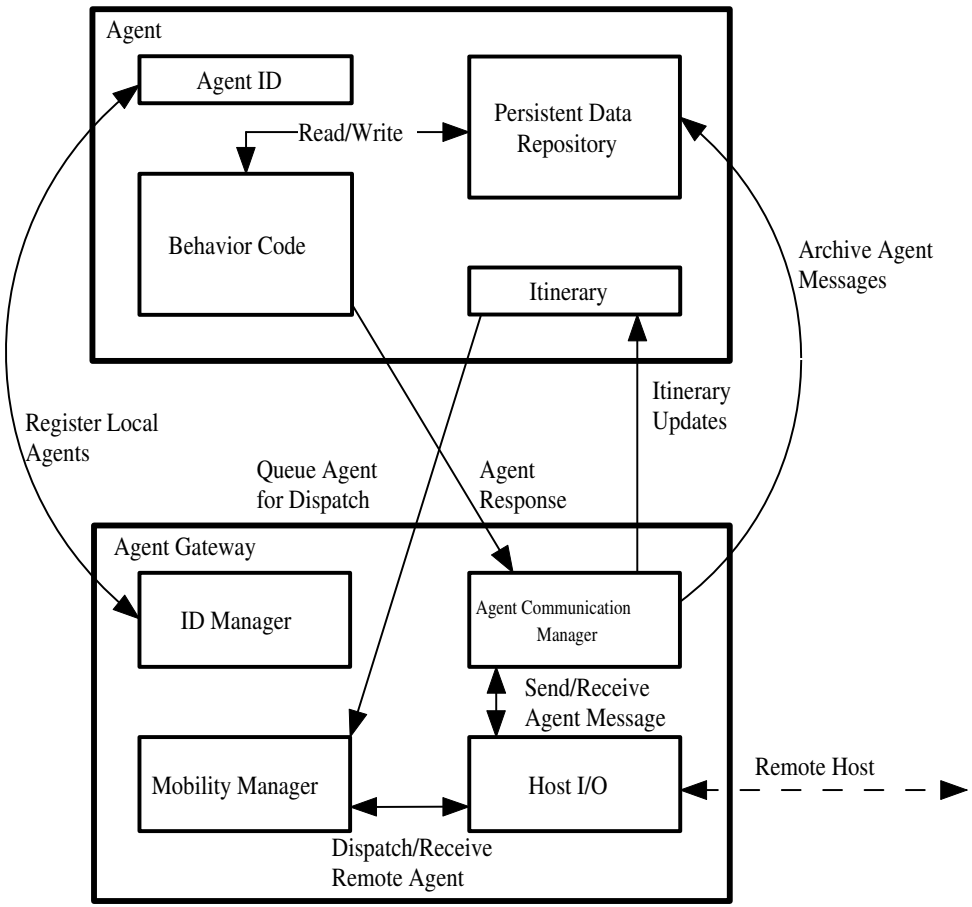
- A test file is created to evaluate agent operation.
- Results show the Initiator/Examiner agent protocol is able to identify changes in file contents.



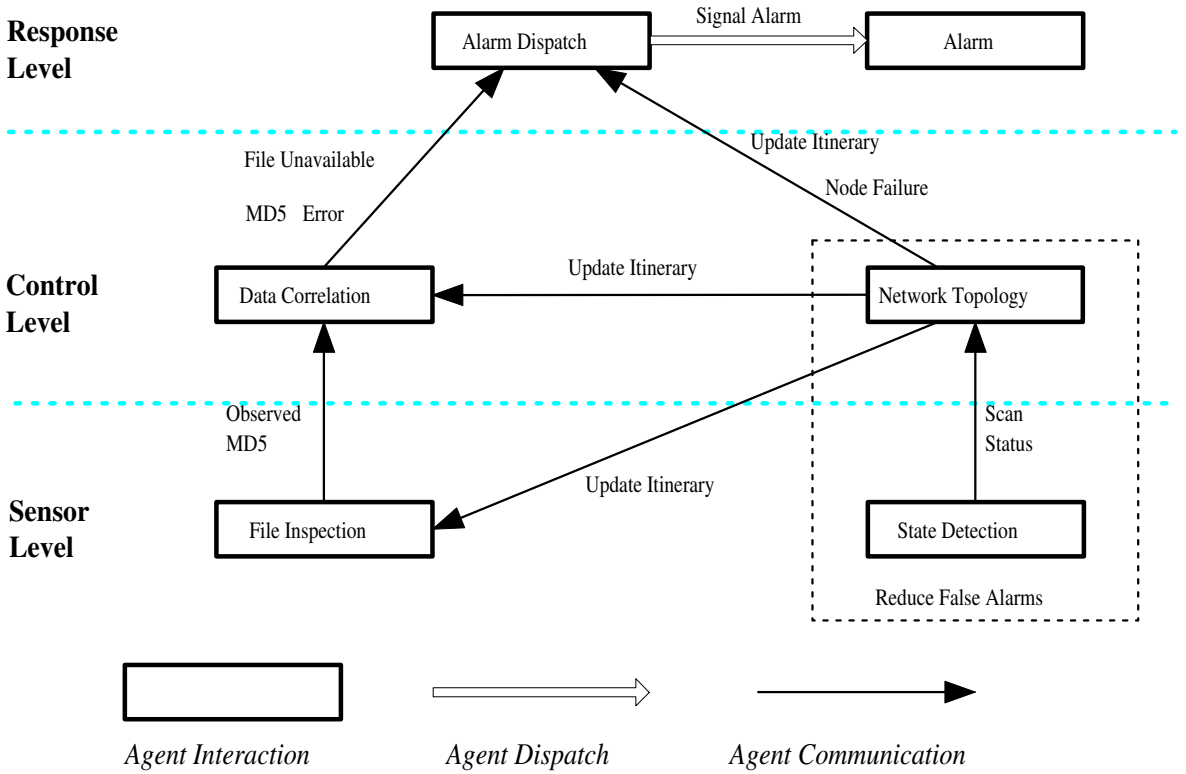
Tampering Modes

IDS Vulnerability		Tampering Mode	Instantiation for File Integrity Analysis
Spoofing	Sensor	Spoonfeeding	Alternate data stream is conveyed during file scan
	Control	Sugarcoating	Unfavorable cryptographic digest is modified to appear as the desired result
	Alarm	Recanting	Fraudulent command is issued to deactivate alert
Termination	Sensor	Blindfolding	Detection mechanism is disabled
	Control	Commandeering	Decision-making process is usurped
	Alarm	Soundproofing	Notification mechanism is eliminated or muted
Sidetracking	Sensor	Blockading	Resource usage is forestalled to starve access
	Control	Pacing	Scan timing reference is corrupted or execution priority is reduced
	Alarm	Scapegoating	Attention is diverted to a contrived distraction
Alter internal data	Sensor	Retroactive Baselining	Reference values for digests are modified
	Control	Descoping	Exemption is added to policy file to exclude scan coverage
	Alarm	Value Jamming	Stand-alone process continuously writes `FALSE` into the memory location
Selective deception		File Juggling	Target files interchanged before and after scanning

CONFIDANT Agent Structure

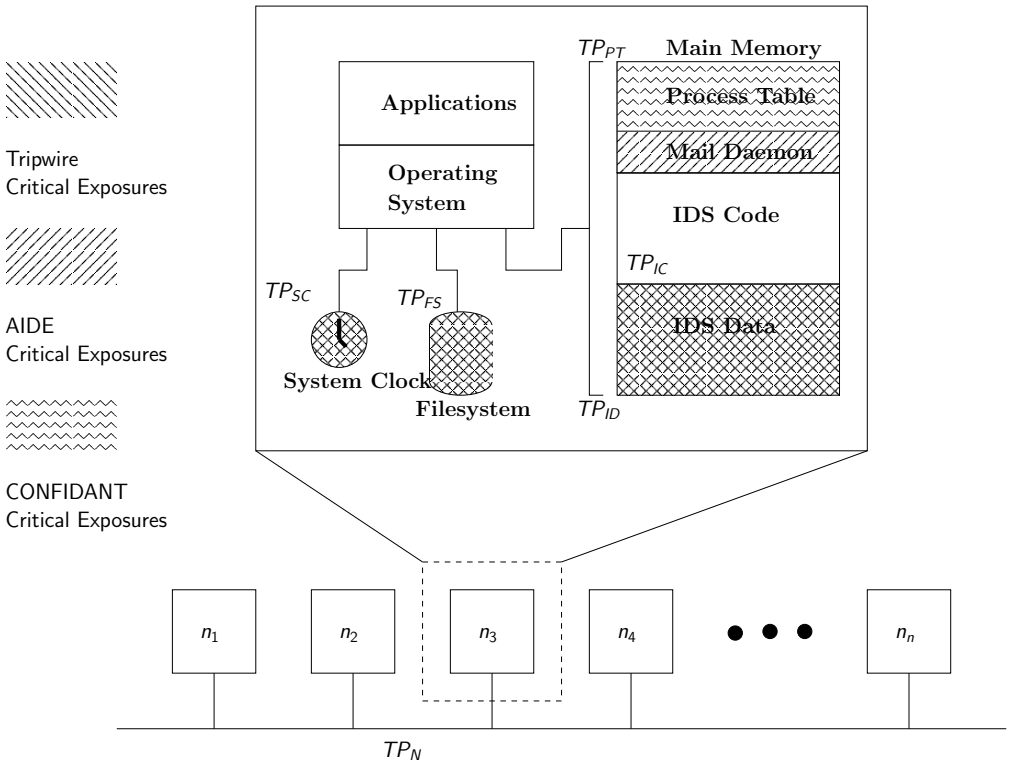


CONFIDANT Echelons



Current Status

- Successfully handled single point-of-failure exposures.
- Reduced exposures to insider tampering.
  - Possible exposures remain at the hardware level.



Mobility-Enhanced File Integrity Analyzer For Networked Environments

Guantong Wang, Ronald F. DeMara, Adam J. Rocke
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2450

July 12, 2005