Sana Security™

Immunological Approaches to Computer Security

JET Workshop
April 14, 2004

Steven Hofmeyr
Chief Scientist, Sana Security

Copyright Sana Security, Inc. 2003

Securing Complex Systems

Sana Security™

- Knowledge-based approaches are limited
  - Signature-based/Rules-based
  - Dependant on human expertise
  - Subject to human bias and error
  - Cannot scale or keep up with complexity
  - Similar to limitations in Expert Systems in AI
- Need a bottom-up solution:
  - *Autodidactic* (self-learning)
- How do you develop such a solution?
  - Biological inspiration
  - Immune system as a model for security

Copyright Sana Security, Inc. 2003

The Immune System (IMS)

Sana Security

- Teleological viewpoint
  - The IMS is “designed” to protect the body
- Exceptionally difficult security problem
  - One human body is vastly more complex than entire IT infrastructure
- Continuously under attack by new and evolving threats
  - Evolution operates over millennia & bacteria replicate within days
  - No one has ever built a biowarfare agent from scratch
- IMS is highly effective
  - most of us are healthy most of the time
  - No human intervention or control needed
- If the IMS was at the same technological stage as computer security systems, we’d be extinct

Copyright Sana Security, Inc. 2003

Self Versus Nonself

- The IMS learns to discriminate between self and nonself
- Discrimination in the IMS based on *peptides*
  - Sequences of amino acids
  - IMS learns to recognize self peptides & attacks harmful nonself
  - Learning is ongoing
- Need the equivalent of a peptide for a computer system
  - *Sequences of system calls* made by running programs
  - System call sequences indicate paths through program code
  - Suited to programs with repetitive behavior (e.g. servers)
  - Exploits of vulnerabilities follow unusual code paths
  - Early research at UNM – later commercialized at Sana Security.

Copyright Sana Security, Inc. 2003

Adaptation in the Immune System

Sana Security™

- Innate IMS
  - Evolved defenses to common pathogens, e.g. bacterial coat
  - Fixed during the organism’s lifetime
- Primary Response (Adaptive IMS)
  - Learn self and detect deviations
  - Protects against new pathogens
- Secondary Response
  - Refine recognition of nonself through Darwinian evolutionary process
  - Remember for future responses
  - Better detection & elimination of known pathogens
- Adaptation confers dynamic protection
  - Cells are continually dying and being born
  - Allows adaptation to changes in self

Copyright Sana Security, Inc. 2003

Adaptation in Computers

Sana Security

- Innate = heuristics
  - e.g. buffer overflow detection (common attacks)
- Primary response = anomaly detection
  - Learn profile of normal sequences of system calls
  - Monitor for deviations from profile
  - Detect 0-day
- Secondary response = signature detection
  - Automated signature extraction
  - Drop packets at network level
  - Pattern matching to identify attack variants
- Dynamic protection
  - Forget unused system call sequences
  - Incremental learning during system changes (manual/automatic)

Copyright Sana Security, Inc. 2003

Responding to Attackers

Sana Security

- Detection inextricably linked with response in IMS
  - Detection is binding (action)
- But computer defense usually separates detection from response
  - Gather data from distributed sensors
  - Analyze/correlate centrally
  - Human-mediated response
- Need automated response:
  - IPS = “Intrusion prevention systems” (marketing jargon)
  - Localized for speed: stop attacks before they do harm
  - Protect unpatched applications (against both known and 0-day)
  - Essential to stop fast-spreading worms

Copyright Sana Security, Inc. 2003

The False Positive Problem

- False positives + response == block legitimate behavior
- Blocking legitimate behavior can be expensive
  - Million dollar transactions
  - Downtime on mission critical servers
- Failure modes
  - Catastrophic (one disastrous event)
  - Repetitive (ongoing loss of legitimate functionality)
- We can never get away from false positives
  - Scale of systems
  - Dynamic environments
  - Base-rate fallacy

Copyright Sana Security, Inc. 2003

Base-Rate Fallacy (BRF)

Sana Security

- Low base-rate of incidence => most alarms will be false
- Example: test for disease
  - Test accuracy = 99% symmetrical
  - Base-rate incidence = 1/10000
  - Probability of disease = 0.0098 (approx 1%)
- Applied to intrusion detection
  - Human operators will not trust alarms
  - Response to alarms will mostly be harmful
- Also a problem for the IMS

Copyright Sana Security, Inc. 2003

BRF and the Immune System

Sana Security

- Chemical binding is not perfect
  - Errors in “detection”
- Compute base-rate of incidence
  - Self: 1ml blood contains 5 x 10^4 cells
  - Nonself: HIV treatment threshold = 55,000 copies per ml
  - 1 x 10^{-5} base-rate
- Assume accuracy of
  - P(true positive) = 0.9
  - P(false positive) = 0.001
- Result: 99.9% of all bindings are to self
- Why is this not a problem in the IMS?

Copyright Sana Security, Inc. 2003

Immunological Costimulation

Sana Security

- IMS overcomes false positive problem via *costimulation*
- Two signals required for immune cell activation
  - First signal: anomalous peptide
  - Second signal: indicator of damage (cell death)
- Signals of cell death
  - Explicit: yell of “murder!”, e.g. heat shock proteins
  - Implicit: unusual exposed cell contents (non-apoptosis)
- Hence response occurs only in the presence of damage
- And response is proportional to damage
- Self-recognition not associated with damage

Copyright Sana Security, Inc. 2003

Costimulation in Computers

Sana Security

- Two signals
  - First signal == anomaly detection
  - Second signal == damage indication
- Signals must be reliable
  - Explicit, e.g. local system load recorded through remote secure logging
  - Implicit, e.g. server response time
- Three areas of damage
  - Availability
  - Integrity
  - Confidentiality

Copyright Sana Security, Inc. 2003

Damage in Computers

Sana Security

- Damage to *availability*
  - Easiest to measure
  - Explicit signals, e.g. local resource loads, memory usage, etc.
  - Implicit signals, e.g. server response times, network congestion, etc.
- Damage to *integrity*
  - Disk and memory content modifications
  - Example: monitoring for file system changes (integrity checkers)
  - Example: monitoring for code-injection into memory
- Damage to *confidentiality*
  - Hardest to measure
  - Example: monitoring for reads of confidential file information
  - Example: monitoring for network transmissions of confidential information

Copyright Sana Security, Inc. 2003

Limitations of Damage Monitoring

Sana Security

- Damage dependence works in the IMS because:
  - Cells are cheap (can afford to lose some)
  - Damage is incremental, i.e. no catastrophic failures
- Sometimes components are “cheap” in a computer system:
  - Server farm
  - Desktops
  - Ordinary web-server transactions
- But often fails in current computer systems
  - Components not discardable, e.g. critical databases
  - Events not discardable, e.g. million-dollar transactions
  - Catastrophic failures, e.g. widespread vulnerability
  - Loss of confidentiality, e.g. access to credit card database

Copyright Sana Security, Inc. 2003

Damage-Response Architectures

Sana Security

- Only react when there is damage
  - No reaction to false positives
  - Initial attacks will be successful
- Ensure successful attacks do not lead to catastrophic failure
  - Components must be cheap and redundant
  - Events must be cheap and repeatable
  - Failures in confidentiality must be limited, e.g. fragmentation scattering
  - Use diversity to prevent failure replication
- Ensure damage signals are reliable
  - Prevent spoofing
  - Prevent blocking, e.g. remote secure logging

Copyright Sana Security, Inc. 2003

Distributed Damage-Response

Sana Security™

- Example: network of desktops
- Assume compromise of any individual machine is tolerable
- First signal
  - Each machine runs an anomaly detection system
  - Each machine communicates anomalies to its neighbors
- Second signal
  - Each machine communicates its internal state to its neighbors
  - Each machine monitors its neighbors for damage
- Example: stopping a worm
  - Anomalies dispatched to neighbors as compromise occurs
  - Machines monitor neighbors for damage, e.g. port scanning, network overload, crashing, poor response times, etc.
  - Machines react by increasing their security posture (prevention mode)

Copyright Sana Security, Inc. 2003

Correlation and False Positives

Sana Security

- Another perspective
  - Multiple signals == correlation of multiple anomaly sources
- Reduce false positives with little impact on false negatives
  - Maximize sensitivity of each detection system
  - Correlate to reduce false positives
- Example
  - Two independent detection systems with FP = 0.1 and FN = 0.2
  - Assume a decrease of FN to 0.1 results in increase in FP to 0.2
  - Then correlated FP = 0.2^2 = 0.04 and FN = 1 – 0.9^2 = 0.19
  - No change in FN, 5 times reduction in FP
- Sources must be independent but not disjoint
- If FN = 0 then correlation simply reduces FP

Copyright Sana Security, Inc. 2003

False-Positive Tolerant Architectures

Sana Security

- Minimize false negatives at the cost of false-positive increase
- Requires tolerance of false positives
    - Remove humans from the loop (automated response)
    - Ensure no catastrophic failures from automated response
    - Ensure no repetitive failures (hard – diversity?)
- Potentially easier to design a system tolerant of false positives
    - Known quantity (predictability?)
    - Failures don’t target the weakest points
- Can we transform problem from security to fault tolerance?

Copyright Sana Security, Inc. 2003

Example False-Positive Tolerance

- Example: scale-free networks
  - Tolerant of random node failures
  - Not tolerant of attacks targeting hub nodes
  - Prevent targeted attacks, even at cost of more random failures

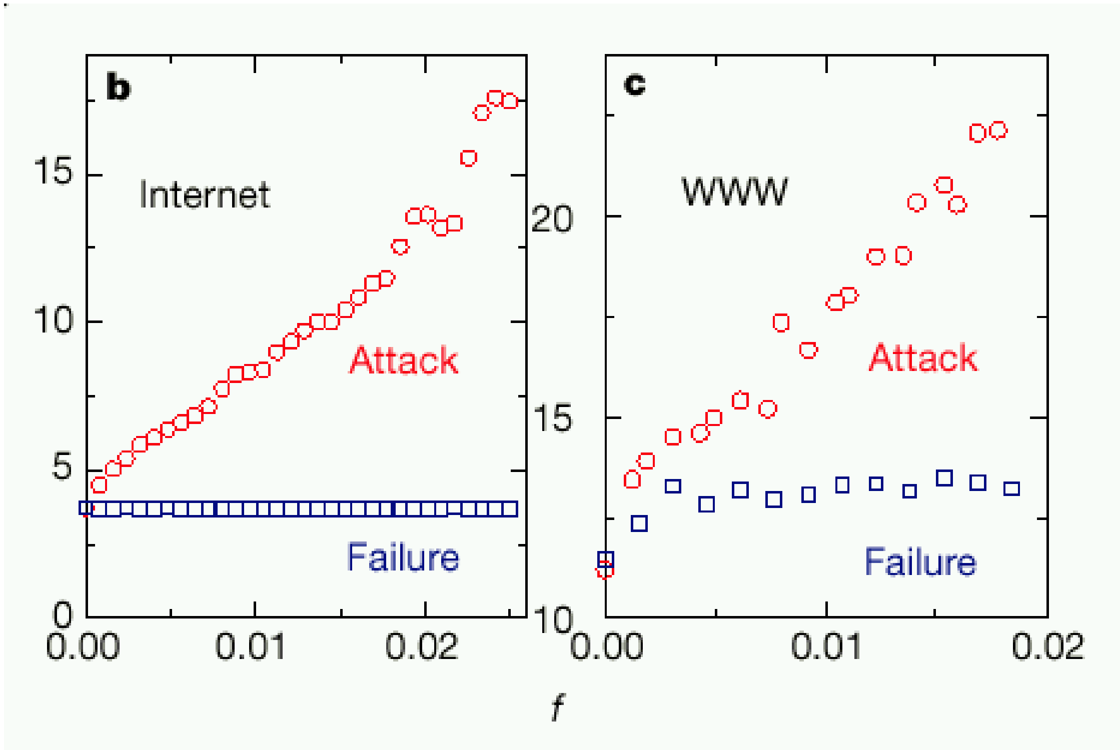

Image source: Albert et al. *Nature*, V406, 27 July 200.

Copyright Sana Security, Inc. 2003

Limits of the analogy

Sana Security™

- Neonatal tolerance
- Frequency of self vs nonself
- Confidentiality vs availability
- Discardable components
- Discardable events
- Maximizing human expertise

Copyright Sana Security, Inc. 2003

Summary

- Knowledge-based approaches to security are struggling
  - Cannot deal with increasing complexity
- A study of the IMS offers hope for securing increasingly complex IT systems
- But the body has co-evolved with the IMS to be easier to protect
  - Cells are cheap and discardable
  - Damage is incremental and easy to measure
- In the long-run, we need to redesign the systems being protected
  - Very different from old notion of trusted computing base
  - Design for false-negative tolerance, e.g. network of desktops
  - Design for false-positive tolerance, e.g. scale-free networks

Copyright Sana Security, Inc. 2003