When Good Models Meet Bad Data:
Applying Quantitative Economic Models to Qualitative Engineering Judgments

Shawn Butler
Computer Science Dept
School of Computer Science
Carnegie Mellon University
Pittsburgh Pa 15213
+1 412 268 8101
shawn.butler@cs.cmu.edu

Somesh Jha
Computer Science Dept
School of Computer Science
Carnegie Mellon University
Pittsburgh Pa 15213
+1 412 268 5544
somesh.jha@cs.cmu.edu

Mary Shaw
Institute for Software Research, Int'l
School of Computer Science
Carnegie Mellon University
Pittsburgh Pa 15213
+1 412 268 2589
mary.shaw@cs.cmu.edu

SYNOPSIS

My model for choosing investments
Expects to get ratio-scale measurements
But for software design
Even ordinal is fine
Can the model give reasonable guesstimates?

ABSTRACT

We have been attempting to apply financial portfolio analysis techniques to the task of selecting an application-appropriate suite of security technologies from the technologies available in the marketplace. The problem structures are sufficiently similar that the intuitive guidance is encouraging. However, the analysis techniques of portfolio analysis assume precise quantitative data of a sort that we cannot realistically expect to obtain for the security applications. This will be a common challenge in applying quantitative economic models to software engineering problems, and we consider ways to address the mismatch.

Keywords: economic models in software engineering, security technology selection problem

1 INTRODUCTION

At the First International Workshop on Economics-Driven Software Engineering Research (EDSER) we proposed using portfolio analysis to assist software engineers in selecting security technologies [2]. We briefly described security technology selection constraints and suggested that portfolio analysis might be used to select a reasonable suite of technologies for a system. In ongoing work we have been exploring that possibility, and we have encountered a mismatch between the kinds of data we can reasonably expect security designers to produce and the kinds of data portfolio analysis tools are designed to operate with. It appears to us that this kind of mismatch will arise frequently in attempts to apply economic models to software engineering, so we address the resolution of this mismatch directly.

In this paper, we review the Security Technology Selection Problem and identify some of the difficulties in using portfolio analysis in this setting. We suggest three approaches to resolving these limitations and suggest ways these examples serve as models for the EDSER community

2 PORTFOLIO ANALYSIS AND THE SECURITY TECHNOLOGY SELECTION PROBLEM

The Security Problem

The problem faced by the security engineer of a system is to protect system resources from attack, specifically to reduce the risk of loss. *Threats* launch *attacks* that exploit system *vulnerabilities* in order to gain or deny access to system critical *resources*. A system security engineer must select a set of security *technologies* that minimize the expected *risk*. The set of selected security technologies is called the security *portfolio*.

The system security risk depends on the relevant set of potential attacks and the consequences of a successful attack. The significant threats (and consequently the attacks they may launch) and resources vary among systems, so security portfolios will also vary. Cost constraints and performance requirements prevent software engineers from simply selecting all the best technologies. Cost constraints include both short-term costs, such as the initial purchase and installation costs, and long-term maintenance costs. Maintenance often dominates the cost.

Other constraints also affect the choice of the security portfolio. For example, security engineers often want some breadth of coverage against different attacks. Ideally, there should be at least one countermeasure for each identified threat. However, in practice some threats may not be covered by countermeasures: for example, they might be unlikely to occur, or the consequence of an attack might be less than the cost of the security technology. Since most security technologies are not completely effective, engineers also want redundant countermeasures against an attack. For example, intrusion detection systems provide additional security against attacks that penetrate system firewalls, and encrypted data storage increases security against undetected intruders.

Security engineers must also consider how security technologies interact with other system and security technologies. For example,

- Heterogeneous system environments complicate technology selection. Some security products are only available for TCP/IP environments, certain operating systems, or for specific software versions.
- Distribution complications can arise when technologies are restricted from non-USA countries.
- Security technologies have synergistic effects when combined with other technologies. For example, authentication systems are more effective if passwords are randomly generated or if login attempts are restricted.

Security Technology Selection as Portfolio Analysis

There are many similarities between constructing a financial portfolio and selecting security technologies for an information system. The objective of both activities is to find the set of investments that best supports the investment objectives. For financial portfolio investment, the objective is the desired rate of return; for security systems, the objective is an acceptable risk threshold.

Security technologies can be viewed as investments and the degree to which the technology prevents damage from attacks is the return on investment. The idea behind financial portfolios is that the variance around a desired rate of return is less than individual portfolio elements, therefore the risk of getting that return is less. Similarly, a set of security technologies reduces the system risk and the constraints help to model the expected rates of return. The combination of technologies reduces the consequences of a successful attack.

Although portfolio analysis has been very useful in modeling the security selection problem, the technique has some limitations. Financial portfolios consist of financial instruments with various rates of return; these lend themselves to quantification. For security portfolios, however, the acceptable-risk objective requires substantial subjective evaluation.

The portfolio analysis analogy is helpful in setting up the security technology problem and in informally suggesting heuristic approaches to solution. However, we would like to use more of the knowledge available in the financial portfolio [3]. We turn now to some ideas about coping with the gap between the precision of security information and the mathematical expectations of portfolio analysis tools.

3 MISMATCH BETWEEN MODEL AND AVAILABLE DATA

The greatest impediment to using portfolio analysis is the lack of quantifiable data. This problem appears in two parameters of the security model, *threat expectation* and *security technology effectiveness*.

Software engineers select technologies based on threat expectations, or their anticipation of attacks, but there is rarely statistical data to support these expectations. Threat expectations are often phrased as “We are more likely to get hit by a disgruntled employee than by an internet hacker” or “I really worry that someone will …..”. Statements of this kind reflect attack expectations, but they don’t translate into probabilities.

Security technology effectiveness is also very difficult to quantify. Technology effectiveness depends on the type of attack, configuration and maintenance complexity, design, system characteristics, etc. Most security professionals can comment on the relative degree of effectiveness. For example, statements, such as “Technology ‘x’ is more effective than technology ‘y’”, or “Technology ‘x’ is very effective against this type of attack”, are common. In the end, the security engineer makes decisions based on partially ordered qualitative information.

4 ADDRESSING THE MISMATCH

Much of the mismatch between security technology data and financial analysis methods arises from the fact that the security technology data is expressed on ordinal scales ("X is more effective than Y") but the analysis methods are designed for data expressed on a ratio scale. (The appendix reviews this small aspect of measurement theory).

We consider three general approaches to resolving this discrepancy between the data and the analysis tools:

- increase the precision of the data enough to convert the qualitative rankings to quantitative measures
- find analysis techniques that require less precision
- demonstrate that the analysis technique of interest preserves the relations of the qualitative data

Converting qualitative information to quantitative ranges

Perhaps the easiest approach is to encourage the security professionals to provide a little more information -- enough to establish quantitative ranges for the estimates. Security professionals may be able, with encouragement, to provide upper and lower bounds when they cannot provide precise threat probabilities and effectiveness measures. Each threat, for example, could be characterized by the bounds on the probability of its occurrence. A security professional might be willing to say something like “I think there is a better than 50% chance that someone will try to modify our web pages, but maybe not more than an 80% chance” when he or she cannot say "the probability of this attack is 67.3%". Likewise, reasonable technology effectiveness commitments might come in the form “I believe this security technology will stop between 40 to 60 percent of the IP spoofing attacks”. Once the quantitative bounds have been established, our portfolio analysis techniques [3] can be used for best-case and worst-case analyses to conduct sensitivity analysis and partial ordering validation.

The limitation of using quantitative bounds is that the difference between the upper and lower bounds can be too great to achieve useful results. Large differences weaken the results, for example a lower bound of 10% and an upper bound of 90% doesn't provide much useful information. Further, the range estimates may not be independent of each other. Another limitation is that humans are notoriously bad at estimating probabilities so the upper and lower bounds can be very unreliable. Eventually, security researchers will establish threat probabilities based on empirical evidence, but we will still be left with estimating the effectiveness of a security technology. Other analytical techniques are necessary before we can have confidence in the results.

Use analysis based on partial orderings

Another approach to using qualitative data is to explore decision analysis techniques that use partial orderings. Our problem of selecting security technologies is similar to the *two-sided matching problem*. An example of this is the national intern selection problem. Each year new interns submit a partially ordered list of hospitals where they would like to work. Hospitals also create a partially ordered list of interns they would like to have work at their hospitals. The problem is to find a stable matching of interns and hospitals. Two-sided matching problems are widely studied in game theory and economics [5].

The two-sided matching problem in security is between security technologies and threats. We would like to find a stable matching so that each threat is paired with a security technology, and categories of technologies are selected. In our particular example, the security portfolio may not set up a one-to-one match between technologies and threats; this may require extension of the basic two-sided solutions.

Establish validity of using quantitative techniques after scale transformations

The third approach is to explore the idea of transform spaces. A common technique in mathematics and engineering is to convert data from normal space to a transform space, carry out computations in the transform space, and convert the results back to normal space. This is usually done because the computation in transform space is so much simpler than the corresponding normal-space computation that it offsets the cost of the transformations. The most familiar example is logarithmic space: adding logarithms corresponds to normal-space addition, and when this is done by manipulating a slide rule, the transformation cost is small. The figure shows, among other things, the calculation 1.51 * 2.33= 2.00. Other common transforms used in this way for specialized applications are the LaPlace and Fourier transforms.

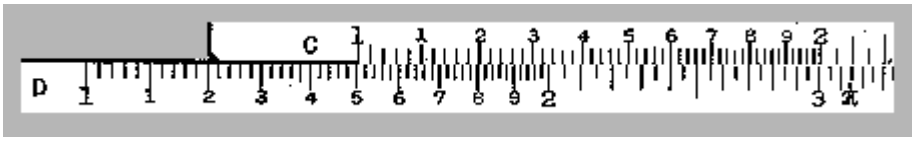
These common cases involve transformations on values rather than on the measurement scale, but they raise the tantalizing question of whether we can do a scale transform on our ordinal information to another scale, use analysis tools intended for that scale, and extract a useful result. In a much-simplified form of our model (ignoring resource constraints and the differential effectiveness of technologies against different threats, among other things),

> Let T be a vector of threat levels, expressed in an ordinal scale
>
> Let E be a vector of the effectiveness of security technologies, expressed in an ordinal scale
>
> We seek a portfolio selection function $S_o(T,E)$ that operates with ordinal scales and returns a set of indices $\{i_j\}$ corresponding to the technologies to include in the portfolio

We don't have a function S_o of this kind, but we do have a function S_r that performs a similar function with ratio scales. Can we find mappings M_T and M_E with the properties that $M_T(T)$ and $M_E(E)$ are expressed in ratio scales and that $S_r(M_T(T),M_E(E))$ yields results equivalent to $S_o(T,E)$, possibly after applying an inverse mapping on the result? This is clearly possible for functions S_r that perform only monotonic transformations and max/min functions. What restrictions on functions S_r allow them to be used in this way? Are these restrictions generous enough to include any useful functions?

REFERENCES

1. [BEM96] L. Briand, K. El-Emam, and S. Morasca. On the Application of Measurement Theory in Software Engineering. *Empirical Software Engineering*. 1(1), 1996.
2. [BCJRS99] S. Butler, P. Chalasani, S. Jha, O. Raz, and M. Shaw. "The Potential of Portfolio Analysis in Guiding Software Decisions." Position paper submitted to First Workshop on Economics-Driven Software Engineering Research, March 1999.
3. [BCJS00] S. Butler, P. Chalasani, S. Jha, and M. Shaw. Selecting a Portfolio of Security Technologies. Unpublished manuscript, 2000.
4. [FP97] Norman E. Fenton and Shari Lawrence Pfleeger. *Software Metrics: A Rigorous & Practical Approach*, International Thomson Computer Press, 1997.
5. [RS92] A.E. Roth and M. Sotomayor. Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis, *Econometric Society Monograph Series*, Cambridge University Press, 1990. Paperback edition, 1992.



5 APPENDIX: QUICK REVIEW OF MEASUREMENT THEORY

Not all measurements are created equal. More precise initial measurements enable more precise analyses and conclusions. Measure theory provides models that explain the differences and limitations.

Most members of this community are already familiar with this material, but many have forgotten the terminology. As a reminder, measure theory recognizes five scales for classification or measurement, ordered from less to more powerful [1,4]:

Scale	*Intuition*	*Preserves*	*Example*	*Legitimate transformations*
Nominal	Simple classification, no order	Differences	Horse, dog, cat	Any one-to-one remapping
Ordinal	Ranking according to criterion	Order	Tiny, small, medium, big, huge	Any monotonic increasing remapping
Interval	Differences are meaningful	Size of difference	Temperature in Celsius or Fahrenheit	Linear remappings with offset (ax+b)
Ratio	Has a zero point	Ratios of values are meaningful	Absolute temperature (Kelvin), values in currency units	Linear remappings without offset (ax)
Absolute	Exact	All relations	Number of instances	None

Some examples of abuse:

"The temperature in Miami is 20 degrees Celsius, the temperature in Pittsburgh is 10 degrees, so it's twice as hot in Miami." Wrong. Celsius is an interval scale, and this kind of comparison is only valid in ratio or absolute scales. The Kelvin temperature scale is an interval scale, so it's ok to convert to Kelvin and compare: "The temperature in Miami is 293 degrees Kelvin, the temperature in Pittsburgh is 283 degrees Kelvin, so it's 7% warmer in Miami."

"We surveyed the population for preferences on a scale of Strong Yes / Yes / OK / No / Strong No and coded the results on a 5-point scale with Strong Yes as 5 and Strong No as 1. Option A averaged 4.0, option B averaged 3.0, and option C averaged 2.0. Therefore option A dominated option B by as much as option B dominated option C." Wrong. The preferences are measured on an ordinal scale, and the comparison requires at least an interval scale. This sort of comparison is especially noxious when coupled with comparisons of the costs of the options. This is the kind of problem we're addressing in this paper.