Metrics for Detecting Compromised Systems: In Distributed Enterprise

Shivaraj Tenginakai

CONTEXT – SECURITY WITHOUT CONTROL

Security for Enterprise Consisting of

- Unsecured Systems
- Unsecured Network
- Unknown System Ownership

ENTERPRISE – THE “FORT”

Application Architecture

- Client Server

Infrastructure

- Data Center - the “Fort”

Security Goal

- Defend the Fort

ENTERPRISE– FORTS ARE EXPENSIVE

Capital Costs

- Real Estate
- Network
- Construction

Operational Costs

- Energy Cost
- Bandwidth Cost
- Real Estate Cost

Security

- Strength - Control
- Weakness – Known Location

ENTERPRISE – THE "COLLECTIVE"

Application Architecture

- Peer-to-Peer

Infrastructure

- Internet linked Personal Computers- the "Members"

Security Goal

- Maintain "Member" integrity

ENTERPRISE – "COLLECTIVES" LACK CONTROL

Capital Cost

- None

Operational Cost

- None

Security

- Strength – no easy target
- Weakness – lack of control

COLLECTIVE – SECURITY GOAL

Maintain

- Operational Integrity
- Data Integrity

COLLECTIVE – INTEGRITY WITHOUT CONTROL

Unreliable

- Data
- Log files
- Operating System
- System Configuration
- Time

COLLECTIVE – ACHIEVABLE SECURITY

Not to “Defend” Members

But..

“Detect and Drop” Compromised Members

COLLECTIVE – SECURITY STRATEGY

If you cannot trust “what” a member says

then..

You need to rely on “how” they say it

MEMBERS – COMPROMISES

TYPES

- Structural
- Temporal
- Data

MEMBERS – POTENTIAL THREATS

TYPES

- Byzantine
- Non-Byzantine

MEMBERS – BYZANTINE COMPROMISES

CHALLENGES

- Defend against unknown
- Always on Alert
- Risk of immunization

MEMBERS – DETECTING COMPROMISES

BYZANTINE APPROACH

- Lamport
- Vogels
- Haeberlen

Collective

- Not Practical

MEMBERS – DETECTING COMPROMISES

NON-BYZANTINE APPROACH

- Know Correct Behavior
- Verify Expected vs. Actual Behavior
- Subject High Value Targets to Greater Audit

OPERATIONS – DEFINING CORRECTNESS

Using Operation Data

- Privacy
- Encryption
- Retention
- Compliance

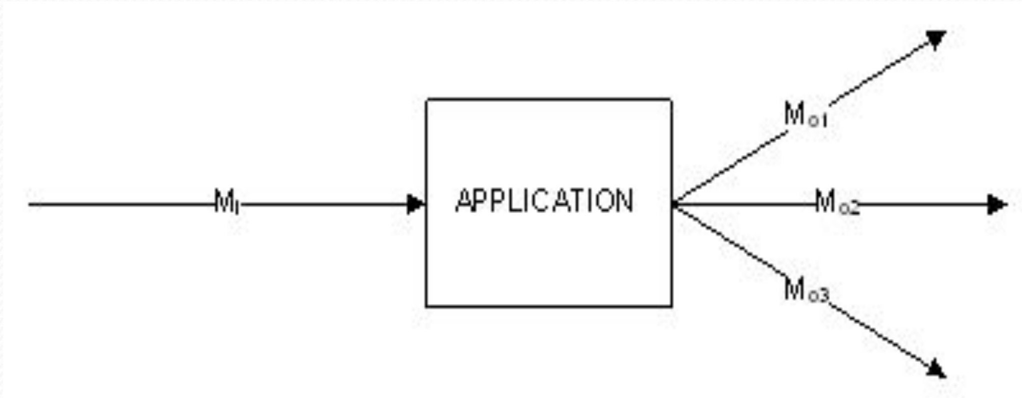
OPERATIONS – DEFINING CORRECTNESS

Communication

- Message Pattern
- Message Order

METRICS– MESSAGE PATTERN

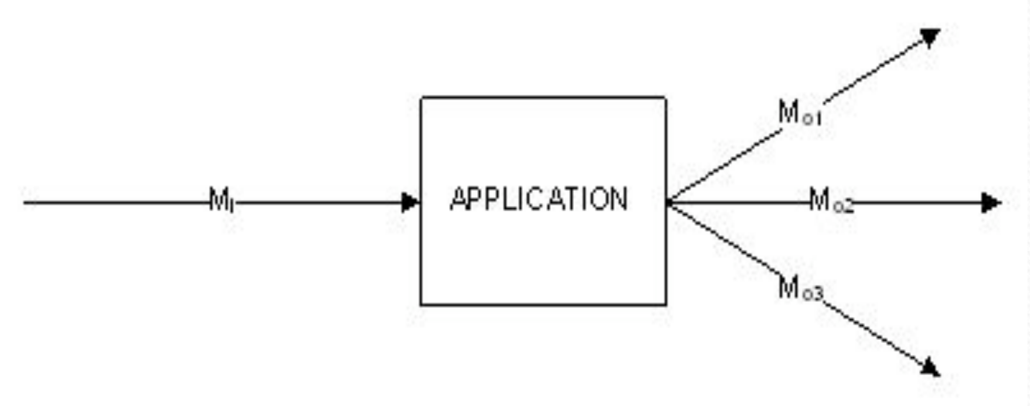
Message Counter (MC)



$$MC = f(MC_c, M_i, M_{o1}, M_{o2}, M_{o3}, \ldots)$$

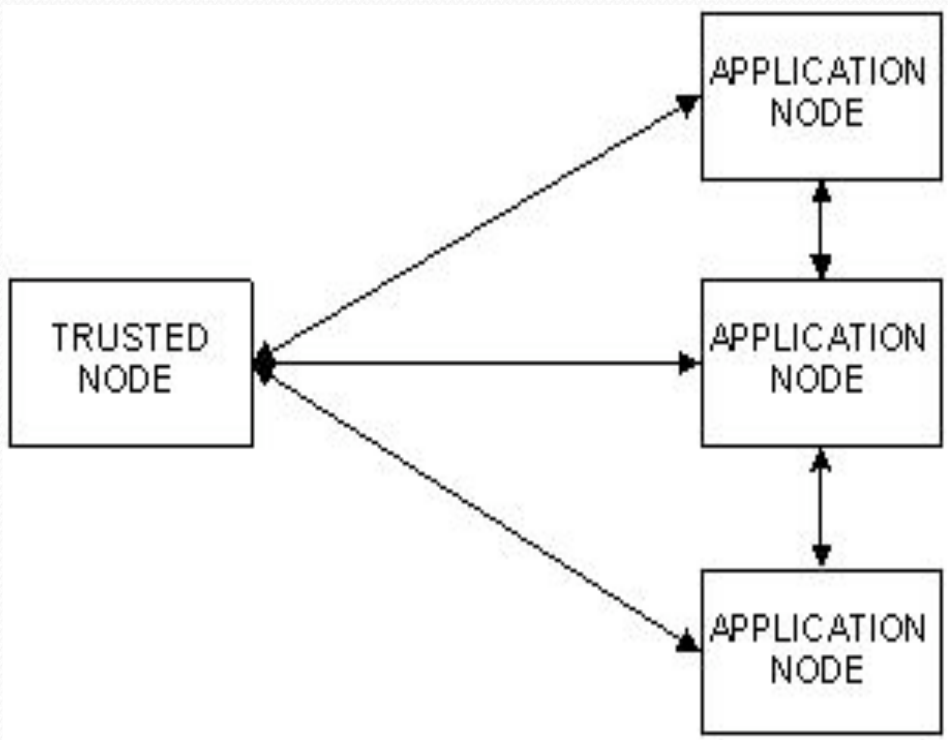
METRICS– MESSAGE ORDER

Lamport Counter (LC)



$$LC = max\{ LC_c, Lamport\ Clock\ (M_i, M_{o1}, M_{o2}, M_{o3},\ldots\_\} + 1$$

MEASUREMENT– ARCHITECTURE



MEASUREMENT– DATA COLLECTION

Issues

- Noise
- Storage Cost
- Computation Cost

MEASUREMENT– NOISE

Solutions

- Time Series
- Prediction Bands

MEASUREMENT– METRIC STORAGE SIZE

per Metric per Member

- 4 integer points (16 bytes) every 5 minutes
- 16 * 288 = 4608 ~ 5KB per day
- 5 * 365 = 1825 ~ 2MB per year

MEASUREMENT– METRIC STORAGE COST

1 million members

- 1,000,000 * 2MB = 2TB
- $0.15 * 2000 = $300/ month
- $300 * 12 = $3600/per year

MEASUREMENT– COST PER MEMBER

1 million members, 2 Metrics

- $ 7,200 * 2 ~ $15000/ per year
- $15000/1,000,000 = $.015/ per year

MEASUREMENT– TRUSTED NODE COST

Trusted Node Ratio	= 1:10,000
Bandwidth Requirement	= (2 * 16 * 10,000)/(5 * 60)
	~ 2KB/s
Number of Computations	= (2 * 4 * 10,000)/(5 * 60)
	~270/s
Computations/Core	~ 70/s

MEASUREMENT– TOTAL COST

For 1 million members

Number of Trusted Nodes = 100 members
Trusted Node Instance = 1 Amazon EC2 Large
Amazon EC2 Large Instance = $1300/ per year (8 GB, 4 Core, 64-bit, 850 GB store)

Cost

- $ 1300 * 100 ~ $130,000/ per year
- $130,000/1,000,000 = $.13/ per year
- $.13 + $.015 = $.14/per year

IMPACT – COST

Cost

- Cheaper Than Datacenter
- Cheaper Than Cloud

IMPACT – ENVIRONMENT

- Lower Space
- Lower Carbon Footprint

IMPACT – RESOURCE REQUIREMENTS

- Not Computationally Intensive
- No Storage Required
- Can be computed by devices with limited capabilities

IMPACT – VS. LOG FILES

Cost

- Less Than Cost of 1 Software Engineer
- Economically Negligible

IMPACT – ARCHITECTURE

Security Metrics Enable Architecture and not just detect runtime issues.

Why do cars have brakes?' Everyone says, 'So they can stop.' But the real reason cars have brakes is so they can go fast"

\- **Sara Gates**

VP Sun Microsystems

IMPLEMENTATION– DEFINING CORRECTNESS

Static

- Extending WSDL

Runtime

- Member Set

IMPLEMENTATION– INTRODUCING RANDOMNESS

- Dummy Members
- Dummy Operations
- Dummy Messages
- Varying Operation Names
- Vary Operation Implementation

AUDIT– STRATEGY

- Random Tracer Bullet
- High Value Operations

AUDIT – GOAL

- Data Integrity
- Metric Value Correctness

EFFECTIVENESS - SCALING

- Jitter
- Scaling Model Validated
- Cost Model Validated

EFFECTIVENESS - CORRECTNESS

Complete Under Simulation

- Able to absorb “tweaks”
- Not yet validated in real world

EFFECTIVENESS – UNRESOLVED ISSUES

- Enabling Compromised Members
- Dealing with compromised member set

QUESTIONS?

Shivaraj Tenginakai
shivaraj@sarithi.com