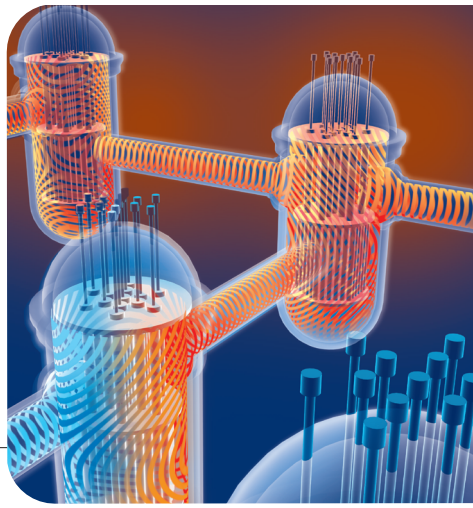
Security Meter: A Practical Decision-Tree Model to Quantify Risk

Several security risk templates employ nonquantitative attributes to express a risk's severity, which is subjective and void of actual figures. The author's design provides a quantitative technique with an updated repository on vulnerabilities, threats, and countermeasures to calculate risk.



Mehmet Sahinoglu
Troy University

As part of my research to quantify risk in security risk assessment, I've devised and proposed Security Meter, a model that provides a purely quantitative and semiquantitative (hybrid) alternative to frequently used qualitative models,[1] such as Symantec's Enterprise Security Architecture (www.symantec.com). The proposed approach is a quick, bird's-eye-view way of calculating a system's information security risk (http://socrates.tsum.du/~mesa).

In this article, I also propose a modification of some of the decision-tree-based model's qualitative attributes, in case the quantitative data are unavailable. The proposed model is practical and simple to use for beginners in the field, but it also provides a mathematical–statistical foundation on which strategists or practitioners can construct a practical risk valuation. The probabilistic assumptions, such as using a uniformly distributed random variable for the input variables, can be improved by using other statistical distributions. Other techniques used hitherto within a nonprobabilistic frame, such as attack trees, don't provide an accurate overall picture of the risk to the system that's being protected.[2–4]

Risk scenarios

Conventionally, risk scenarios involve possible chance-based catastrophic failures with scarce modeling of maliciously designed human interventions that threaten inherent system vulnerabilities. Risk scenarios concerning critical computer communication networks are now more pervasive and severe than ever before because of the cost of nonmalicious chance failures that occur due to insufficient testing and lack of adequate reliability. We can use software reliability modeling and testing techniques to examine these chance failures in more detail.[5–8] However, for the intentional failures or malicious activities that critically increase the risk of ill-defined attacks, no one has ever thoroughly modeled a physical scenario, at least not one that considers a unified consistent scheme of vulnerabilities, threats, and countermeasures. A quantitative risk assessment provides results in numbers that management can understand, whereas a qualitative approach, although easier to implement, makes it difficult to trace generalized results. My proposed security-meter design fills a void in the arena of much-sought quantitative risk evaluation favorably compared to most current assessments that provide qualitative results. This is achieved by a probabilistically accurate quantitative model that measures security risk. The design's concrete numerical approach, which always works for all systems, can further facilitate security risk management and security testing. This means that the final risk measure calculated as a percentage can be tested, improved, compared, and budgeted as opposed to attributes such as high, medium, or low, which cannot be managed or quantified numerically for an objective assessment.[9]

Banks and other financial institutions, for example, employ several commercially available security risk templates, mostly in verbal or qualitative form, that express the severity of a risk by classifying them as low, medium, or high. This approach is not only highly subjective, but it also lacks any actual risk figures. Quantitative risk figures help mitigate or avoid future errors by allowing risk managers to objectively compare project alternatives and identify priorities for software maintenance. In existing analyses that favor a quantitative study, either a probabilis-

tic frame about whether to add or multiply risks doesn't exist, or the risk calculations are handled on a case-by-case basis without a network-oriented conclusion.[1]

Without using a probabilistic framework such as the one suggested in my Security Meter design, the conclusions to assess a risk's severity might be misleading and costly due to over- or underestimation, especially during military conflicts and wars, where risk scenarios are often underestimated. My design could be useful not only for commercial companies and military or government entities whose job it is to run daily risk assessments, but also for regular end users such as anyone who uses a household PC to send email. Much statistical planning and design remains to be done to reach a point at which end users of any skill level will have a consistently updated repository of vulnerability, threat, and countermeasure.

A quantitative security-meter model

Let's look more closely at my model. It includes a description of the input and output in a probabilistic decision-tree diagram approach. Further, the same principles will be applied to those cases within a modified approach of the proposed method where all quantitative data are invariably not available for the input parameters.

Risk management

Risk management is the total process of identifying, measuring, and minimizing the uncertain events that can affect resources. This definition also implies the process of bringing management (remedial action) and control into the risk analysis. A basic ingredient of risk assessment and analysis is the concept of vulnerability. A *vulnerability* is a weakness in any information system, system security procedure, internal controls, or implementation that an attacker could exploit. It can also be a weakness in a system, such as a coding bug or design flaw. An attack occurs when an attacker with a reason to strike takes advantage of a vulnerability to threaten an asset. The second most important ingredient in risk assessment is the concept of a threat, which is any circumstance or event with the potential to adversely impact an information system through unauthorized access, destruction, disclosure, modification of data, or denial of service. Similarly, a threat to a system is a potential event that will have an unwelcome consequence if it becomes an attack asset.[10, 11]

We can define risk as the possibility that a particular threat will adversely impact an information system by exploiting a particular vulnerability. The third ingredient in the risk analysis is the *countermeasure* (CM); or lack thereof. A CM is an action, device, procedure, technique, or other measure that reduces risk to an information system. Consequently, the residual risk is the portion of risk remaining after a CM is applied. Residual risk could be "none" if a perfect CM exists. My proposed physical model identifies the deterministic (constant) and probabilistic (random) inputs for the target output of a residual risk—namely, an attack as well as the expected cost to avoid or mitigate the calculated risk.

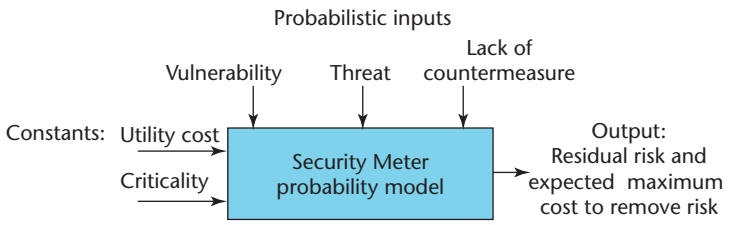


Figure 1. The quantitative security-meter probability model to contain probabilistic and deterministic inputs. The blue box is the probabilistic tree-diagram that performs the operations.

Probabilistic inputs

The suggested vulnerability values range between 0 to 1.0 (or 0 to 100 percent), adding up to one. In a probabilistic sample space of feasible outcomes of "vulnerability," the sum of probabilities adds up to one. This is like the probabilities of the faces of a die, such as 1 to 6, totaling to one, whether the die is fair or tilted. If a cited vulnerability isn't exploited in reality, then it can't be included in the model or Monte Carlo simulation study (which we'll examine in more detail later). Each vulnerability has from one to several threats. A *threat* is defined as the probability of the exploitation of some vulnerability or weakness within a specific timeframe. Undesirable threats that take advantage of hardware and software weaknesses or vulnerabilities can impact the violation and breakdown of availability, integrity, confidentiality, and nonrepudiation as well as other aspects of software security such as authentication, privacy, and encryption.[1] Each threat has a CM that ranges between 0 and 1 (with respect to the first law of probability) whose complement gives the lack of CM (LCM). The binary CM and LCM values should add up to one, keeping in mind the second law of probability. The security risk analyst can define, for instance, a network server (v_1) as a vulnerability located in a remote, unoccupied room in which a threat (t_{11}), such as individuals without proper access or a fire (t_{12}), could result in the destruction of assets if not countermeasured by items such as a motion sensor (CM_{111}) or a fire alarm (CM_{121}), respectively.

Deterministic inputs

System criticality, another constant that indicates the degree of how critical or disruptive the system is in the event of entire loss, is taken to be a single value corresponding to all vulnerabilities with a value ranging from 0 to 1. Criticality is low if residual risk is of little or no significance, such as the malfunctioning of an office printer, but in the

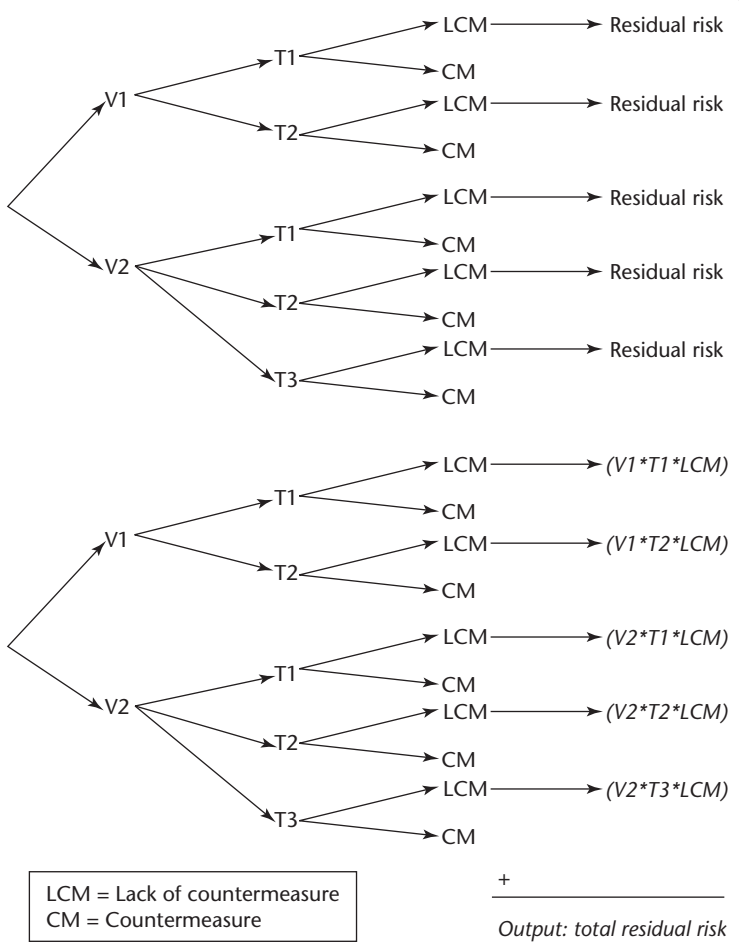


Figure 2. A general-purpose decision-tree diagram example for the Security Meter model.

case of a nuclear power plant, criticality is close to 100 percent, because its security has vital safety ramifications for humans. Capital (investment) cost is the total expected loss in monetary units (dollars) for the particular system if it is completely destroyed and can't be utilized anymore, had the system continued to generate added value for other parts of the system or society.

Decision-tree diagram

Given that a simple sample system or component has two or more outcomes of each risk factor, vulnerability, threat, and CM, the following probabilistic frame holds. For the sums of $\sum v_i = 1$ and $\sum t_{ij} = 1$ for each i, and the sum of LCM + CM = 1 for each ij, within the tree-diagram structure shown in Figure 2. Using the probabilistic inputs, we get the residual risk:

$$\text{Residual risk} = \text{Vulnerability} * \text{Threat} * \text{LCM} \quad (1)$$

We can calculate the residual risks for all vulnerabilities with threats and LCMs, as well as the total residual risk when added. That is, if we add all the residual risks due to lack of CMs as in Figure 2, we can find the overall residual risk. We apply the criticality factor on the residual risk to calculate the final risk. Then, we apply the capital investment cost on the final risk to determine the expected cost of loss (ECL), which helps budget avoid (before the attack) or repair (after the attack) the entire risk:

$$\text{Final risk} = \text{Residual risk} * \text{Criticality} \quad (2)$$

and

$$\text{ECL} = \text{Final risk} * \text{Capital cost}. \quad (3)$$

A model application and results

A risk analyst conducts a Monte Carlo simulation to mimic the relationship between vulnerabilities, threats, and CMs as they occur in real life. That is, a certain vulnerability is threatened by a threat, and therefore becomes an attack at the next level if said threat isn't countermeasured by a firewall in a computer or motion sensor in the case of intrusion, or fire alarm in the case of fire, according to what the actual situation could present. If fully countermeasured (that is, CM = 1 or LCM = 0), then no attack occurs, as Equation 1 shows: where the residual risk is zero because one of the factors is zero. Equation 2 further determines the ECL (posthumously if no action is taken) or expected cost of repair to avoid the entire risk (if action is taken proactively). Risk analysis has various inputs, such as vulnerability types and each threat's CM; criticality and capital cost are constants as are the number of simulation runs. From these input values, we determine the expected maximum monetary loss to mitigate the residual risk. To represent each risk factor, such as vulnerability (v_i), threat (t_{ij}), and CM (CM_{ijk}), with an educated guess, we can assume the uniform (or rectangular) density parameters, which take on values between a lower limit "a" and an upper limit "b." Table 1 shows the lower and upper limits for all risk factors. The average or expected value of a uniformly distributed random variable is $\mu = (a + b)/2$; that is, the expected values when placed in the decision-tree diagram as in Figure 2 and Table 2 will result in an expected output, which the Monte Carlo simulation verifies, conducting thousands of runs to converge to the expected output.

Let's examine a sample application whose input data, which Table 1 shows, revolves around a home office PC. In this hypothetical example, we assume that five reportedly recognized types of vulnerabilities (v_1 to v_5) exist, whose pertaining threats for each vulnerability might be either two- or threefold. Again, for each threat, there is a CM or LCM, whose probabilities complement to 1.

Table 1. Probabilistic input data for vulnerabilities, threats, and countermeasures in a hypothetical home PC office.

VULNERABILITY	THREAT	LACK OF COUNTERMEASURE
1) $v1(a = 0.1, b = 0.3)$, $\mu_{v1}=0.2$	1) $t1(a = 0.1, b = 0.6)$, $\mu_{t1}= .35$	$LCM_1(a = 0.1, b = 0.5)$, $\mu_{LCM1}=.3$; $\mu_{CM1}=.7$, by subtraction
	2) $t2$ by subtraction, $\mu_{t2}=.65$	$LCM_2(a = 0.2, b = 0.6)$, $\mu_{LCM1}=.4$; $\mu_{CM2}=.6$, by subtraction
2) $v2(a = 0.0, b = 0.4)$, $\mu_{v2}=0.2$	1) $t1(a = 0.2, b = 0.6)$, $\mu_{t1}= .40$	$LCM_1(a = 0.1, b = 0.7)$, $\mu_{LCM1}=.4$; $\mu_{CM1}=.6$, by subtraction
	2) $t2(a = 0.1, b = 0.3)$, $\mu_{t2}= .20$	$LCM_2(a = 0.0, b = 0.2)$, $\mu_{LCM1}=.1$; $\mu_{CM2}=.9$, by subtraction
	3) $t3$ by subtraction $\mu_{t3}= .40$	$LCM_3(a = 0.1, b = 0.4)$, $\mu_{LCM1}=.25$; $\mu_{CM3}=.75$, by subtraction
3) $v3(a = 0.1, b = 0.2)$, $\mu_{v3}=0.1$	1) $t1(a = 0.1, b = 0.5)$, $\mu_{t1}= .30$	$LCM_1(a = 0.1, b= 0.4)$, $\mu_{LCM1} =.25$; $\mu_{CM1} =.75$, by subtraction
	2) $t2$ by subtraction, $\mu_{t2}=.70$	$LCM_2(a= 0.0, b = 0.3)$, $\mu_{LCM1}=.15$; $\mu_{CM2}=.85$, by subtraction
4) $v4(a = 0.0, b = 0.1)$, $\mu_{v4}=0.05$	1) $t1(a = 0.1, b = 0.4)$, $\mu_{t1}= .25$	$LCM_1(a= 0.1, b = 0.4)$, $\mu_{LCM1}=.25$; $\mu_{CM1}=.75$, by subtraction
	2) $t2(a = 0.0, b = 0.5)$, $_{t2} = .25$	$LCM_2(a = 0.2, b = 0.6)$, $\mu_{LCM2} = .4$; $_{CM2} = .60$, by subtraction
	3) $t3$ by subtraction, $\mu_{t3} = .50$	$LCM_3(a = 0.2, b = 0.6)$, $\mu_{LCM3} = .4$; $\mu_{CM3} = .60$, by subtraction
5) $v5$ by subtraction, $\mu_{v5} =.45$	1) $t1(a = 0.1, b = 0.5)$, $\mu_{t1}= .30$	$LCM_1(a = 0.1, b = 0.3)$, $\mu_{LCM1}=.2$; $\mu_{CM1}=.80$, by subtraction
	2) $t2$ by subtraction $\mu_{t2} = .70$	$LCM_2(a = 0., b = 0.3)$, $\mu_{LCM2}=.15$; $\mu_{CM2} =.85$, by subtraction

Table 2 shows the expected values of the input random variables given in Table 1 to produce the expected theoretical output. Residual risk is what is left of the risk (vulnerability * threat) after we apply the CM to circumvent the risk; that is, residual risk = risk * LCM. If we have perfect CM (CM = 1), then the LCM is null (LCM = 1 – CM), which results in zero residual risk. The residual risks corresponding to each threat of a given vulnerability are added to find the total residual risk resulting from the entire set of vulnerabilities, and their attached threats, either countermeasured or not, by a preventive measure.

Thus, the final risk, using Equation 2, is when the resulting total residual risk is multiplied by the criticality factor. As explained earlier, if the criticality is zero, then there's no final risk. If equipment is critical for a job, school, or nation, such as in the case of a nuclear plant, then you attach a high criticality factor, such as 1, to it. In the home PC example, when we use Equation 2 employing a criticality factor of 0.4, we find a final risk of 0.0975 or 9.75 percent. Now using Equation 3 and employing a sample invested capital cost of $2,500, the ECL due to final risk is calculated to be $239.38. Therefore,

Final risk = Residual risk * Criticality = 0.239375 * 0.4 = 0.0975.

ECL = Final risk * Capital cost = 0.0975 * $2,500 = $239.38.

The Monte Carlo simulation produces 0.2393777 (in Figure 5) versus an expected result of 0.2393775 (in Table 2), where the difference can only be described as negligible. For ECL, the simulation generates $239.38, identical to the calculated ECL result of Equation 3 after conducting 10,000 * 5,000 = 50 million simulation runs in a little over five minutes. The purpose for the simulation (explained later) is to mimic the actual operation and verify the theoretical results.

Modifying the quantitative model for qualitative data

In the event that we don't possess purely quantitative values for each attribute in the decision-tree diagram in Figure 2, and all we have are qualitative adjectives such as H (high; often), M (medium; seldom), or L(low; rare), we must modify our approach (Figure 3). We can then use

Table 2. Expected (theoretical) results for application 1 in Table 1.

VULNERABILITY	THREAT	LACK OF COUNTERMEASURE	RESIDUAL RISK
0.2	0.35	0.3	**0.021**
		0.7	
	0.65	0.4	**0.052**
		0.6	
0.2	0.4	0.4	**0.032**
		0.6	
	0.2	0.1	**0.004**
		0.9	
	0.4	0.25	**0.02**
		0.75	
0.1	0.3	0.25	**0.075**
		0.75	
	0.7	0.15	**0.105**
		0.85	
0.05	0.25	0.25	**0.3125**
		0.75	
	0.25	0.4	**0.005**
		0.6	
	0.5	0.4	**0.01**
		0.6	
0.45	0.3	0.2	**0.027**
		0.8	
	0.7	0.15	**0.04725**
		0.85	

TOTAL RESIDUAL RISK = 0.239375 OR 23.94%

the probabilities of H, M, and L—that is, P(H), P(M) and P(L)—as long as the addition rule of unity holds for disjoint events (second and third laws of probability). Such outcomes of vulnerability (the first branch of the tree diagram) or threat (the second branch) can include H + L = 1, or M + L + L = 1 or L + L + L + L = 1, where H = 0.75, M = 0.5 and L = 0.25 hold. Another feasible possibility is when H + L = 1, or M + L + L + L = 1, or 5L = 1 to signify five outcomes at most for either vulnerability or threat variable, where H = 0.8, M = 0.4, and L = 0.2 hold. For up to five vulnerabilities, H + M = 1, or M + L + L + L = 1, or 5L = 1, then H = 0.6, M = 0.4, and L = 0.2. If, for example, H = 0.75, M = 0.5, and L = 0.25, as Figure 3 shows, there's a 37.5 percent risk of losing the facility's or system's availability.

A hybrid security-meter model

If we don't possess purely quantitative data, and all we have is a hybrid of quantitative risk values (namely, probabilities between 0 and 1) and qualitative attributes such as H, M, or L, then the model of Figure 3 will transform to Figure 4's hybrid model. There will be some branches with letters out of uncertainty (represented by probabilities according to the fundamental laws of probability) and some quantitative probability values obtained from the previous data monitoring.

We can combine these inputs as long as the fundamental laws of probability hold. This necessity arises when the risk analyst is unsure about the risk values but can only identify certain quantitative risks combined with uncertain adjectives as too high, medium, or low. For example, we might have H + .25 = 1, or M +.25 + .25 = 1, or .3 + .2 + L + L = 1, or 4L=1, where H = .75, M = 0.5, L = .25 hold.

In Figure 4, for some branches, the risks (probabilities) are known, and in others, H, M, or L are given as long as the fundamental laws of probability hold. Figure 4 shows that the values H = 0.75 and L = 0.25 hold true when M is not used, and the total risk is 25.9 percent. As for the qualitative or hybrid model, there might be limitations to the denominations of vulnerabilities or threats according to the choice of estimated values for H, M, and L to reflect the best hypothesis. The analyst sometimes might have to go an extra step and choose H, M, L, and W (very low). For example, in the case in which 8W = 1, M + 3L = 1 and H + 2W = 1, to imply that H = 0.75, M = 0.40, L = 0.2 and

W = 0.l25, at most eight possible outcomes (denominations) of the vulnerability or threat variable could exist.

The study

A Monte Carlo simulation study verifies my model's mathematical accuracy. Five-thousand runs, one of which is displayed in Figure 5 for illustration purposes, are conducted by generating random variables from each vulnerability, threat, or CM. Then, the security-meter method multiplies each branch's conditional probabilities with respect to Equation 1 and Figures 2 to 4 to calculate the residual risks and total them to show residual risk. The average of a selected total number of runs such as 5 million or 50 million will converge to be the final result. Then Equations 2 and 3 can be used to reach the final risk and cost.

Figure 5 displays the input data for $v(a, b)$, $t(a, b)$, and CM(a, b), which are uniformly distributed, $U(a, b)$. The lower and upper bound values for the last windows in the case of fifth vulnerability or second or third threats are left blank as the software will complement it to 1 to obey the fundamental probability law. Otherwise, it will refuse the random deviate to seek a new one. The budgetary portfolio at the end of such quantitative analyses is an asset. In this hypothetical or educational example, $239.38 is needed to proactively avoid or repair after the fact. Figure 5 shows the final MC simulation result for the final risk and monetary extent of the physical damage.

I plan to work with companies and state agencies to bring a probabilistically sound framework from a design stage into an application stage. I envision a product that might result from this project for company or end-user benefit. Security is a process, not a product, but we still need accurate and reliable products to calculate security quantitatively to improve security. □

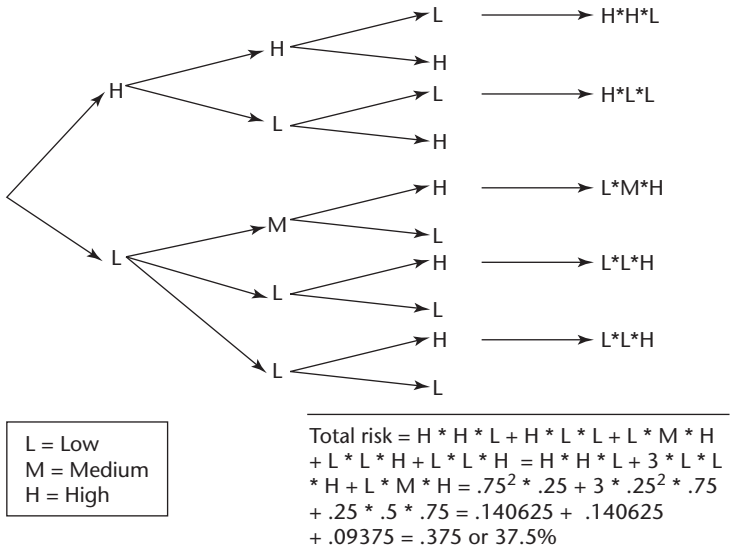


Figure 3. A modified purely qualitative decision-tree diagram on the Security Meter model. Each branch now has letters to signify certain quantities for the vulnerability, threat , and countermeasure (CM) where the three fundamental laws of probability hold.

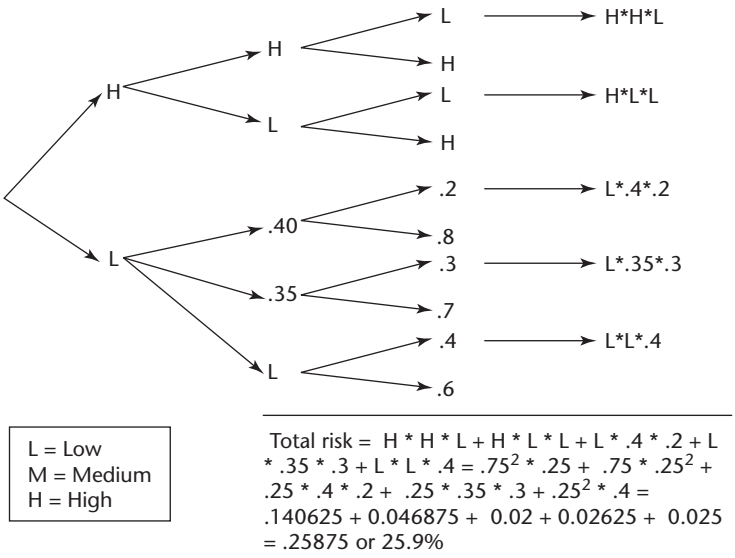


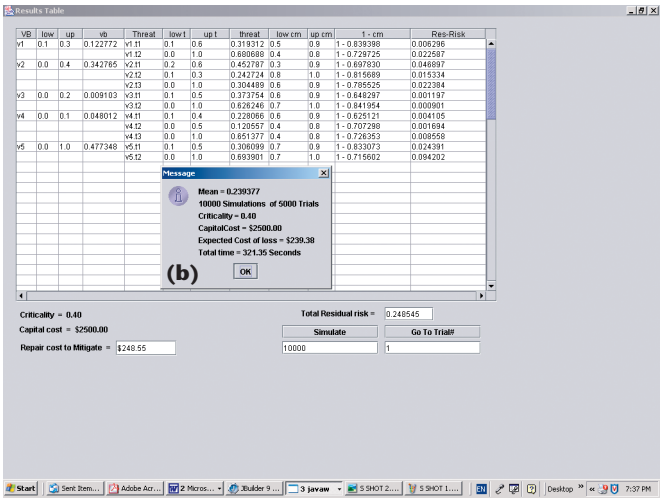
Figure 4. A modified hybrid decision-tree diagram on the Security Meter model. This differs from Figure 3 in that the branches can carry qualitative attributes and quantitative values as well for the same vulnerability or threat variable.

Acknowledgments

I thank Justin Larson from Troy University-Montgomery's Department of Computer Science for his programming contributions.

References

1. E. Forni, "Certification and Accreditation," AUM Lecture Notes, Data Systems Design Laboratories, 2002; www.dsdlabs.com/security.htm.
2. B. Schneier, *Applied Cryptography*, 2nd ed., John Wiley & Sons, 1995
3. K.A. Forcht, *Computer Security Management*, Boyd and Fraser, 1994.
4. *Integrated Research in Risk Analysis and Decision Making in a Democratic Society*, Workshop Report, US Nat'l Science Foundation, 2002, www.nsf.gov/pubs/nsf03209/start.htm.
5. M. Sahinoglu and E.H. Spafford, "A Bayes Sequential Statistical Procedure for Approving Products in Mutation-Based Software Testing," *Proc. IFIP Conf. Approving Software Products* (ASP 90), W. Ehrenberger, ed., Elsevier Science Publishers, 1990, pp. 43–56.
6. M. Sahinoglu, J.J. Deely, and S. Capar, "Stochastic Bayes Measures to Compare Forecast Accuracy of Software-



(a)

Figure 5. (a) The screenshot of the residual risk of the simulation run $n = 1$ out of a total of the first 5,000 runs gives a result of 0.248545 for a single run. Although different than the theoretical solution, this will converge to the expected result after many such runs. The background screenshot displays the calculating mechanics of the Security Meter technique with the probabilistic and constant inputs and risk output in a concise summary spreadsheet. (b) The screenshot for the averaged residual risk of 0.239377 after conducting 50 million (10,000 times 5,000 runs each) most satisfactorily compared to the expected total residual risk of 0.239375 (as Table 2 shows).

Reliability Models," *IEEE Trans. Reliability*, vol. 50, no. 1, 2001, pp. 92–97.

7. M. Sahinoglu, "An Empirical Bayesian Stopping Rule in Testing and Verification of Behavioral Models," *IEEE Trans. Instrumentation and Measurement*, vol. 52, no. 5, 2003, pp. 1428–1443.
8. M. Sahinoglu et al., "Measuring Availability Indices with Small Samples for Component and Network Reliability using the Sahinoglu-Libby Probability Model," to appear in *IEEE Trans. Instrumentation and Measurement*, vol. 54, no. 3, June 2005.
9. M. Sahinoglu, "Security Meter: A Probabilistic Framework to Quantify Security Risk," Certificate of Registration, US Copyright Office, TXu 1-134-116, 5 Dec. 2003.
10. S.A. Scherer, *Software Failure Risk*, Plenum Press, 1992.
11. B. Potter and G. McGraw, "Software Security Testing," *IEEE Security & Privacy*, vol. 2, no. 5, 2004, pp. 81–85.

Mehmet Sahinoglu *is the Alabama Commission on Higher Education eminent scholar and chairman of the computer science department at Troy University. He is currently researching stopping rules in cost-effective software and hardware testing, and component and system reliability and security indices. Sahinoglu has a BS from the Middle East Technical University, Ankara, an MS in electrical engineering from the University of Manchester Institute of Science and Technology in England, and a dual PhD in electrical engineering and statistics from Texas A&M. He is a fellow member of the Society of Design and Process Science, a senior member of the IEEE, and a member of the ACM, the Armed Forces Communications and Electronics Association, and the American Statistical Association. Contact him at mesa@troy.edu.*