MANAGING UNCERTAINTY IN SECURITY RISK MODEL FORECASTS WITH RAPSA/MC

James R. Conrad,[1] Paul Oman,[2] and Carol Taylor[3]
Department of Computer Science, University of Idaho, Moscow, ID 83844-1010
[1]conr2286@uidaho.edu, [2]oman@uidaho.edu, [3]ctaylor@uidaho.edu

Abstract This report describes an information security risk assessment process that accommodates uncertainty and can be applied to deployed systems as well as systems under development. An example is given for a critical infrastructure but the technique is applicable to other networks. RAPSA/MC extends the Risk Analysis and Probabilistic Survivability Assessment (RAPSA) systems-level process model with a Monte-Carlo (MC) technique capturing the uncertainty in expert estimates and illustrating its resulting impact on the model's forecast. The forecast is presented as a probability density function enabling the security analyst to more effectively communicate security risks to financial decision makers. This approach may be particularly useful for visualizing the risk of an extreme event such as an unlikely but catastrophic exploit.

Keywords: Security risk analysis and management, Methods for dealing with incomplete or inconsistent information, Critical infrastructure protection.

1. Introduction

RAPSA/MC is a Monte-Carlo technique for capturing and expressing security risks in a computer network. Taylor et al. developed the Risk Analysis and Probabilistic Survivability Assessment (RAPSA) process [Taylor et al., 2002] which combines the quantification of Probability Risk Assessment (PRA) with Survivable Systems Analysis (SSA) [Ellison et al., 1999] to focus resources on the mission critical services of a system. The goal was to use SSA to tame the complexity of a large network and thereby permit quantification of critical threats. The RAPSA process captured each expert's estimate as an expected value, a single fixed quantity representing the average of the possibilities. But a security analyst needs a method to capture the uncertainty in an expert's estimate and to visualize its resulting impact on the model's forecasts; the Risk Analysis and Probabilistic Survivability Assessment Monte-Carlo (RAPSA/MC) process extends RAPSA with a Monte-Carlo simulation that provides that capability.

Decisions about new security technologies become investment decisions when, for example, an urgently needed intrusion detection system competes for resources with the other needs of an organization. The security analyst may be called to help position the value of the proposed investment with other contenders, to explain to the decision makers why this security investment is needed and to quantify the risks facing the organization if it is or is not implemented. Quantifying security risks as financial exposures simplifies their evaluation with the organization's other needs. Charting uncertainty in the forecast guides the financial decision makers to focus on what they do best, managing risks and opportunities.

2. Modeling Information Security Risks

Information security models often take one of two approaches, a low-level approach that models the detailed topology of a network and the specific vulnerabilities of its objects, or a systems-level approach that abstracts the details into high-level risks. A well-known example of the low-level approach includes the Take-Grant model [Lipton and Snyder, 1977] addressing the question as to whether a particular system with a given initial state is safe with respect to some specific access right [Bishop, 2003]. An advantage of the Take-Grant model is its ability to answer this important question in linear time.

As the size of a network increases, the application of low-level approaches becomes increasingly challenging for the analyst. In a complex enterprise, an analyst may become responsible for the security of hundreds or even thousands of network nodes, each with a myriad of unknown and known vulnerabilities. When human intervention is required to analyze the vulnerabilities of each node, even linear-time models may become unbearably expensive. To make matters worse, the security analyst's best intentions may be thwarted by an organization's limited budget that precludes a low-level detailed analysis of every vulnerability. Problems of this nature are often addressed with systems-level models.

Several systems-level security models express threats or opportunities as financial variables that quantify information security risks in terms of their financial impact. This approach can be traced back at least as far as the U.S. Government's pioneering FIPS guideline on information security in large data centers that proposed the use of the Average Loss Expectancy (ALE) metric [Soo Hoo, 2000]. Given a set of harmful outcomes, O_i, the frequencies of those outcomes, F_i, and the economic impacts of those outcomes, $I(O_i)$, the ALE metric is defined as:

$$ALE = \sum_{i=1}^{n} I(O_i)F_i \tag{1}$$

Soo Hoo cautions [Soo Hoo, 2000] that ALE, an *expected value*, equates high-probability but low-impact events with *extreme events* (those that though unlikely are catastrophic when they do occur). This critique can be important for financial decision makers because the reliance upon expected values obfuscates the consequences of a decision with the potential to kill the "cash cow on their watch." Financial decision makers should be made aware of which decisions ask them to "bet the farm." Haimes notes that the Partitioned Multi-objective Risk Method (PMRM) [Haimes, 1998] offers an alternative to the expected value approach and warns that when "...expected value is used as the sole risk measure, risk is likely to be grossly misinterpreted" with the potential for bad management decisions. Haimes continues, "...incorporating the risk of extreme events into the total risk management framework enhances the realism and the representation of risks." In short, the use of expected values alone misleads the financial decision makers with oversimplification. The analyst needs a straightforward mechanism to help the financial decision makers visualize the information security risks to the organization's assets and services.

A second critique of ALE is the potential for ALE-based models to auger into complexity [Soo Hoo, 2000] as they attempt to enumerate and address all known threats, assets and vulnerabilities in a large enterprise network. This path leads back to the pitfall of the low-level models: many organizations simply cannot afford to manually review every known detail of their operating network. This fundamental issue, how to analyze the security of a complex network within the constraints of a limited budget, likely underlies the motivation for many systems-level models.

Other approaches for quantifying security in systems-level models have been proposed. Geer urges [Geer, 2001] the use of Return-on-Investment (ROI) to guide information risk management decisions. Magnusson argues [Magnusson, 2005] for Net Present Value (NPV). Likewise, Butler champions the use of portfolio analysis methods [Butler et al., 1999] to guide software investment decisions. Schechter introduces a novel market-based approach [Schechter, 2004] to quantify the strength of a secured system as the market price to discover its next new vulnerability. Within this context, Schechter asserts that the strength of a system having a known vulnerability is negligible. Soo Hoo suggests a stochastic approach [Soo Hoo, 2000] to evaluate the role of uncertainty in information security decisions, and Conrad champions using Monte-Carlo methods [Conrad, 2005] for this endeavor.

3. Survivable Systems Analysis

Each of the above security risk models offers an approach to quantify a set of known threats to an information system. But how can we discover the threats to a complex system? Survivable Systems Analysis (SSA) defines a qualitative

methodology that focuses the search on just those threats to the most critical features [Software Engineering Institute, 2005] of an information system. The SSA method emerged from the study of Survivable Network Analysis [Ellison et al., 1999] at Carnegie Mellon. SSA offers several useful features:

- SSA focuses resources on the most critical functions of a software system found to be essential for its survival. This addresses Soo Hoo's second critique of ALE above.
- SSA methodology can be applied to an existing (already deployed) software system.
- SSA can also be applied to a new software system under development. Executable code is not required to support the SSA methodology.

The SSA methodology emphasizes survivability defined as a "system's capability to fulfill its mission (in a timely manner) in the presence of attacks, failures or accidents" and focuses on the "delivery of essential services and preservation of essential assets with timely recovery of full services and assets following the attack" [Ellison et al., 1999]. SSA approaches this goal by bolstering three characteristics of a survivable system: resistance, recognition and recovery. Resistance refers to the system's capability to repel an attack. Recognition refers to its ability to detect an attack in progress as well as assess the resulting damages. And recovery denotes the system's ability to fulfill its mission (albeit in a reduced capacity) during the attack, limit the extent of damages during the attack, and restore full service following the attack. The SSA methodology consists of four steps:

1 The System Definition step reviews both the system's responsibilities as defined in its Use-Cases as well as the components of the system's architecture.

2 The Essential Capability Definition step identifies those services and assets that are essential to the system's mission. This step identifies the essential components of the architecture that must survive an attack.

3 The Compromisable Capability Definition step enumerates the threats to the system and maps them onto the architecture's compromisable components, those whose security (confidentiality, integrity or availability) would be damaged by intrusion.

4 The Survivability Analysis identifies the soft-spot components of the architecture and constructs and analyzes a survivability map for opportunities to enhance the components' resistance, recognition and recovery characteristics.

4. Adding Monte-Carlo Simulation to RAPSA

Taylor et al. formed RAPSA by merging the SSA process with Probabilistic Risk Assessment (PRA) in order to simplify the quantification of security risks by focusing on just the mission-critical threats [Taylor et al., 2002]. The four steps of the RAPSA approach are:

1 The System Self-Assessment step identifies the mission objectives and partitions the system into essential and non-essential services.

2 The Threat Identification step enumerates the system's vulnerabilities, threats to those vulnerabilities and captures the attack stages for the essential services.

3 The Risk Quantification step quantifies the threats for each intrusion scenario and proposes mitigations. RAPSA considers the use of event or fault trees that are championed in some detail by threat modeling processes [Swiderski and Snyder, 2004].

4 The Risk Mitigation Trade-off constructs a survivability map augmented to quantify the mitigated risks and the costs of the proposed information security investments.

RAPSA addresses the expected value problem of risk analysis calculations by manually partitioning the random variable distributions into a few segments known as fractiles. However, the tedious nature of the calculations and the subjectivity of the partitioning process can frustrate its application in large, complex networks. The Monte-Carlo technique automates the partitioning and the manipulation of random variable distributions. A Monte-Carlo tool partitions the probability axis into hundreds or even thousands of fractiles, evaluating the model at each and collecting the resulting forecast. In each *iteration* of the simulation, the Monte-Carlo tool samples the chosen distributions and repeatedly executes the risk analysis model for each set of sampled values.

With the Monte-Carlo approach, RAPSA becomes RAPSA/MC and its application is very straightforward for the security analyst to apply. Because no executable code is required to support the methodology, RAPSA/MC can be applied before as well as after a system is placed in production. Thus, RAPSA/MC can be proactively incorporated into a software life-cycle to address security requirements during a system's development as opposed to waiting for the maintenance phase. The proactive application of security methodologies early in the life-cycle might lead to efficiency improvements over the reactive maintenance approach of securing a released product.

5. A RAPSA/MC Example

The data presented here are derived from Taylor et al. [Taylor et al., 2002] to enable comparison and validation with the RAPSA/MC approach. The example discusses a security assessment of the Supervisory Control And Data Acquisition (SCADA) equipment within a hypothetical electric utility's distribution substation. Because of the substation's unattended and often remote nature, considerable attention has been directed towards its physical security [Luo and Tu, 2005]. However, SCADA systems are increasingly networked [Brown, 2000] and are vulnerable to an electronic cyber attack that could be launched from a great distance with little risk to an intruder [Oman et al., 2002].

The electric power grid is a very large, complex real-time control system with hundreds of operators and thousands of real-time control actuators. High voltage optimizes efficient power transmission over long distance lines. However, local distribution requirements call for substantially lower voltages. An electrical substation often contains transformers to "step-down" high transmission voltage to lower voltage for local distribution as well as remotely accessed switch-gear, protection, phase-adjustment and metering equipment. Remote operation can be achieved through standard information system networking protocols augmented with protocols that are unique to the electric power industry [Woodward, 2001]. In some cases, the substation may even have an IP address or an 802.11 transceiver.

RAPSA/MC Step 1 leverages methodologies of RAPSA [Taylor et al., 2002] and SSA [Ellison et al., 1999] to identify the system's mission and partition its architecture, essential components and services into those that must survive an attack from those that are non-essential. The objective is to reduce the complexity of the security assessment by focusing the analyst's resources on the system's critical mission. Within the context of the electric power example, Taylor notes that the substation's mission is clear: deliver power to customers. While the details are omitted here for brevity, the analysis concludes that the example's essential services include remote monitoring and controls that are necessary for protection and changing loads.

RAPSA/MC Step 2 leverages from RAPSA and SSA to enumerate threats to the essential services and their supporting components. Table 1 identifies three example threats (drawn from Taylor [Taylor et al., 2002]) and their compromised essential (soft-spot) components. Note that example threats two and three use the same trojan exploit that becomes more disruptive in the hands of a knowledgeable attacker who uses the substation's SCADA equipment to attack the capital intensive assets of the utility's distribution system. Alternative methodologies for discovering these threats include Hierarchical Holographic Modeling [Haimes, 1998] [Longstaff et al., 2000] and Threat Modeling [Swiderski and Snyder, 2004].

Table 1. Example Threats and Compromised Components

Threats	*Compromised Components*
1. Hacker discovers the phone number of a modem on the substation's computer, successfully penetrates its authentication system and logs in.	Real-time control system device data altered/destroyed, devices reset, communication blocked or re-routed.
2. Utility employee is tricked into installing a trojan on a computer with access to the substation. The trojan installs a root-kit and "phones" home to the remotely located hacker.	Real-time control system device data altered/destroyed, devices reset, communication blocked or re-routed.
3. Similar to #2 only the attack is launched by a skilled attacker with power distribution knowledge who has deliberately targeted electric power utilities for sabotage.	Similar to the above plus damage to the electrical power distribution equipment.

RAPSA/MC Step 3 quantifies the threats. Like RAPSA, the RAPSA/MC process relies upon expert estimates for modeling parameters such as the number of intrusion events and the extent of damages. But the RAPSA/MC process captures the uncertainty of the experts' estimates with probability distributions (similar to Figure 1) in a Monte-Carlo simulation of the model's random variables.

Table 2 illustrates example values for each of the identified threats from Table 1 using hypothetical expert estimates to illustrate the process. All three threats compromise the SCADA equipment and could potentially lead to power outages (failure of the critical mission) or even damage to the electrical distribution equipment.

The variables $\{r_1, r_2, r_3\}$ capture the expected value of the expert's estimates for the annual intrusion rates. This particular example assumes that while the experts can predict the long-term intrusion rate, the year-to-year intrusion count is a random process. The example simulates the variability in the intrusion count in any given year using the Poisson random number generators in column three of Table 2.

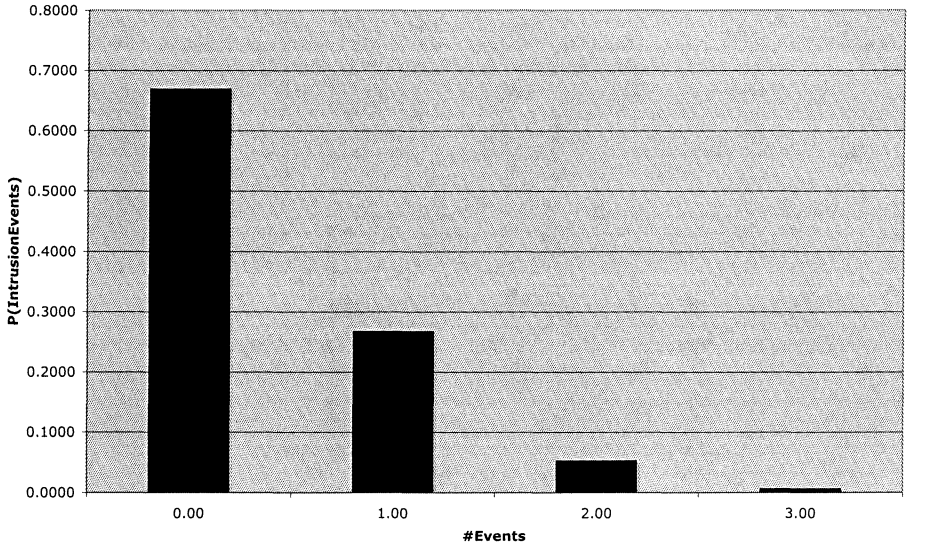

Figure 1. Probability of Intrusion Events for Threat-1

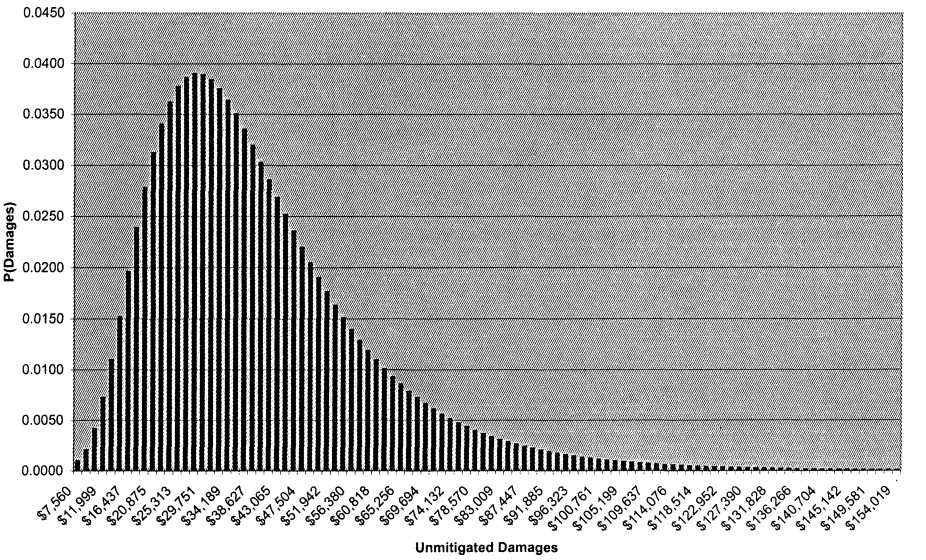

Figure 2. Prob. of Annual Unmitigated Damages from Threat-1

Table 2. Quantified Example Threats Against Unmitigated Targets

Threat	*Estimated Annual Intrusion Rate*	*Simulated Annual Intrusion Count*	*Estimated Unmitigated Damages*	*Estimated Stdev of Unmitigated Damages*	*Simulated Unmitigated Damages*	*Annual Unmitigated Damages*
1	$r_1 = 0.4$	$c_1 = \text{RPoisson}(r_1)$	$e_1 = \$100,000$	$s_1 = \$250,000$	$d_1 = \text{RLognormal}(e_1, s_1)$	$u_1 = c_1 * d_1$
2	$r_2 = 0.5$	$c_2 = \text{RPoisson}(r_2)$	$e_2 = \$150,000$	$s_2 = \$250,000$	$d_2 = \text{RLognormal}(e_2, s_2)$	$u_2 = c_2 * d_2$
3	$r_3 = 0.1$	$c_3 = \text{RPoisson}(r_3)$	$e_3 = \$1,000,000$	$s_3 = \$1,100,000$	$d_3 = \text{RLognormal}(e_3, s_3)$	$u_3 = c_3 * d_3$

Table 3. Example Survivability Map for Threat-1

Threat	*Mitigation*	*Resistance Strategy*	*Recognition Strategy*	*Recovery Strategy*
1	0	Current: Regular modem, weak password, no logging.	Current: Analysis of substation fault	Current: Locate damage, reset devices and restore from backups.
1	1	Dial-back modem, policy requires individual passwords and forbids single system password.	Logging. Weekly review of logs for intrusion evidence.	Reset devices and restore from backups.
1	2	Restrict dial-in user actions, restrict user access by time-of-day.	Logging + Intrusion Detection System (IDS) transmits alert	Reset devices and restore from backups.

The variables $\{e_1, e_2, e_3\}$ and $\{s_1, s_2, s_3\}$ capture the expert's estimates for the mean and standard deviation (stdev) of damages arising from successful unmitigated intrusions. The variables $\{d_1, d_2, d_3\}$ simulate the lognormal uncertainty in the expected damages arising from a single intrusion of each threat using the mean and extreme values of the previous two columns. Column seven calculates the annual damages in $\{u_1, u_2, u_3\}$ as the product of the simulated intrusion count $\{c_1, c_2, c_3\}$ and the simulated unmitigated damages $\{d_1, d_2, d_3\}$. Figure 2 illustrates the probability density function from a simulation of the annual unmitigated damages arising from Threat-1 (u_1).

Some details behind Table 2 deserve a closer examination.

- The choice of a Poisson distribution (e.g., Figure 1) to simulate the annual intrusion count reflects a desire to model unique successful intrusions (as opposed to clusters of identical unsuccessful attempts or even repeated successful exploits of the same target). In this particular example, the analyst has great confidence in the estimates for the $\{r_1, r_2, r_3\}$ rates and is concerned only with simulating the *variability* (randomness) of the Poisson process.

- The example's choice of the Lognormal distribution for damages (e.g., Figure 2) reflects a potential for extremely high (unbounded) damages, positively skewed with a minimum of zero and a strong mode below the mean. The $\{d_1, d_2, d_3\}$ variables simulate the expert's *uncertainty* (lack of knowledge) in the damage estimate.

- This example equates the expert's *uncertainty* (lack of knowledge) about the damages (e.g., Figure 2) with the *variability* of the successful intrusion events (e.g., Figure 1). When a distinction must be made between *uncertainty* and *variability*, alternative approaches [Vose, 2000] are available.

- RPoisson() and RLognormal() are representative of functions that are typically available in a Monte-Carlo tool for generating random numbers in Poisson and Lognormal distributions.

RAPSA/MC Step 4 summarizes the qualitative aspects of the proposed mitigations in a survivability map, quantifies the mitigations and finally selects an optimal investment. Table 3 presents the survivability map for just the first threat identified in Table 1. The mitigation strategies in Table 3 originated from RAPSA's application of SSA [Taylor et al., 2002].

Table 4. Quantified Mitigations for the Example Threats

Threat	*Mitigation*	*Estimated Minimal Effectiveness*	*Estimated Typical Effectiveness*	*Estimated Maximal Effectiveness*	*Simulated Effectiveness*	*Annual Mitigated Damages*
1	0	$me_{10} = 0.000$	$te_{10} = 0.000$	$xe_{10} = 0.000$	$se_{10} = 0.000$	$sd_{10} = u_1 - u_1 * se_{10}$
1	1	$me_{11} = 0.495$	$te_{11} = 0.550$	$xe_{11} = 0.605$	$se_{11} = \text{RTriangle}(me_{11}, te_{11}, xe_{11})$	$sd_{11} = u_1 - u_1 * se_{11}$
1	2	$me_{12} = 0.788$	$te_{12} = 0.875$	$xe_{12} = 0.963$	$se_{12} = \text{RTriangle}(me_{12}, te_{12}, xe_{12})$	$sd_{12} = u_1 - u_1 * se_{12}$
2	0	$me_{20} = 0.000$	$te_{20} = 0.000$	$xe_{20} = 0.000$	$se_{20} = 0.000$	$sd_{20} = u_2 - u_2 * se_{20}$
2	1	$me_{21} = 0.540$	$te_{21} = 0.600$	$xe_{21} = 0.660$	$se_{21} = \text{RTriangle}(me_{21}, te_{21}, xe_{21})$	$sd_{21} = u_2 - u_2 * se_{21}$
2	2	$me_{22} = 0.780$	$te_{22} = 0.867$	$xe_{22} = 0.953$	$se_{22} = \text{RTriangle}(me_{22}, te_{22}, xe_{22})$	$sd_{22} = u_2 - u_2 * se_{22}$
3	0	$me_{30} = 0.000$	$te_{30} = 0.000$	$xe_{30} = 0.000$	$se_{30} = 0.000$	$sd_{30} = u_3 - u_3 * se_{30}$
3	1	$me_{31} = 0.558$	$te_{31} = 0.620$	$xe_{31} = 0.682$	$se_{31} = \text{RTriangle}(me_{31}, te_{31}, xe_{31})$	$sd_{31} = u_3 - u_3 * se_{31}$
3	2	$me_{32} = 0.720$	$te_{32} = 0.800$	$xe_{32} = 0.880$	$se_{32} = \text{RTriangle}(me_{32}, te_{32}, xe_{32})$	$sd_{32} = u_3 - u_3 * se_{32}$

Table 5. Forecasted Total Annual Costs

Mitigation	*Annual Mitigated Damages*	*Estimated Annual Investment Expenses*	*Total Annual Cost*	*Forecasted Mean Cost (From Simulation)*	*Forecasted 90th Percentile Cost (From Simulation)*
0	$amd_0 = sd_{10} + sd_{20} + sd_{30}$	$ie_0 = \$0$	$tac_0 = amd_0 + ie_0$	\$64,000	\$160,000
1	$amd_1 = sd_{11} + sd_{21} + sd_{31}$	$ie_1 = \$6,000$	$tac_1 = amd_1 + ie_1$	\$38,000	\$87,000
2	$amd_2 = sd_{12} + sd_{22} + sd_{32}$	$ie_2 = \$20,000$	$tac_2 = amd_2 + ie_2$	\$49,000	\$94,000

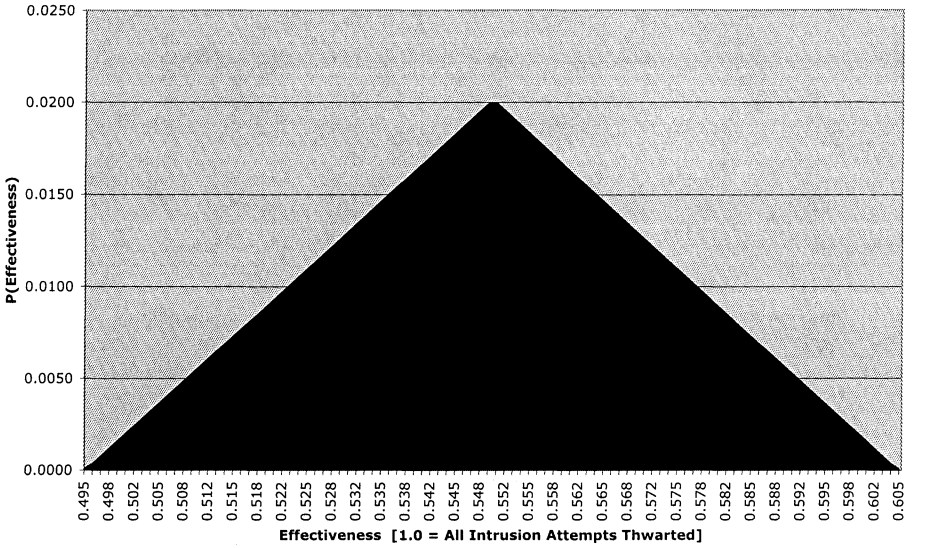

Figure 3. Effectiveness of Mitigation-1 Against Threat-1

Table 4 introduces the quantitative behaviors of the three mitigations discussed above. Each row of the table addresses one proposed mitigation for a single threat. Estimates are solicited from the experts for the minimum effectiveness (me_{ij}), the typical effectiveness (te_{ij}) and the maximum effectiveness (xe_{ij}) for each mitigation (j) of each threat (i). The effectiveness estimates define the fraction of intrusion attempts of a particular threat that will be deterred by this mitigation. In each iteration (year) of the simulation, these three effectiveness estimates are used as parameters in column six to calculate a simulated (random) effectiveness value using a rough triangle distribution (e.g. Figure 3). Column seven calculates the simulated annual mitigated damages for each iteration.

Table 5 forecasts the Total Annual Costs (tac_j) arising from both the annual mitigated damages (amd_j) and the annual investment expenses (ie_i). The total annual costs are calculated as the sum of the annual mitigated damages and the annual investment expense for each mitigation.

6. Analyzing the Example RAPSA/MC Simulation

The RAPSA/MC example illustrates a tremendous potential for uncertainty in the unmitigated damage estimates. It highlights the expert's struggle to estimate the potential damages that might be inflicted by the casual intrusion of

Threat-1 (Figure 2). In this case, the expert selected a lognormal distribution with a mean value at $40,000 and the 90th percentile at $66,000, there exists a remote possibility for very much larger damages. The uncertainty in this estimate reflects the hypothetical expert's lack of knowledge as to whether a "casual" intruder's damages will be confined to the information systems, extend to power outages, or further extend to the power distribution equipment. The RAPSA/MC process captures this uncertainty rather than oversimplify it into an *expected value*.

The Monte-Carlo tool runs the simulation by executing the model through hundreds or even thousands of iterations, each modeling the events of one potential year. It begins by "throwing" random numbers into the distributions (c_i, d_i and se_{ij}) and recalculating the forecasted Total Annual Costs (tac_j) and capturing those results. The example employs 10,000 iterations with Latin Hypercube Sampling to simulate the variability and uncertainty in the random variables. Following the simulation run, the Monte-Carlo tool charts the captured results as forecasts (Figures 4, 5 and 6) for each mitigation. Please note that the very large P($0), P($6,000) and P($20,000) columns are omitted from the left-hand side of the three forecast charts. This is done to facilitate readability of the distributions as the probability of no damages (just mitigation investment expenses) in this particular example is quite high (about 0.38 which is nearly an order of magnitude larger than any other single column).

Table 5 also documents the forecasted mean and 90th percentile values for the three proposed mitigations in Figures 4, 5 and 6. These and other statistics are available from the Monte-Carlo tool following a simulation run.

The forecast chart for Mitigation-0 (Figure 4) illustrates the risks of doing nothing (no investment). The mean annual loss due to damages is $64,000 and there exists a 10% probability of an annual loss in excess of $160,000.

When the annual investment expenses are included as they are in this version of the example, the decision makers are likely to choose Mitigation-1. The statistics for Mitigation-1 suggests that an annual investment of $6,000 will limit the mean costs to $38,000 but there still remains a 10% exposure to annual costs exceeding $87,000.

We might contemplate alternative scenarios that would lead to a different investment choice. If, for example, the investment expenses for Mitigation-2 ($20,000) could be reduced, it would become far more attractive. Or if the 90th percentile figure for Mitigation-1 had been extremely high (say... $500,000 vs $87,000) then the decision makers might choose Mitigation-2 simply to buy down their exposure to an extreme event. Likewise, if the investment costs had been significantly higher for both Mitigation-1 and 2, then the decision makers might choose to do nothing at all (e.g. choose Mitigation-0) unless the 90th percentile figures were extremely high for doing nothing in which case they might again choose a "real" mitigation investment to buy down their

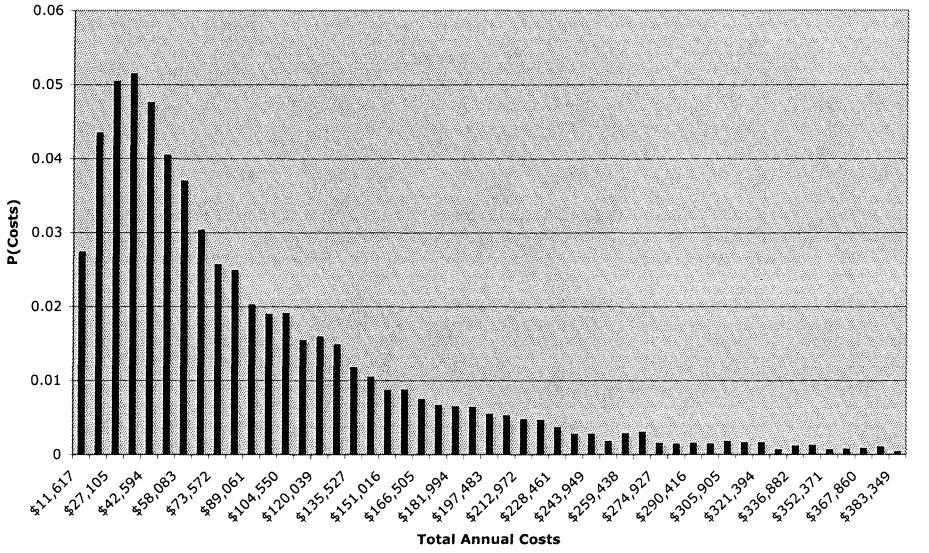

Figure 4. Forecasted Annual Costs for Proposed Mitigation-0

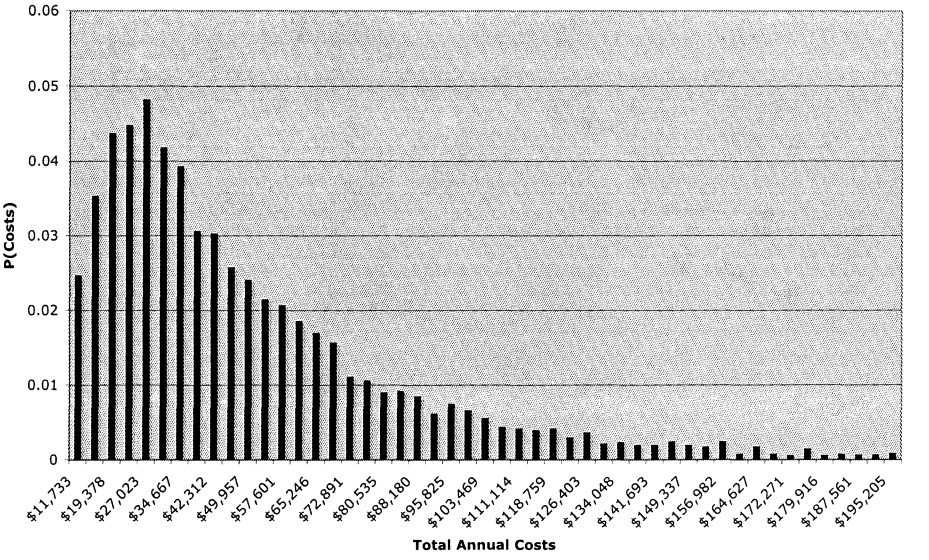

Figure 5. Forecasted Annual Costs for Proposed Mitigation-1

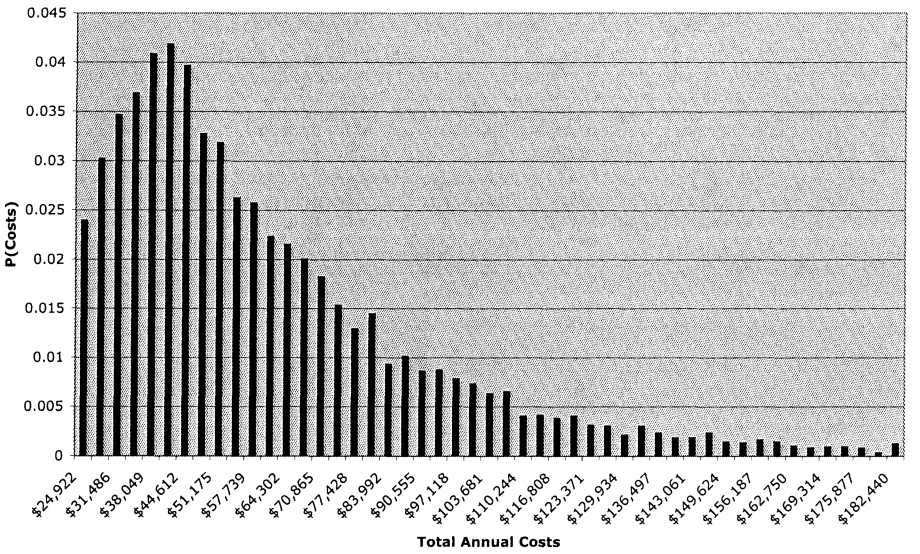

Figure 6. Forecasted Annual Costs for Proposed Mitigation-2

exposure to an extreme event. Tradeoffs between an attractive expected return and catastrophic risk potential are common in financial decisions.

7. Conclusions

RAPSA/MC quantifies information security risks as financial variables facilitating the comparison of security mitigations with an organization's other opportunities. RAPSA/MC combines qualitative methodologies for identifying threats with Monte-Carlo quantitative methodologies for simulating uncertainty in security parameters. While the Monte-Carlo technique cannot alone break the dependency of the systems-level models on expert estimates, it does enable these models to express uncertainty in their forecasts. The RAPSA/MC forecasts are probability density functions (Figures 4, 5 and 6) that provide financial decision makers with the opportunity to consider the potential for extreme events as well as the mean value of the proposed mitigation. Even though a proposed mitigation might offer the optimal mean value, the decision makers may choose an alternative that offers a lower risk of a catastrophic extreme event. The process is usable both for evaluating released systems and systems under development as no executable code is required for the risk assessment.

References

Bishop, Matt (2003). *Computer Security: Art and Science*. Addison-Wesley, Boston, MA.

Brown, Steven M. (2000). Applying internet technology to utility scada systems. *Utility Automation*, 5(5):25–26.

Butler, S., Chalasani, P., Jha, S., Raz, O., and Shaw, M. (1999). The potential of portfolio analysis in guiding software decisions. First Workshop on Economics-Driven Software Engineering Research.

Conrad, James R. (2005). Analyzing the risks of security investments with monte-carlo simulations. In *Fourth Workshop on the Economics of Information Security (WEIS05)*, Harvard University (USA).

Ellison, Robert J., Linger, Richard C., Longstaff, Thomas, and Mead, Nancy R. (1999). Survivable network system analysis: A case study. *IEEE Software*, 16(4):70–77.

Geer, Daniel E. (2001). Making choices to show ROI. *Secure Business Quarterly*, 1(2).

Haimes, Yacov Y. (1998). *Risk Modeling, Assessment, and Management*. John Wiley and Sons, New York, NY.

Lipton, R. J. and Snyder, L. (1977). A linear time algorithm for deciding subject security. *J. ACM*, 24(3):455–464.

Longstaff, Thomas A., Chittister, Clyde, Pethia, Rich, and Haimes, Yacov Y. (2000). Are we forgetting the risks of information technology? *IEEE Computer*, 33(12):43–51.

Luo, Yi and Tu, Guangyu (2005). Who's watching the unattended substation. *IEEE Power and Energy Magazine*, 3(1):59–66.

Magnusson, Christer (2005). Shareholder value and security investments. *IEEE Communications Magazine*, 43(1):3–4.

Oman, Paul, Schweitzer III, Edmund O., and Frincke, Deborah (2002). Concerns about intrusions into remotely accessible substation controllers and scada systems. In *Proc. 27th Annual Western Protective Relay Conferences*.

Schechter, Stuart Edward (2004). *Computer Security Strength and Risk: A Quantitative Approach*. PhD thesis, Harvard University, Cambridge, Massachusetts.

Software Engineering Institute (2005). Survivable systems analysis.

Soo Hoo, Kevin J. (2000). How much is enough? A risk-management approach to computer security. Technical report, Stanford Consortium for Research on Information Security and Policy.

Swiderski, Frank and Snyder, Window (2004). *Threat Modeling*. Microsoft Press, Redmond, WA.

Taylor, Carol, Krings, Axel, and Alves-Foss, Jim (2002). Risk analysis and probabilistic survivability assessment (RAPSA): An assessment approach for power substation hardening. In *ACM Workshop on the Scientific Aspects of Cyber Terrorism*, Washington, D.C. ACM.

Vose, David (2000). *Risk Analysis: A Quantitative Guide*. John Wiley and Sons, West Sussex, England, 2nd edition.

Woodward, D. (2001). The hows and whys of ethernet networks in substations. Technical report, Schweitzer Engineering Labs.