Hierarchical, model-based risk management of critical infrastructures

A Paper by F. Baiardi, C. Telmon, D. Sgandurra
Motivation

- Large number of infrastructure components
- Large number of interdependencies among the components
- Simplification desired
The infrastructure hypergraph is a labeled directed hypergraph.

Components of the system are nodes.

Dependencies in the system are hyperarcs.

The tail label is the attribute of the tail component to be controlled.

The head label is the attribute of the head component that is controlled.
Infrastructure hypergraph

- confidentiality, c
  - ability of reading the component state

- integrity, i
  - ability of updating the component state

- availability, a
  - ability of managing the component, of determining who can invoke its operations

Fig. 1. A dependency hypergraph.
Users and attacks

- Users have rights (legal and illegal)
  - set of component attributes $U$ controls
- Transitive closure to determine full control of User by including dependencies
  - algorithm given
- Attacks have 3 parts
  - target component
  - preconditions (rights)
  - postconditions (new rights)
- complex attack is a sequence of attacks to achieve a goal
Evolution graphs

- model attack evolution with a directed acyclic graph
  - nodes are states
  - there is an arc from two nodes if an elementary attack transitions from one to the other
  - labeled by the user and attack

- can be autonomous or multiple users colluding
Risk

- Risk Analysis
  - Evaluate the probability of each evolution using the complexity and resources required

- Risk mitigation, implement countermeasures
  - remove a vulnerability
  - update dependencies
  - update initial rights of some users
  - increase resources required for attack
Hierarchical model

- Real infrastructure is large, analysis is computation intensive
- allow components of the infrastructure to be hierarchically decomposed
- model will be a high level abstraction as a simple hypergraph with the components expandable as further nodes.
- conditions mentioned which ensure reduction in computation cost
Hierarchical Model

Fig. 7. An example, (a) A high-level description of a (subset) of a system and (b) a hierarchical decomposition of the hand-held device.

Fig. 4. Hierarchical decomposition of a component. (a) An infrastructure hypergraph and (b) hypergraph after the decomposition of C1.
Hierarchical Decomposition

- Hyperarcs allow decomposition of components
- Decomposability Condition 1: No low-level evolution results in new rights for any user.
- Decomposability Condition 2: A low-level evolution may result in new rights for a user only if these rights do not enable any elementary attack of a high-level evolution.
- Decomposability Condition 3: A low-level evolution may result in new rights for a user only if these rights do not increase the number of elementary attacks of high-level evolutions the user can implement.
- These are all sufficient conditions but not necessary conditions.
- Decomposability Condition 4: Further countermeasures are introduced to stop any low-level evolution that violates the previous condition.
- Defining a necessary and sufficient condition is complex. It depends on user, vulnerabilities of and attacks against components not involved in the decomposition.
- Extend to decomposing multiple components to a hypergraph.
Relevance

- General model founded on hypergraph
- Might allow for hypergraph algorithms to determine risk
- Statistical hypergraph prediction of attacks
- Allows for hypergraph research on an explicit or general example of infrastructure