

Intrusion Tolerance by Unpredictable Adaptation (ITUA)

Probabilistic Validation of Intrusion Tolerance



Michel Cukier, William H. Sanders, and Franklin Webber

OASIS PI Meeting, March 13, 2002

Motivation

- Intrusion tolerance is an emerging approach to security that aims to increase the likelihood that an application will be able to continue to operate correctly in spite of malicious attacks that may occur and may result in successful intrusions.
- Before intrusion tolerance can be accepted as an approach to providing security, techniques must be developed to validate its efficacy.
- Validation should be done:
 - During all phases of the design process, to make design choices
 - During testing, deployment, operation, and maintenance, to gain confidence that the “amount” of intrusion tolerance provided is as advertised.

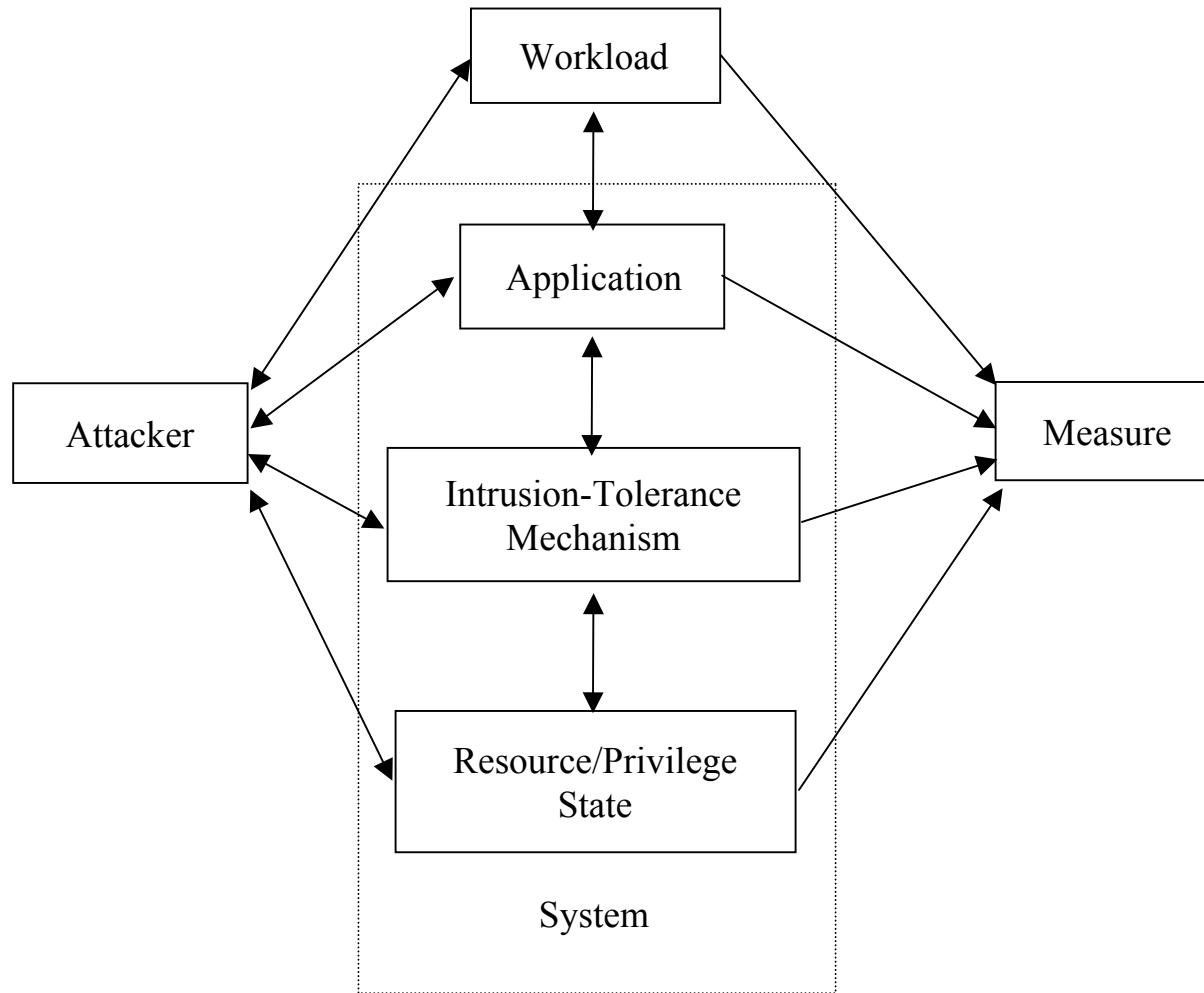
Existing Validation Approaches

- Most traditional approaches to security validation have focus on avoiding intrusions (non-circumventability), or have not been quantitative, instead focusing on and specifying procedures that should be followed during the design of a system (e.g., the Security Evaluation Criteria [DOD85, ISO99]).
- When quantitative methods have been used, they have typically either been based on formal methods (e.g., [Lan81]), aiming to prove that certain security properties hold given a specified set of assumptions, or been quite informal, using a team of experts (often called a “red team,” e.g. [Low01]) to try to compromise a system.
- Both of these approaches have been valuable in identifying system vulnerabilities, but probabilistic techniques are also needed.

Existing Probabilistic Approaches

- Earliest work on probabilistic quantification of security (that we know of) was done by Littlewood et al. [Lit93]. Exploratory work; Investigated the similarities between dependability and security
- Jonsson et al. [Jon97] presented a quantitative model of the security intrusion process based on attacker behavior
- Gong et al. [Gon01] present a general 9-state model of an intrusion-tolerant system for describing known and unknown security exploits by considering the impacts of attacks, rather than explicitly representing vulnerabilities that can lead to intrusions.
- Jha et al. [Jha01] proposes that state-level modeling, formal logic, and a Bayesian analysis be used together to quantify the survivability of a system.
- Ortalo et al. [Ort99] propose modeling known vulnerabilities in a system using a “privilege graph” (a privilege graph is similar to the scenario graph described above).

Proposed Probabilistic Validation Framework



Probabilistic Research Validation Issues

- What is the appropriate level of detail/abstraction for various model components?
- What is an appropriate interface between submodels; can they be constructed in a way that they can be used in a mix/match fashion?
- What are appropriate measures of intrusion tolerance?
 - Binary - “Does the realization meet the specification?”
 - Multivalued - Reward/Penalty (or Cost) analysis
- Can we quantify the likelihood of the assumptions that are made (and use ideas from assumption coverage work and risk analysis to provide overall assessment)?
- How can measurement/intrusion injection results be coupled with models to provide meaningful overall assessments?

Our Approach

- Take concrete approach, look at redundancy management as an example, and consider models at various levels of detail/abstraction
- Considered two models to date:
 - High-level model analyzed symbolically to determine environmental constraints necessary to achieve desired level of intrusion tolerance (Franklin).
 - More detailed model numerically analyzed to understand finer grained tradeoffs between system and environmental parameter values (Bill).
- Generalize models and refine framework
- Move beyond comparative studies, using experimental data to parameterize developed models and directly measure intrusion tolerance.

Redundancy Management for Intrusion Tolerance: A Common Idea

- At least 8 of 23 OASIS projects use some form of redundancy management as an intrusion-tolerance mechanism (based on data from last year's OASIS CD-ROM).
- Basic goal: increase availability while preserving integrity

A Simple Model

- Redundant components are distributed across multiple security domains
- Attacker can infiltrate at most 1 domain in any interval D , gaining “root” privilege in that domain (a “staged attack”)
- Any infiltration is detected with probability P ; false alarm rate is Q (uncorrelated)
- Redundancy management takes reconfiguration time R , and has communication overhead that depends on redundancy management protocol

Apply Model to ITUA

- Active replication used in conjunction with other defense mechanisms (e.g., dynamic reconfiguration of firewalls) and mechanisms coordinated by a nondeterministic defense strategy
- Byzantine fault tolerant replica management
 - intrusion detection is necessary in the worst case because effects of infiltration can be delayed

Analysis

- Estimate useful life of system under worst-case attack
 - useful life: time available for computation after redundancy management overhead is subtracted
 - worst case: attacker infiltrates as fast as possible
- Depends on redundancy management **policy**
 - e.g. “replicate to tolerate K failures and replace replicas in any detectably infiltrated domain”
 - e.g. “increase tolerance to $K+1$ after first detected infiltration”
- Analysis permits comparison of policies and allows intelligent choice of tunable parameters (e.g. “ K ”)

Basic Result

- To gain an order-of-magnitude increase in useful life:
 - $P > 1 - K/9$
 - policy of replacement only
 - If $K=1$ (four-fold replication), infiltration must be detected with greater than $8/9$ probability
 - false alarms and redundancy management overhead, neglected here, increase this burden on intrusion detection
- See <http://itua.bbn.com> for details

Conclusion

- Replication management **must** be complemented by other defense mechanisms, e.g.,
 - design diversity, so attacker is unlikely to be able to infiltrate all domains and less likely to make components fail Byzantine;
 - unpredictability, so attacker is slowed down by the need to discover facts about the defense;
 - dynamic containment of the attack, to increase the chance the attacker can be blocked completely;
 - tools for rapid operator intervention, to make increases in useful life smaller than 10X more valuable.
- ITUA is implementing some of these and assuming others

Work In Progress and Near-Term Plans

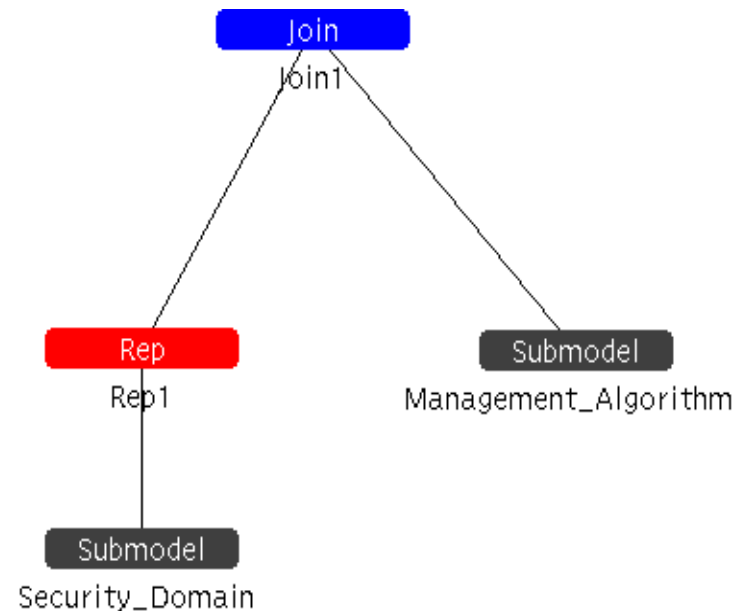
- Model other mechanisms for responding to and containing an attack
- Add diversity (of operating systems and of application components) to model
- Ensure that resulting models could be applied to as many OASIS projects as possible

Detailed Model Description

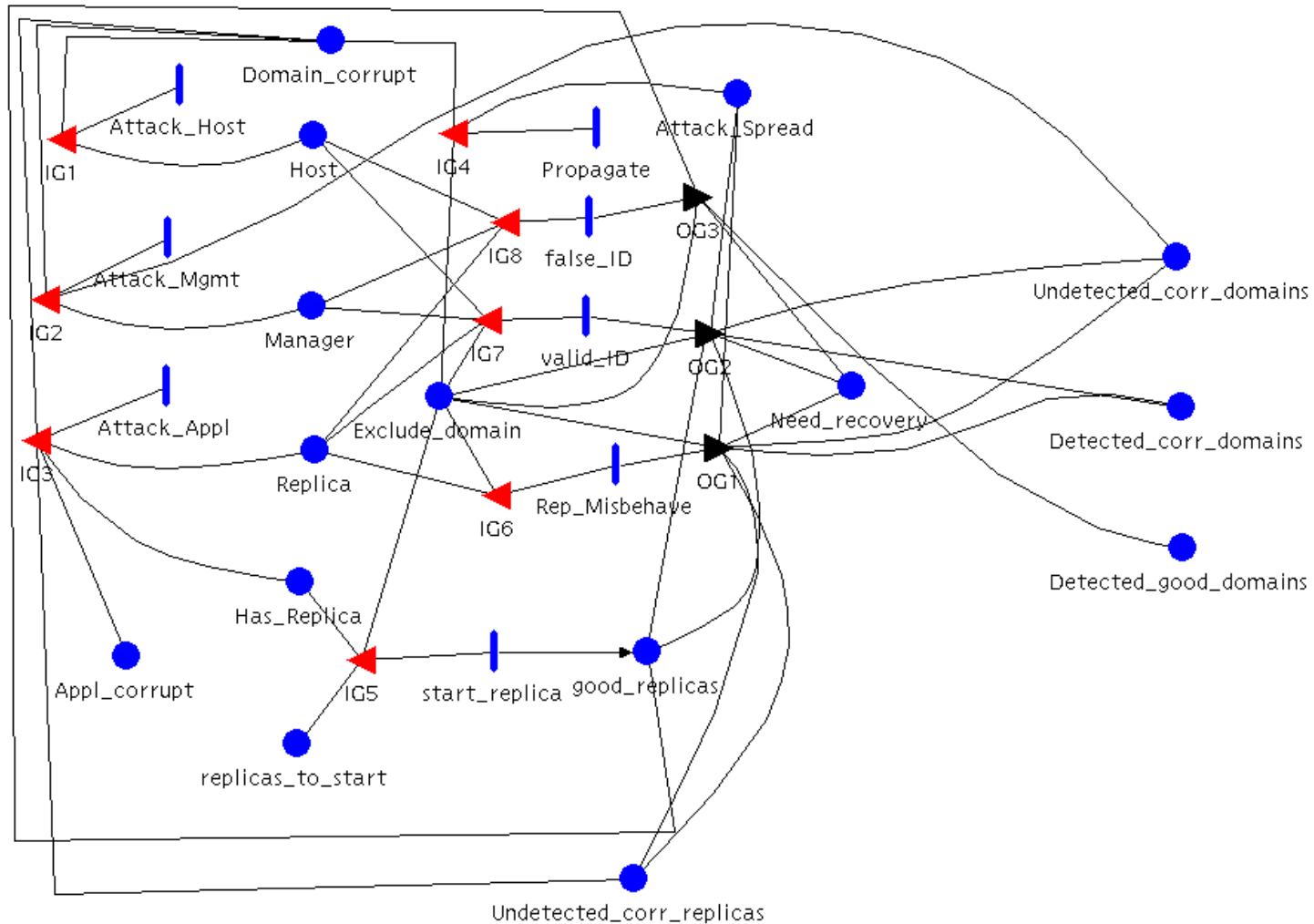
- System consists of predetermined number of security domains
- One host per security domain
- Effects of attacks on application, intrusion-tolerance mechanism, and host considered separately, but with attack rate that can depend on number previous successful attacks
- Intrusion detection system deployed on hosts – can detect valid intrusions as well as raise false alarms
- System service is proper if no more than one third of the replicas are in corrupt state
- Measures considered
 - *Unreliability*: probability system service remains proper through mission time
 - *Unavailability*: fraction of time system service is improper during mission time

Composed Model

- Model composed of multiple atomic Stochastic Activity Networks (SANs); solved using Mobius modeling tool (<http://www.crhc.uiuc.edu/PERFORM>)
- Two sub-models
 - *Security Domain*
 - replicated multiple times – each submodel represents one host possibly running one replica.
 - *Management Algorithm*
 - Creates new replicas when corrupt replicas are detected if non-infiltrated security domain is available for use



SAN for a Security Domain

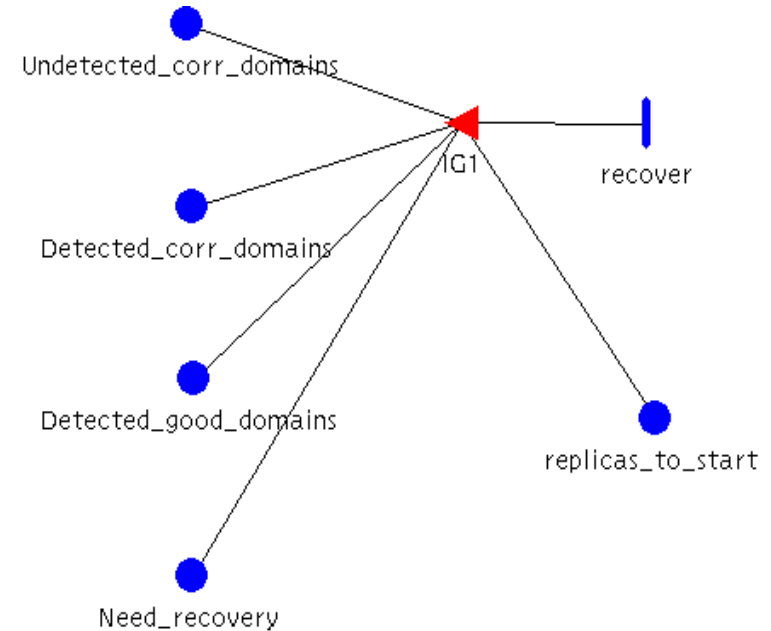


SAN for a Security Domain, cont.

- Each security domain composed of a host and a manager. May be running an application replica (indicated by *Has_Replica*)
- Attacker can target any of these entities simultaneously. Intrusion of host (OS) leads to increased vulnerability of manager and replica
- Successful intrusions aid in propagating attack to other domains (due to attacker learning etc.) as represented by activity *Propagate* and place *Attack_Spread*
 - Rates of attack activities increasing functions of attack spread
- A corrupt replica may exhibit corrupt behavior that leads to detection by other replicas in the replication group (provided there are enough good replicas)
- Intrusion Detection can be valid or a false alarm. Both lead to exclusion of the domain if its manager is not corrupt yet
- Replicas started randomly among various security domains
- Model keeps track of corrupt (detected and undetected) replicas and domains.

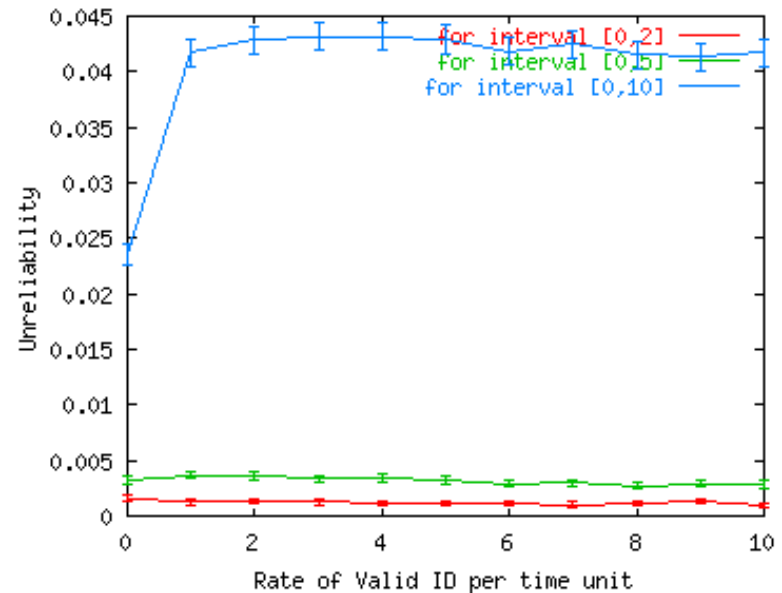
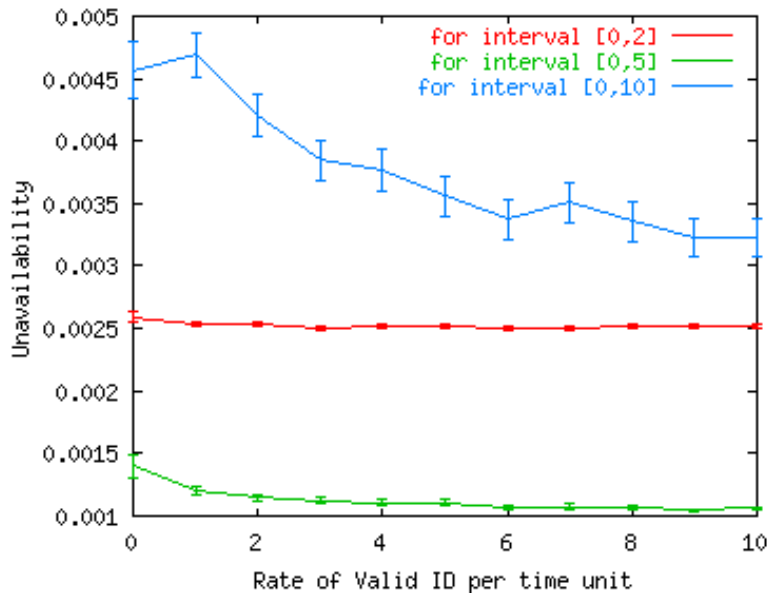
SAN for Management Algorithm

- Checks if there is a need to start new replicas (indicated by *Need_recovery*)
- Checks if there are enough good managers to reach a consensus
- If consensus reached, starts a new replica in a random non-corrupt domain



Example Results

Unavailability and unreliability vs. rate (efficiency) of valid ID, with no false alarms. One time unit = 1 week. Cumulative attack rate on the system is 3 successful intrusions per week. System left entirely to itself, without any human intervention.

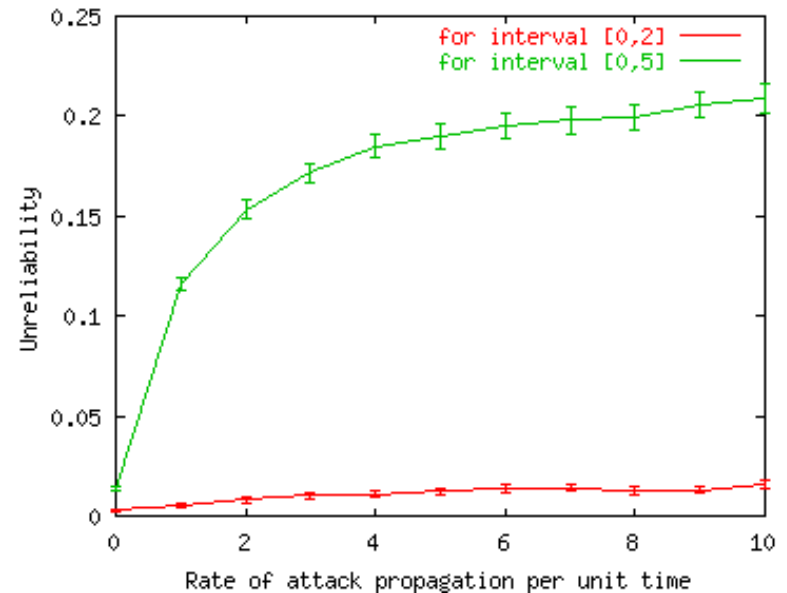
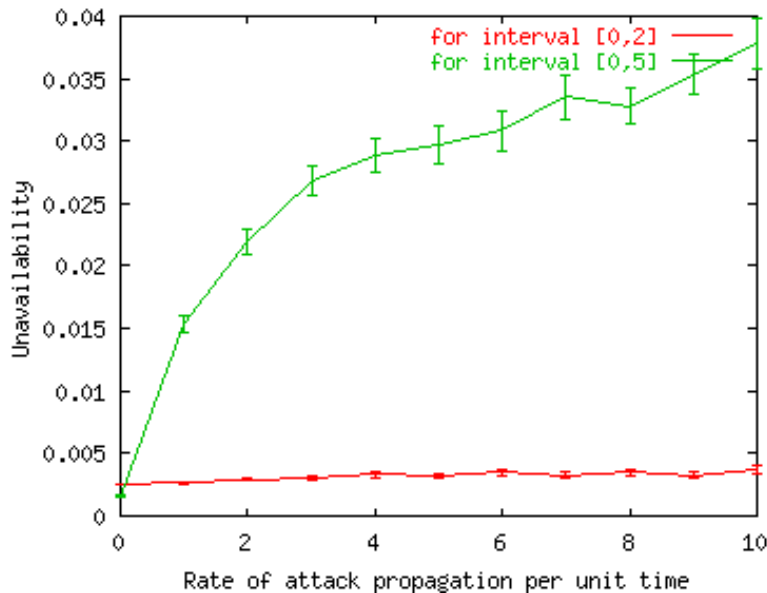


Observations:

- System has reasonably high availability and reliability, even when left exposed to high attack rate for long time
- Changes rate of valid Intrusion Detection does not change the behavior of the system in parameter range studied when false alarm rate kept constant

Example Results, cont.

Unavailability and unreliability vs. rate of attack propagation. The base attack rates are again roughly 3 intrusions per week on the system. One time unit \cong one week.



Observation:

- There is competition between propagation of attack and its containment by the management subsystem (through starting of new replicas and exclusion of corrupt domains.) This indicates that system reliability is high if it is difficult for an attacker to cross-over from one security domain to another (i.e. low rate of propagation)

Moving Beyond Comparative Studies

- Parameters need to be estimated in the different components of the probabilistic security model structure
- Attacker model:
 - Need to estimate attack rates for specific circumstances. Jonsson showed that attackers can be categorized in 3 classes based on their experience; the rate of attacks originating from the 3 classes of attackers has not been estimated
- Application model:
 - Application parameters highly dependent on the application
 - Applications need to be instrumented to estimate the parameters
- Resource and privilege state model:
 - The model includes OS and services
 - Vulnerability checking tools have been used for vulnerability and error configuration checking, and password guessing
 - Methods are needed to estimate network-wide parameter values
- Intrusion-tolerance mechanism model:
 - The location of the intrusion-tolerance mechanisms will impact the way of estimating the related parameters

Some Parameters of the Presented Models

- For both presented models, we list some of the key parameters
- Abstract model:
 - Interval of duration D between security domain infiltration
 - The fraction $1-B$ of computation taken by replica coordination
 - Blocking time R in the group membership protocol when adding/removing a replica
 - Intrusion detection system:
 - False alarm rate Q
 - Successful detection rate P
- More-detailed, SAN model:
 - Intrusion detection system:
 - False alarm rate
 - Successful detection rate
 - Host attack rate
 - Application/management attack rate
 - Attack propagation rate

Summary

- Probabilistic validation is an important technique for validating intrusion-tolerant systems
- It should be used in all phases of a system's lifecycle: requirement specification, early/detailed design, implementation, testing, deployment, and maintenance
- Models are useful for making comparative studies and evaluating design alternatives, even if exact parameter values are not known
- Better parameter value estimation is necessary, for implemented systems, to quantify intrusion tolerance obtained
- More work is needed to build better models, and to better determine input parameter values