Surviving Attacks and Intrusions: What can we Learn from Fault Models

This discussion is based in part on the paper:

What is your answer?

- Given an existing system or application, what are the impacts of *adjustments* in the *fault assumptions*?

- Given an existing system or application, what are the impacts of *adding* or *subtracting* security features?

- What are the impacts of *infrastructure changes* on performance or any of the “-ility” requirements?
Approach

- Let’s first try to define our system with respect to its functionalities

- Then we consider what fault models should be used for individual functionalities

- Then we need to see if the infrastructure can support what we need or

- How the infrastructure can help us to get to where we need to be.
How Survivable/Resilient is my System?

- Lessons learned from Fault-tolerance
- FT design: the possible and the impossible
Design for Testability

- Testing electronic circuits
  - Test pattern generation problem is NP-hard
  - Solution: Design for Testability
    - e.g. SCAN, partial SCAN
Design for Survivability

- When Systems become too complex
  - Design by Integration of Survivability mechanisms
  - Build-in *not* add-on
  - Design for Survivability has surfaced in different contexts
Design for Analyzability

- Not a new concept
- e.g., Series-Parallel RBD
- Not all systems are Series-Parallel!

![Diagrams showing series-parallel configurations]
Fault Models play Critical Role

- All Faults
  - Malicious
    - Asymmetric
      - Transmissive Asymmetric
      - Strictly Omissive Asymmetric
    - Symmetric
      - Omissive Symmetric
      - Transmissive Symmetric
  - Benign
    - Asymmetric
    - Symmetric
    - Transmissive
    - Benign
Fault Assumptions

- Do hybrid fault models apply outside of fault tolerance?
  - Many mechanisms from security & fault-tolerance exist
  - BUT in the end, their impact on the faults they can produce is what really counts
Fault Assumptions

Example: authentication

- authentication mechanism reveals fault
- potentially benign, depends on how many nodes are affected
- authentication is broken
- potential for symmetric or asymmetric

Slight departure from strict definitions of fault of the dependability community
System Definition

- A collection of Functionalities $f_i$
  - applications (software modules)
  - system components

- Fault Descriptions $F_i$
  - defines fault model with respect to functionality $f_i$
  - defines fault model that $f_i$ is designed to tolerate

$$System = \sum_{i=1}^{k} f_i$$

$$System = \bigcup_{i=1}^{k} f_i$$
System Definition

Fault Descriptions $F_i$

- example: communication with authentication
  - if authentication is assumed uncompromisable:
    - $F_i = (b)$
  - if authentication is assumed to be compromisable:
    - $F_i = (b,s,a)$
Dynamic Environment

- What are the impacts of
  - changes in fault assumptions
  - security feature availability (or their failure)?

Boils down to the analysis of $f_i$ in the context of $F_i$ and its support infrastructure
Important Questions:

- Given an existing system or application, what are the impacts of **adjustments** in the **fault assumptions**?

- Given an existing system or application, what are the impacts of **adding** or **subtracting** security features?

- What are the impacts of **infrastructure changes** on performance or any of the “-ility” requirements?
System Analysis

- Quantification of survivability under assumption of
  - fault model
  - e.g. hybrid fault model
  - fault environment
    - very complex as it addresses statistical assumption about the faults themselves, e.g.
      - fail rates
      - hazard function
      - independence or dependence of faults...
Model Analysis

- Reality however is moving towards “UUUR Events”
  - Unpredictable, latent,
  - Unobserved and
  - Unobservable Risks
Model Analysis

- Recent introduction of 3-layer survivability analysis architecture [Ma & Krings 2008]
  - tactical, strategic, and operational level

- Key observation: fundamental definition in survival analysis is survivor function $S(t) = Pr(T>t)$, which has same definition as reliability function

- hazard function $h(t)$ and cumulative hazard function $H(t)$ even use same terminology, besides common mathematical definitions.
Advantages of survival analysis:

1) more flexible, time-variant or covariates-dependent hazard functions

2) built-in procedures to deal with censored events

3) multivariate failure beyond binary failure

4) more effective modeling for dependent failure events though competing risks and shared frailty modeling

Our focus is on the hazard functions in 1)
Constant Hazard Function

- Simplest model: constant fail rate
  - Failures follow exponential distribution
  - Hazard function \( h(t) = \lambda \)
    - used in traditional reliability model (with constant fail rate) is not generally suitable

\[
R(t) = e^{-\lambda t}
\]

- strength
- weakness
- applications: RBD, FT, Markov Chain, Petri Net
Cox Prop. Hazards Model

“Fundamental Model of Survival Analysis”

Hazard Function is a function of time \( t \) and covariate vector \( z \):

\[
\lambda(t, z) = \lambda_0(t)e^{Z\beta}
\]

Extensions of PHM: time-dependent covariates

- unstratified PHM \( \lambda[t; z(t)] = \lambda_0(t)e^{Z(t)\beta} \)
- stratified PHM \( \lambda_j[t; z(t)] = \lambda_{0j}(t)e^{Z(t)\beta}, \quad j = 1, 2, ..., q \).
Model and State Changes

- Different functionalities can have different fault descriptions
- Different functionalities can utilize different hazard functions
- Each functionality may change its fault description and/or hazard function in time

Figure 1. Thread Model State Machine
Adaptation

- Integral feature in any design for survivability

- Adaptation addresses
  - dynamics of changing Fault Descriptions $F_i$
  - different definitions of fault descriptions (active, imposed)
Adaptation may be the result of diverse scenarios:

- The fault description is no longer valid due to specific event. For example, intelligence suggests that authentication is broken.

- The fault description of functionality should be strengthened by design. For example, $f_i$ is identified as the weakest link.

- Infrastructure that $f_i$ depends on has changed. For example, may not support tolerance to certain fault types anymore.
Fault Model Adaptation

Active Fault Description: \( F_i \)

- fault model that system (functionality) currently subscribes to, i.e.,
- the faults that \( f_i \) assumes to be able to tolerate or deal with
- for \( f_i \) fault description \( F_i \) represents the active fault description
- \( F_i \) is determined by system designer (designer of \( f_i \))
Fault Model Adaptation

- **Imposed Fault Description:** $\hat{F}_i$
  - the fault model the infrastructure of system imposes on $f_i$
  - encompasses those fault types the system (or application) has to explicitly deal with by distinct mechanisms
  - Example $\hat{F}_i = (b, s)$
    - for given infrastructure benign and symmetric faults are possible and theoretically unavoidable
    - note that no asymmetric faults are listed (there is no “a”)
    - infrastructure is assumed to be capable of theoretically eliminating this fault type, e.g., broadcast network
Example TCP/IP

1) Assume TCP/IP provides reliable transmission
   - W.r.t. infrastructure this leads to \( \hat{F}_i = (s, a) \)
     - there are no benign (omission) faults
     - value fault (s and a) cannot be resolved without explicit mechanisms

2) Now assume that TCP times out
   - Leads to \( F_i = (b, s, a) \)
     - benign fault was added
Example

Interesting case: authentication is compromised

- introduced value faults (s,a)
- explicit mechanisms need to be added
- symmetric: $N > 2s$
- asymmetric: $N > 3a$

- not only requires higher degree of redundancy
- but agreement algorithm
Example

Authentication example cont.

System designer choices:
- live with the risk of authentication compromises
- pay the cost of module and message overhead

But how high is that cost?
- depends on desired \( s \) and \( a \)
- in addition: common mode fault need to be addressed
Design Changes

- Imposed fault description gives insight about what infrastructure cannot inherently deal with
  - allows for adaptation

Authentication example

- assume authentication may be compromised:
  \[ F_i = \hat{F}_i = (b, s, a) \]
  asymmetric faults are a problem!

- changing to broadcast protocol we can avoid asymmetries
  \[ \hat{F}_i = (b, s) \]
Adaptive Policies

- Select the lowest overhead solution possible under a given threat level

- Similar to the “shifting gear” approaches used in agreement algorithms
Infrastructure Changes

- What happens if infrastructure used by $f_i$ change?
  - Any changes to the imposed fault description?
  - Carefully analyze the implication of the changes
    - can be good or bad news
    - misjudging fault descriptions may render application non-survivable!
Conclusions

System Definition
- functionalities, active/imposed fault models

System Analysis
- Model Analysis (include UUUR)
- Resilience based on active and imposed fault descriptions
- Adaptation (functionalities and fault models)
  - different fault description
  - different hazard functions
  - dynamic fault descriptions and/or hazard functions