Pareto-Optimal Situaton Analysis for Selection of Security Measures

Andres Ojamaa

Joint work with Jüri Kivimaa and Enn Tyugu

Institute of Cybernetics at TUT

CS Theory Days, Feb 1 2009, Kääriku

Outline

Introduction
- Background and Motivation

Graded Security Model
- Security Goals
- Parameters and Functions
- Optimizing Security Measures
- Discrete Dynamic Programming

Graded Security Expert System

Example
- Visual Specification
- Example of Results

Security Situation Management

- The aim is to provide the best possible security of a system with given amount of resources.
- At the same time at least the standard requirements should be satisfied, if possible.
- Solutions are usually needed yesterday. Therefore detailed risk analysis is not a good option.
- The goal is achieved by coarse-grained analysis of security situation and optimisation of resource usage.

Security Awareness Simulation Games

- CyberCIEGE — video game and tool to teach network security concepts (2005)
- CyberProtect — DISA-produced game that includes hacker attacks and budget constraints (1999)

Situation Description: Security Goals

Security class is determined by security levels, associated with security goals:

- confidentiality (C),
- integrity (I),
- availability (A),
- non-repudiation (N).

e.g. `C2 I1 A1 N2`

The model can be *extended* by adding security goals.

Situation Description: Parameters of the Model

- Available resources — r
- Integral measure of security — S
- Security measures groups — $g_1, g_2, \ldots, g_n$
- Security levels of measures groups — $l_1, l_2, \ldots, l_n$
- Security confidences granted by measures groups — $q_1, q_2, \ldots, q_n$
- Relative importance of measures groups: weights — $a_1, a_2, \ldots, a_n$, where $\sum_{i=1}^{n} a_i = 1$

Abstract Security Profile

An *abstract security profile* p is an assignment of security levels to each group of security measures:

$$p = (l_1, l_2, \ldots, l_n)$$

Cost Function

The cost function h gives the costs $h(l, g)$ required for implementing security measures of a group g for a level l.

The costs of implementing a given abstract security profile:

$$costs(p) = \sum_{i=1}^{n} h(l_i, g_i)$$

Goal 1: Keep the value of $costs(p)$ as low as possible.

Levels Requirement Function

Function s produces a required security level $s(c, g)$ for a group g when the security class is c. The requirements may be prescribed by security standards such as BSI, NISPOM or ISKE.

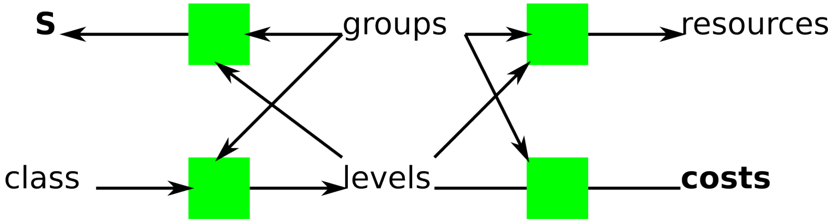
Integrated Security Metrics

The overall security of a system is described by means of an integrated security metrics (integral security confidence) S.

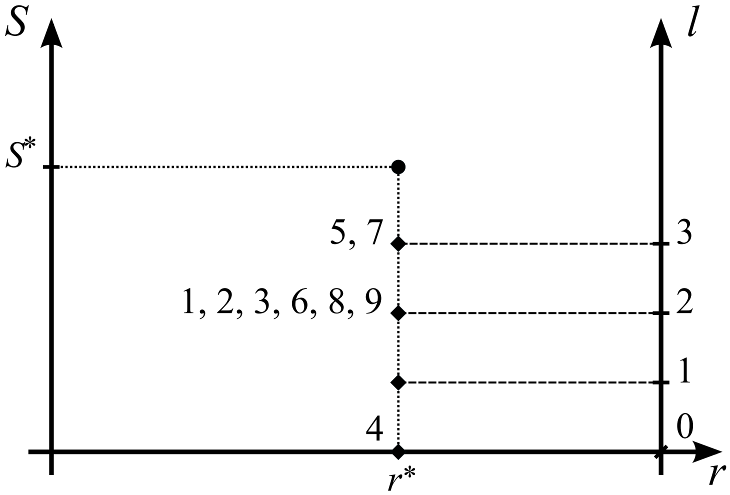
$$S = \sum_{i=1}^{n} a_i q_i$$

Goal 2: Increase security confidence of a system.

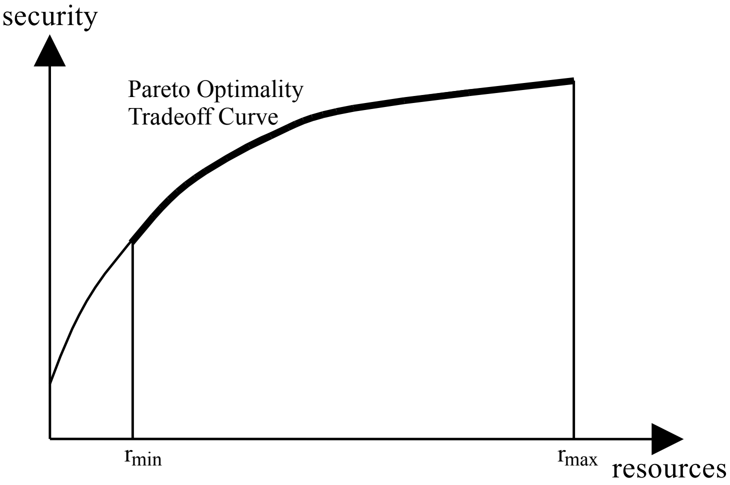
Dependencies



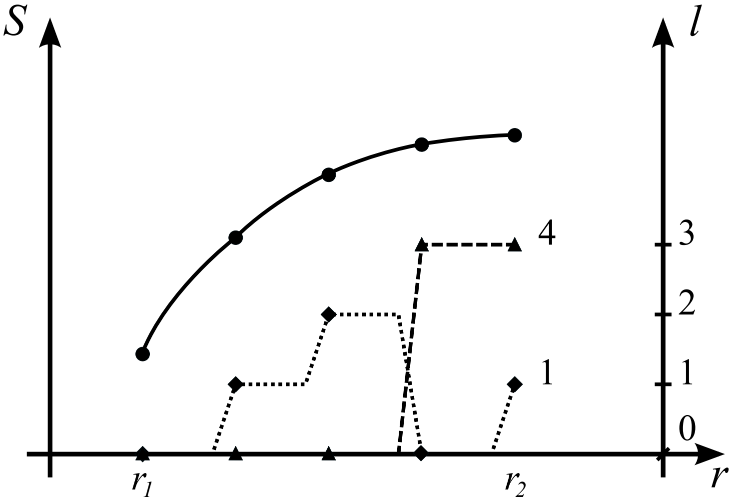
Conventional Graded Security Solution



Pareto-Optimality Curve



Pareto-Optimal Security Solutions



Dynamic Programming

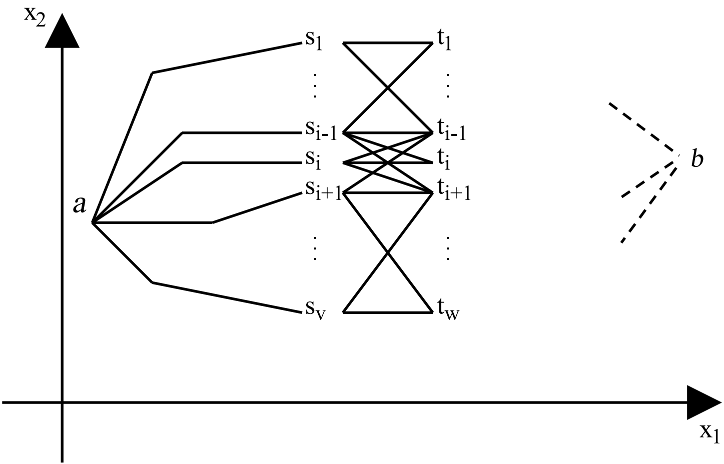
Building optimal solutions gradually, for $1, 2, \ldots, n$ security measures groups enables us to use discrete dynamic programming, and to reduce considerably the search.

The fitness function S defined on intervals from j to k as

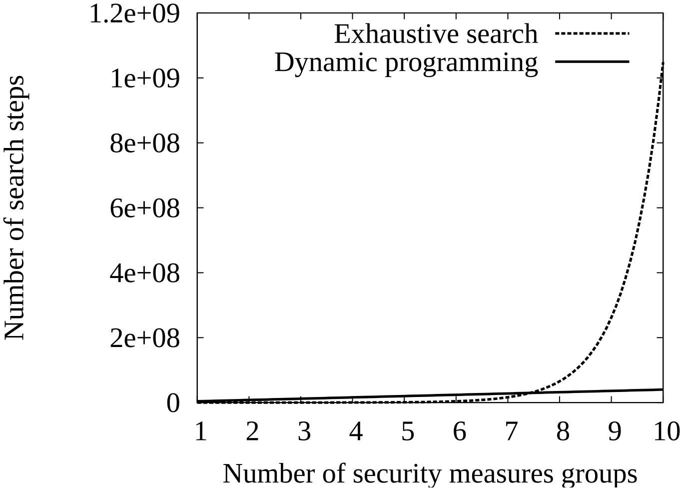
$$S(j,k) = \sum_{i=j}^{k} a_i q_i$$

is additive on the intervals, because from the definition of the function S we have $S(1,n) = S(1,k) + S(k,n)$.

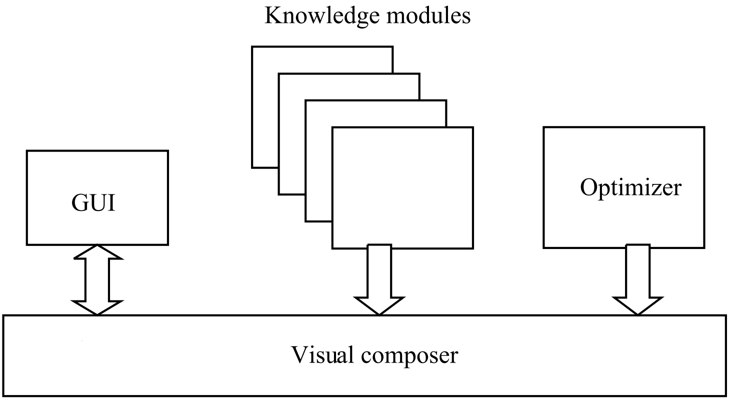
Discrete Dynamic Programming



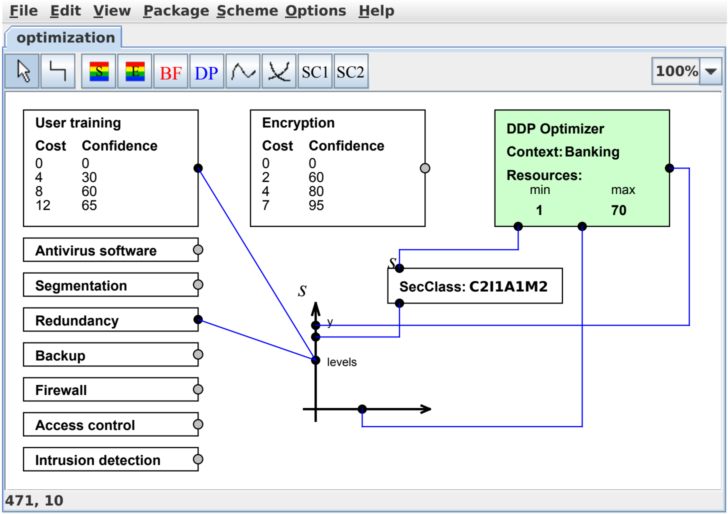
Complexity Compared



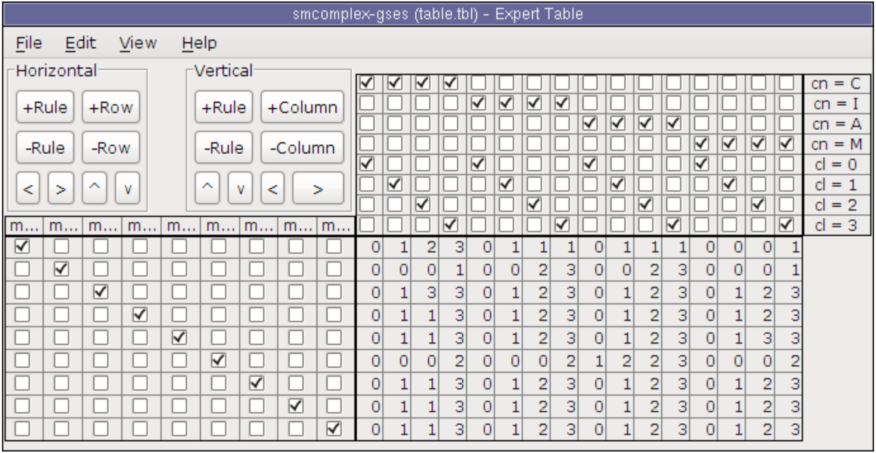
Graded Security Expert System



Visual Specification



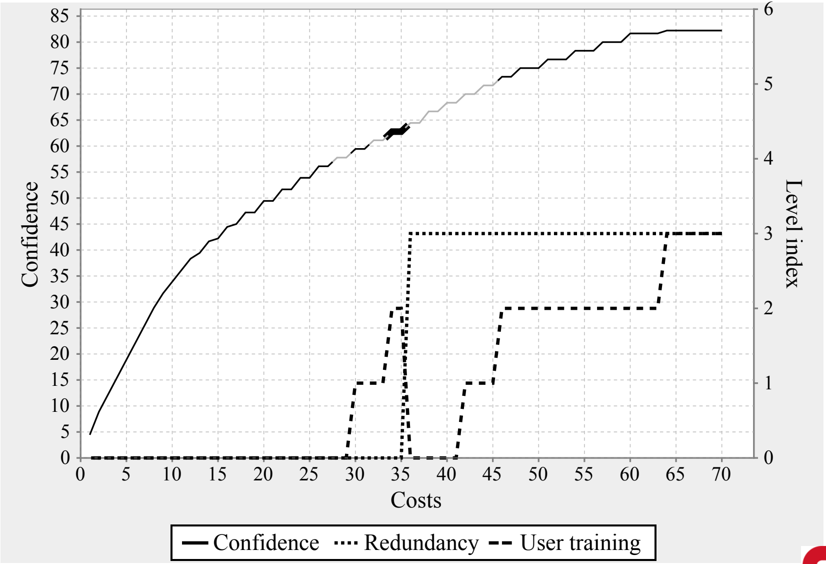
Knowledge Modules as Decision Tables



✓	✓	✓	✓													cn = C
				✓	✓	✓	✓									cn = I
								✓	✓	✓	✓					cn = A
												✓	✓	✓	✓	cn = M
✓				✓				✓				✓				cl = 0
	✓				✓				✓				✓			cl = 1
		✓				✓				✓				✓		cl = 2
			✓				✓				✓				✓	cl = 3

m...	m...	m...	m...	m...	m...	m...	m...	m...																
✓									0	1	2	3	0	1	1	1	0	1	1	1	0	0	0	1
	✓								0	0	0	1	0	0	2	3	0	0	2	3	0	0	0	1
		✓							0	1	3	3	0	1	2	3	0	1	2	3	0	1	2	3
			✓						0	1	1	3	0	1	2	3	0	1	2	3	0	1	2	3
				✓					0	1	1	3	0	1	2	3	0	1	2	3	0	1	3	3
					✓				0	0	0	2	0	0	0	2	1	2	2	3	0	0	0	2
						✓			0	1	1	3	0	1	2	3	0	1	2	3	0	1	2	3
							✓		0	1	1	3	0	1	2	3	0	1	2	3	0	1	2	3
								✓	0	1	1	3	0	1	2	3	0	1	2	3	0	1	2	3

Example of Results



Future Work

- Combine the optimization package with risk analysis tools (e.g. attack trees)?
- Improve the visual language and the user interface
- Collect and accumulate expert knowledge and real data
- Experiments with real data
- Implement dependant measure groups
- Analyze sensitivity of results wrt inaccurate input data

Summary

A CoCoViLa package was developed to help the IT manager/security expert answer the following questions quickly:

- How much resources are needed to achieve the required level of information security?
- What is the best way to spend the IT security budget?

References

- CoCoViLa — Compiler Compiler for Visual Languages, `http://www.cs.ioc.ee/~cocovila`
- CyberCIEGE — `http://cisr.nps.edu/cyberciege/`
- CyberProtect — `http://iase.disa.mil/eta/online-catalog.html`
- E. Tyugu. Algorithms and Architectures of Artificial Intelligence. IOS Press, 2007.
- A. Ojamaa, E. Tyugu, J. Kivimaa. Pareto-optimal situation analysis for selection of security measures. In: MILCOM 08: Assuring Mission Success: Unclassified Proceedings, November 17-19 San Diego, 2008, 7 p.