

# Cyber Security Risk Assessment and Optimal Risk Management of a National Vulnerability Database

Mehmet Sahinoglu

Department of Computer Science, Troy University, Troy 36082, USA

Email: mesa@troy.edu

Manuscript received December 13, 2023; revised January 12, 2024; accepted May 30, 2024; published October 21, 2024

**Abstract**—This research article focuses principally on a certain quantitative rather than qualitatively subjective and non-numerical cyber security risk assessment method, the Security Meter (SM) algorithm, to compute vital security indices in a national vulnerability database case study. The primary purpose will be followed up by a secondary, although critical, goal of managing an unfavorable risk percentage by optimally mitigating it to one selected acceptable level. This objective will be realized by an optimization method using the well-known Linear Programming (LP) technique via both SM and an alternate LP-feasible solution method, EXCEL (XL) Solver. Information and cyber security risks become essential to an organization's or user's daily operations in today's IT-centric world. Vulnerabilities and threats can pose many challenges to the core security of any system, second only to the electric power grid that supplies the Internet. Without the vulnerability and associated threat-exposure management process, organizations remain blind and indifferent to those risks directly related to their IT infrastructure security. This advantage allows any organization or user (given they understand the security risks they face) to take well-advised decisions concerning remediating actions for managing the risks seriously with a cost-effective roadmap. Along with the rampant rise of the potential risks from unexpected cyber-attacks, damage due to uncountable breaches of cyber security is growing at an unprecedented rate and becoming a serious economic concern and peril to users, organizations, and nations. This research article proposes application-based quantitative analyses of commonly encountered security risks in a national vulnerability database, as initial steps toward the optimal security-centric technological investment-savvy evaluations and cost-effective decision-making processes to best manage and prioritize risk mitigation. The SM optimization results favorably compare with XL Solver solutions although SM's cumulative percentage of countermeasure changes to achieve the mitigation target is demonstrated to be less than that of the XL Solver's, and therefore, the more cost-optimal. As a major takeaway, the proposed quantitative algorithms are more competitive, practical, goal-oriented, functional, and cost-conscious than conventionally limited descriptive and categorical cyber security risk assessment and management options.

**Keywords**—quantitative, vulnerability, threat, countermeasure, Common Vulnerabilities and Exposures (CVE), Linear Programming (LP)-feasible, cost, Security Meter (SM), EXCEL (XL) solver, game theory, SysAdmin, Audit, Networking, and Security (SANS), National Institute of Standards and Technology (NIST), MITRE Corporation, Computer Emergency Readiness Team (CERT)

## I. INTRODUCTION AND MOTIVATIONAL ASPECTS

Cybercrime has become a crucial problem with many new viruses, malware, security breaches, and hacking techniques. With the increasing threat of cyber security breaches and information leaks, the importance of Computer and Information Security (CIS) in organizations is growing rapidly. Enterprises are investing significant resources to

accommodate risk analysis as the basis of information protection, risk assessment, and risk management [1]. According to SANS (deriving from SysAdmin, Audit, Networking, and Security) Information Security Resources, Information Security refers to the processes and methodologies that are designed and implemented to protect printed material, electronic or any other form of confidential information or data from unauthorized access, misuse, modification, or disruption [2].

As the Internet has become accessible, it has increasingly turned into a hunting ground for criminals, activists, and terrorists motivated to steal money, demand ransoms, compromise trust, cause disruption, or bankrupt corporations and governments through online attacks. Cybercrime and those associated ripple effects are estimated to cost around \$10.5 trillion annually by 2025 within a year, upwards climbing from \$3 trillion a decade ago at a growth rate of 15% every year since then [3]. Risk analysis includes processes such as identification of activity, threat exposure, and vulnerability analysis. This method is often called the metric-based approach. Any quantitative approach creates a precise numerical interpretation that can openly represent which risk-resolving measures have economically and engineering-wise been most well-suited. This makes the quantitative approach favored by many organizations since risk assessments can be clearly and explicitly represented in tangible forms like probabilities, percentages, or costs [4]. The Risk Assessment system has been outlined to be a security evaluation technique, which allows users to perform threat and vulnerability analysis [5]. The periodic assessment of risk toward an organization's operations or assets resulting from the operation of an information system is an important activity required by the Federal Information Security Management Act (FISMA) [6]. The application is used for risk assessment according to the National Institute of Standards and Technology (NIST) and Risk Management Guide for Information Technology Systems Common Vulnerabilities and Exposures [7].

By analyzing and segregating threat and vulnerability data, one would predict the likelihood of occurrence of each threat and vulnerability. One can use this data to analyze the most occurring vulnerabilities and threats in a system, and successfully calculate the residual risk helping to take preventions against any system exploitation in an organization [1, 4, 5, 8, 9]. To build a practical and accurate quantitative model, one needs to initially collect data from different sources following which the essential risk probabilities will be estimated using the equations that were developed. The purpose of the security risk assessment approach is also to provide organizations with a more

approachable view of assessment and management regarding the system dependencies between vulnerabilities [4, 9]. The data presented on the Common Vulnerabilities and Exposures (CVE) website provides the details of the vulnerabilities and threats. One can use this data to predict the most occurring vulnerabilities and threats with maintenance priorities using the empirical Bayesian principles [1, 4, 10].

Conventionally, risk scenarios involve possible chance-based catastrophic failures by 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 non-malicious chance failures that occur due to insufficient testing and lack of adequate reliability [11]. Organizations can use software reliability modeling and testing techniques to examine these chance failures in more detail [12]. There are two fundamental types of risk analysis: i) Qualitative and ii) Quantitative. Qualitative risk analysis does not involve numerical probabilities or predictions of loss. They are usually represented by inadequate non-numerical labels, such as high, medium, and low, or colors of the spectrum [1, 4, 13, 14]. The quantitative risk analysis, whereas, involves numerical probabilities of various adverse events and also determines the extent of losses if a particular event occurs. Several security risk templates employ non-quantitative attributes to express a risk's severity, which is subjective and void of actual figures. The author's SM with its decision tree diagram-oriented design provides a quantitative technique with an updated repository on vulnerabilities, threats, and countermeasures to calculate the inherent risk [1, 4, 5, 14].

However, for the intentional failures or malicious activities that critically increase the risk of ill-defined attacks, no previous work has ever thoroughly modeled a physical scenario, at least not one that considers a probabilistically holistic and consistent framework of vulnerabilities, threats, and countermeasures [1, 4, 5, 14]. The proposed Security Meter design fills a void in the arena of much-desired quantitative risk evaluation favorably compared to all current assessments that provide qualitative results. This is achieved by the virtue of a math-statistically accurate quantitative model that calculates the security risk. The design's concrete numerical approach, which always works for all systems, can further facilitate security risk management as well as security testing. This means that the final risk measure calculated as a percentage can be tested, improved, compared, and budgeted as opposed to the conventionally subjective attributes, which cannot be managed or quantified numerically for an objective assessment rather than a subjective one. 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 risk by classifying them as low, medium, or high, or else using different colors as said. This approach is not only highly subjective, but it also lacks any actual risk figures. In existing analyses that favor a qualitative approach, either a probabilistic frame about whether to add or multiply risks doesn't exist, or the risk calculations are handled case-by-case without a network-centric perspective. This author personally observed to see risks exceed unity (red flag in probability discipline) when added to each other casually by

a lecturer—because until then, no one had proposed the Security Meter decision tree diagram—an event which triggered this author to design a practical and accurately working probabilistic model [1, 4, 5, 14, 15].

Without this probabilistic framework such as the one suggested in the Security Meter (SM) design, the conclusions to assess a risk's severity might be misleading and costly, especially during military conflicts and wars, where risk scenarios are often over- or underestimated due to lack of quantification. The proposed SM 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 those with personal computers.

Risk management, whereas, is the holistic process of identifying, measuring, and minimizing the uncertain events that can affect resources. This definition also implies the process of bringing management or remedial action and solid control into the risk analysis. A basic ingredient of risk assessment and management 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 [1, 4, 5, 14]. 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 unwanted consequence if it becomes an attack on an asset. One 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 of Countermeasure (LCM). A CM is an action, device procedure technique, or another measure that reduces risk to information. The complement, LCM is the lack of CM action.

Consequently, the residual risk is the portion of remnant risk after a certain remedial measure of CM is applied. The residual risk will not exist if perfect conditions for CM exist. The proposed physical model in Figs. 1 and 2 identify the deterministic (constant) and probabilistic (random) inputs for the output of the calculated residual risk and the expected cost of loss to fully mitigate it. The Residual Risk (RR) is defined as the portion of risk that remains after countermeasures are applied. If all countermeasures are applied properly and perfectly well in the organization, no RR will remain to exist.

The Non-Residual Risk (NRR) is the complement of RR and the risk measure that an activity would pose if all pertinent controls or other mitigating factors were in place, i.e.,  $NRR = 1 - RR$ . Capital Cost (CC) is the total expected loss in monetary units (e.g., \$) for the particular system if it is destroyed and can no longer be utilized, excluding the shadow or opportunity costs, had the system continued to generate value [1, 4, 5, 14].

System criticality, which is another constant that indicates the degree of how critical or disruptive the system is in the event of an 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 the residual risk is of little or no significance, such as the occasional malfunctioning of an office printer. If not confident, analysts use 0.5. However, in the case of a nuclear power plant, criticality is ~100% because its security has vital life and death ramifications [5, 14].

## II. RISK, RISK ASSESSMENT, AND RISK MANAGEMENT

Risk is generally defined as the potential harm that may arise from an actual current process or future event [16]. The management of information security essentially boils down to mitigating that risk wisely with a planned purpose. The purpose of this article is to build a model by performing vulnerability and threat analysis toward risk assessment and risk management using an innovative and unique Security Meter (SM) risk optimization algorithm [1, 4, 5, 14].

Once risks have been identified, they must be assessed as to their potential severity of impact and the probability of their likelihood of occurrence. These quantities can be either simple to measure, in the case of the value of a damaged building, or impossible to know in the case of the probability of an unlikely malware episode. To properly prioritize the implementation of the risk management plan, it is imperative to make the best decisions in the risk assessment process. Risk Assessment allows organizations to determine the level of security controls required and allows them to justify the decisions taken demonstrably [5, 14].

According to the Information Systems Audit and Control Association (ISACA), information risk management defines the areas of an organization's information infrastructure and identifies what information to protect and the degree of protection needed to align with the organization's tolerance for risk [17]. Humans use heuristics or rules of thumb for dealing with risk, but these don't serve adequately in many business and public policy situations and are frequently deceptive, thus causing irreparable harm. Factually, in-depth research shows that they have cognitive biases, such as overweighing or exaggerating the most recent adverse event and projecting current good or bad outcomes too far into the future, all working against one's desire to make the best decisions. The quantitative risk analysis can help avoid these impartial biases although inadvertently, and make better decisions, such as in the example of humidifiers activated by a thermostat, rather than sufficing with less or more humidity.

An analyst can define the security risk, for instance, in the form of 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 vandalistic destruction of an asset if not counter-measured on-site by items such as a motion sensor ( $CM_{11}$ ) or a fire alarm ( $CM_{12}$ ), respectively. Similarly, imagine the network server's installed software as an asset with a vulnerability ( $V_2$ ) against which a threat ( $T_{21}$ ), such as a virus, or a hacker ( $T_{22}$ ), could result in an undesirable corruption of the software if not counter-measured by items such as anti-virus software ( $CM_{21}$ ), or a firewall-based hardware or software device ( $CM_{22}$ ), respectively [1, 4, 5, 8, 14].

## III. LITERATURE SURVEY, COMPARISONS, AND CONTRASTS

In the past decade, important works have been published in the overall literature dealing with quantifying risk [18–26].

These have followed different methods employing various computational techniques with different programming codes from Java to C++ and Python. What makes this research article innovative is its easy-to-comprehend, and simplistic algorithmic approach using SM, a risk assessment, and management software, to overcome a long-lasting overdue issue for agencies, such as SysAdmin, Audit, Networking, and Security (SANS), National Institute of Standards and Technology (NIST), Computer Emergency Readiness Team (CERT) *et al.* The SM's risk management solution outsmarts others and it is validated by the EXCEL Solver by Microsoft.

Other qualitative models and methods, such as Attack Trees and Time-to-Defeat are only deterministic but not quantitative or cost-convertible [1]. Recently, in the latter decade, various scoring systems have mushroomed on the internet such as Common Vulnerability Scoring System (CVSS), National Vulnerability Database Version 2.2 (NVD, Bricade CVSS V2.0 Calculator JVN RSS), CVSS Version 2.0 (JVN), etc., as in Figs. A1 to A3. In the JVN version for one example, JVN is a community-backed solution for describing software security vulnerabilities and it is used as a baseline for vulnerability remediation activities. The purpose is to prioritize the patch development and concurrently communicate the severity to the clients. On the other hand, the scores are computed in sequence such that the Base Score is used to calculate the Temporal Score, and the Temporal Score is used to calculate the Environmental Score. The CVSS quantitative model ensures repeatable accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the scores. No cost and benefit considerations and no room to perform risk mitigation using optimization techniques exist [4]. None of these in the preceding paragraph offer quantitative risk assessments, except for dealing with the popular Confidentiality, Integrity, and Availability attributes, or simple categorical data. The subjective attributes do not lead either to cost and benefit considerations or remedial actions to reduce the risk.

The Triple Uniform Product Rule can only assess risk which requires intensive calculations using LINGO software to validate SM results [4, 18]. However, there is no risk management roadmap available afterward. By the Refs. [1, 27], the author focuses on a universally applicable privacy meter applicable to the security risk assessment calculations using a statistical Compound Poisson (~Negative Binomial) approach with a cost and benefit analysis, assuming a special rule [4]. Sahinoglu *et al.* [18] use LINGO software which has limitations. Bojanc and Jerman-Blazic [19] present a mathematical model for the optimal security-technology investment evaluation and decision-making processes based on the quantitative analysis of security risks and digital asset assessments. However, the proposed practical software does not support a purely quantitative model and a cost-optimal risk management plan such as those herein proposed.

In an M.S. thesis supervised by the author of this article, Ashokan [20] uses PHP code for a risk assessment system. Then, a JAVA code for optimizing risk using Linear Programming follows. The said algorithm is not an easy-to-use procedure with too many programming details, and not readily scalable. The limitations are those to assess the risk using national threat and vulnerability data, the accuracy

problem arises due to Big Data. The data collected from the CVE website was analyzed for 2000 rows of sample data while deciding to split threats and vulnerabilities. This has resulted in duplicity of data in the NVD and has resulted in more memory capability and cost. Yet, this thesis reference is, a project that comes closest it could to the current user-friendly, practical yet scientific, scalable algorithm, no other than SM, to generate industry-standard results.

Evrin outlines qualitative and quantitative risk analysis techniques; however, this descriptive and illustrative online collection of definitions and figures does not lead the user to any practical software to work out an NVD of this article's dimensions, and obtain outputs, let alone no inkling of cost and benefit is offered [21].

Meyer's [22] is of the same content as in [21] in the sense that illustrations and figures explain in the style of four operations how to calculate risk based on very simple basics and rubrics. No extensive data is used, and no risk mitigation is carried out with any cost and benefit calculations as before.

Tan [23], similar to [21, 22], presents both qualitative and quantitative illustrative techniques with no substantial software to estimate and mitigate risk using cost and benefit.

Schmittling and Munns [24] have very similar characteristics to [21–23] with no comparative benefits to this article's proposed Security Meter goal in search of risk estimation and risk management. The reference does not offer any method or data collection techniques to do as proposed to assess and manage risk. A wireless-sensor-oriented proceedings paper completes the risk assessment and management cycle but does not touch the NVD problem [25].

Bansal [26] does not offer to assess risk and manage it. This reference is verbose with many categorical descriptions and illustrations. No software or data collection techniques are offered in any manner compatible with the Security Meter, and both references fail to contribute to risk management and do not lead to any cost and benefit analysis with remediation.

To recap, all past works in [18, 19, 21–26], do not come close anywhere near the computationally intensive execution aimed for in this article for SM and XL Solver algorithms, let alone software-supported risk management including a cost and benefit-oriented econometric analysis in the end. Besides [19, 20], no other contributions are significant except for [18], a lengthy LINGO software solution with no risk management and cost-and-benefit considerations on privacy.

The quasi-millions of computations executed in the Java-coded Security Meter software, will not only assess risk but consequently discover a unique feasible solution. That unique feasible solution will enable the analyst with the least cumulative percentages of  $\pm$  changes in the  $CM_{ij}$  or its complement,  $LCM_{ij}$ . The investment cost involved to accomplish the job for a favorable mitigation from the current undesirable toward a given tolerable risk percentage will be minimized. The Security Meter software is also scalable such that one can modify the cost parameters and vulnerability counts, i.e., the size of the unknown vector,  $LCM_{ij}$ , as desired, hence providing leverage. Once the optimization target is specified, such as 20% and 30%, valuable advice is provided by the SM software as to what to do concerning the mitigation target with cost and benefit analysis in example 2. The execution may continue after pruning the tree limbs (i.e., the threats that are ~100% salvaged) of the decision tree in

Fig. 3. Namely, the analyst will run the SM software for a new optimized preference reduced to such as 10% as specified in example 1 [5] of Figs. A1–A3, or after the initial attempts of mitigations to 20% or 30% as in example 2 [27–30] detailed in Tables A4–A14. This is not sustainable because the analyst may have a limited investment budget, which may not be allowed to exceed. The SM algorithm provides the most money-saving feasible solution for the optimized vector of the  $LCM_{ij}$  or the  $CM_{ij} = 1 - LCM_{ij}$  among all alternatives [1, 4, 5, 14].

#### IV. METHODOLOGY

Before a comprehensive methodology, one needs to examine certain basic foundations as to how to challenge these objectives.

##### A. National Vulnerability Database Descriptions

NVD is the U.S. government repository of standards-based vulnerability management databases presented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. NVD includes databases of security checklists, security-related software flaws, misconfigurations, product names, and impact metrics [28]. In particular, NVD supports the Common Vulnerability Scoring System (CVSS) version standards for all Common Vulnerabilities and Exposures (CVE) vulnerabilities [27–30]. NVD provides CVSS base scores which represent the innate vulnerability characteristics. It does not currently provide temporal scores that change over time due to events external to the vulnerability. However, NVD does provide a CVSS score calculator to allow you to add temporal data and even calculate environmental scores customized to reflect the impact of the vulnerability on an organization [27]. This calculator contains support for U.S. government agencies to customize vulnerability impact scores based on Federal Information Processing Standards (FIPS) System ratings.

Tables A4 and A5 in example 1 help readers attain an idea of why and how these vulnerabilities and their associated threats occur and what to do. Here are their basic reviews:

##### 1) Buffer overflow

A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. When this occurs unwanted things happen, such as a system crash or an opportunity for an attacker to run an arbitrary code [31].

##### 2) Web server

The first task of a Web attacker is to determine your operating system, Web or application server, and database platforms. Thus, for remedy, removing or obfuscating the signatures of your technology platforms is a useful route.

##### 3) Javascript

JavaScript enables malicious actors to deliver scripts over the web and run them on client computers. Many JavaScript security vulnerabilities are the result of browser authors failing to take these measures to contain Document Object Model (DOM)—based JavaScript security risks [32].

##### 4) Race condition

A race condition (or hazard) is the behavior of software or

other system where the output is dependent on the sequence or timing of other uncontrollable events [33].

#### 5) Cross-site scripting

Cross-site scripting (XSS) enables attackers to inject client-side scripting into Web pages viewed by other users. An XSS vulnerability may be used by attackers to bypass access controls such as the same-origin policy [34].

#### 6) SQL injection

Structured Query Language (SQL) Injection is a hacking technique, which attempts to pass SQL commands through a web application for execution by the backend database. SQL injection is mostly known as an attack vector for websites [35].

#### 7) File Inclusion

File inclusion vulnerability enables an attacker to include a file, usually through a script on the web server. This leads to something as minimal as outputting the contents of the file or more serious [36]. Code execution on the client-side such as JavaScript can lead to other attacks such as Cross-Site Scripting (XSS) and Denial of Service (DoS) [37].

#### 8) Format string

The Format String exploit occurs when the submitted data of an input string is evaluated as a command by the application. In this way, the attacker could execute code, read the stack, or cause a segmentation fault in the running application, causing new behaviors that could compromise the security or the stability of the system [38].

#### 9) Directory traversal

A directory traversal (or path traversal) is to order an application to access a computer file that is not intended to be accessible. This attack exploits a lack of security as opposed to exploiting a bug in the code [39].

#### 10) Untrusted search path

The application searches for critical resources using an externally-supplied search path that can point to resources that are not under the application's direct control. If the application uses a search path to locate critical resources such as programs, the attackers could modify that search path to point to a malicious program which the target application would then execute [40].

Tables A1–A3 from Ref. [5], and Tables A4–A7 from NVD [27–30] display input and output tabulations of examples 1 and 2, respectively, regarding the SM Risk Assessment and Management algorithms with their associated cost and benefit analyses.

### B. Quantitative Risk Analysis

A quantitative risk assessment provides results in numbers that the management can understand; whereas a qualitative approach, although easier to implement with categorical likelihoods and their impact, makes it difficult to trace generalized results, let alone make comparisons in terms of hard currency. IT risk is most often represented in terms of expected losses. The losses may include repair costs to information systems or the replacement cost for an asset that is stolen or lost. It assists the user in determining the associated cost and benefit analysis. Then, the probability of the incident can be calculated as the product of the probability of the exposed Threat (T) and asset Vulnerability (V). The

threat probability is defined as the probability of an attack on information assets. It is equal to the number of attacks per unit of time. System vulnerability V is defined as the probability of a threat that is successfully realized in the form of an incident on informational assets. If there is any security incident in the organization, there will be a financial Expected Cost of Loss (ECL) incurred in the organization which will be measured in monetary units. The probability of a security incident occurrence is defined as the number of times that a particular threat can occur during a given time interval. Fig. 1 shows the link between Likelihood, Impact, and Risk which involves vulnerabilities and threats. Risk Optimization is important to consider following a risk assessment in practice [41].

### C. How to Quantify Risk

Data for malicious attacks that have been prevented or not prevented from penetrating are collected. The probabilistic inputs are vulnerability, threat, and LCM (Lack of countermeasure) of all risks that range between 0 and 1. The constants are the capital cost and criticality constant (between 0 and 1). The residual risk and expected cost of loss are the outputs obtained using Eqs. (1)–(4) that follow. The so-called black box in Fig. 2 after an illustrative Fig. 1 leads to the probabilistic tree diagram of Fig. 3 to run the calculations. In Fig. 3,  $V_1$  and  $V_2$  are vulnerabilities, whereas  $T_1$  and  $T_2$  are threats for respective vulnerabilities.  $LCM_{11}$  is the lack of countermeasure for vulnerability  $V_1$  and threat  $T_1$ ,  $LCM_{12}$  is the lack of countermeasure for vulnerability  $V_1$  and threat  $T_2$ ,  $LCM_{21}$  is the lack of countermeasure for vulnerability  $V_2$  and threat  $T_1$ , whereas  $LCM_{22}$  is the lack of countermeasure for vulnerability  $V_2$  and threat  $T_2$ . Eqs. (1)–(4) summarize the contents of Figs. 1–3 from input to output. If malicious attacks are recorded, one needs to come up with a ratio of failed attacks and a ratio of successful attacks. Out of many such attempts, the number of penetrating attacks divided by the total number of attacks will yield the estimate for the risk ratio of LCM. One can then trace the root of the cause to the threat level retrospectively from the outcomes in the tree diagram. Let us imagine that the Anti-Virus (AV) software does not catch the hazard, and a virus attack occurs to reveal the threat exactly. As a result of this attack, whose root threat is known; the e-mail system may be disabled. Then, the vulnerability comes from the e-mail asset itself. This way, one completes the line of attack on the tree diagram as in Fig. 3. The following Eq. (1) computes the RR for each activity in Fig. 3 related to each leg [1, 4, 5, 14] where  $k$  (=criticality) varies between 0 and 1. Fundamental equations follow:

$$RR_{i,j} = P(V_i) \times P(T_j | V_i) \times LCM_{i,j} \quad (1)$$

Covering all legs in Fig. 3, RRs ( $0 < RR < 1$ ) sum to Total Residual Risk (TRR), ( $0 < TRR < 1$ ). FR = Final Risk, CC = Capital Cost and ECL = Expected Cost of Loss follow:

$$FR = TRR \times k \quad (2)$$

$$ECL = FR \times CC \quad (3)$$

$P(V_i)$  is the probability of vulnerability  $V_i$ . The conditional probability of the threat,  $T_j$ , given  $V_i$ , is  $P(T_j | V_i)$ .

FR is the Final Risk. The TRR is calculated using Eq. (4).

$$TRR = \sum_i^n \sum_j^n RR_{i,j} \quad (4)$$

Note that the TNRR (Total Non-Residual Risk) =  $1 - TRR$ .

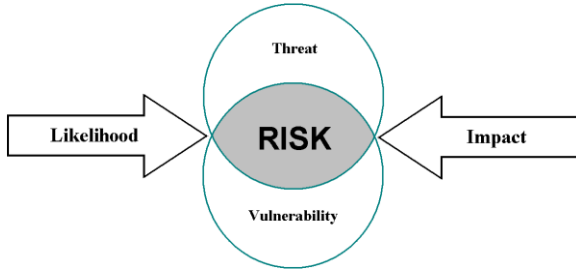


Fig. 1. The link between Likelihood, Impact, and Risk.

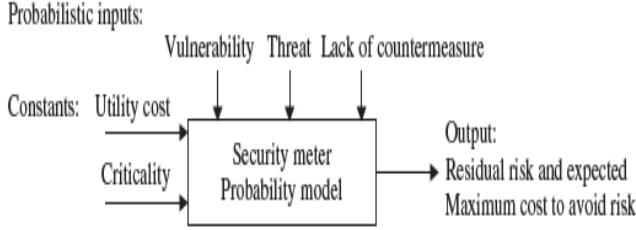


Fig. 2. Quantitative security risk model (black-box) of the probabilistic and deterministic (constant) input data and final output solutions.

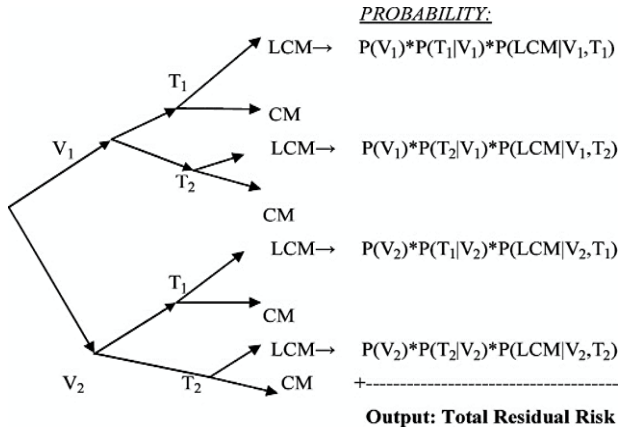


Fig. 3. General tree diagram for the two threats conditional on each of the two vulnerabilities (V for branches, T for twigs, LCM for limbs).

## V. SECURITY METER (SM) AND EXCEL (XL) SOLVER

This pivotal section establishes a robust foundation and sets the primary purpose for the proposed article title. Namely, it consolidates the principles of the SM risk assessment, and SM risk management (in Appendix A and Appendix C), as well as that of the EXCEL Solver (Appendix B) for validation and comparison purposes. Both methods provided by the game theoretic LP algorithm generate a feasible vector solution minimizing the investment cost. But SM achieves this with the least cumulative percentage of changes in the Countermeasure (CM) or Lack of Countermeasure (LCM) [4]. The SM software initially performs the Risk Assessment before executing the Risk Management (see Tables A1–A7, for examples 1 and 2, respectively). But there is no stand-alone Risk Assessment routine in the Microsoft EXCEL Solver itself whereas manually appended, only the LP solution. Security Meter software is downloadable to one's PC (Windows-based) or laptop as described in Appendix C. Quantitative risk measurements are needed to objectively compare alternatives and calculate costs to budget for reducing or minimizing the existing risk. There exist virtually no such quantitative and

probabilistic measures in academia or corporate circles other than descriptive categorical denominations subject to any interpretations as one pleases. They do not carry any analytical cost-related evaluations for comparisons when mitigation is performed. Among those existing analyses that favor a quantitative study, either i) there is no probabilistic frame about whether to add or multiply risks in a correct probabilistic frame or ii) the risk calculations are handled on a singular basis without an overall system picture in mind. Given that in a simple scenario, there are two or three or more of each, the probabilistic frame in Fig. 3's tree diagram holds. Note that the sum of  $V_i = 1$  and the sum of  $T_{ij} = 1$  for each  $i$ , and the sum of  $LCM_{ij} + CM_{ij} = 1$  for each combination of  $i$  (#vulnerabilities) and  $j$  (#threats), i.e.,  $ij$ , in a tree diagram per Fig. 3 and refs. [1, 4, 5, 14].

### A. Linear Programming (LP)

Linear programming (also called linear optimization) is a method to achieve the best outcome (such as maximum profit or minimum cost) in a mathematical model whose requirements are represented by linear relationships. Linear programming is a special case of mathematical programming. Linear programming problems are optimization issues, where the objective function and the constraints are all linear. The objective of all linear programming problems is the maximization or minimization of a given target quantity, similarly phrased as the objective function. Linear programming is the most popularly used sector of optimization [42, 43]. In operations research, many practical projects can be expressed as linear programming problems. Historically, ideas from linear programming have inspired many of the central concepts of optimization theory, such as duality, decomposition, and the importance of convexity and its generalizations. Similarly, linear programming is densely applicable to micro- and macroeconomics and company management, such as planning, production, transportation, technology, and many other disciplines. Although modern management issues are ever-changing, most companies would like to maximize profits or minimize costs with limited resources. Many issues can be formulated by LP. No better feasible solutions exist than those of the SM applications [44–50]. Tables A4–A14 will further support this in Appendix A.

### B. Constraints and Variables, and Governing LP Equations

A constraint is an equation or inequality that rules out certain combinations of decision variables as feasible solutions. A decision variable is a controllable input for a linear programming model. One considers a probabilistic variable, LOSS, to correspond to the equilibrium value when feasibly solved by the LP problem, so minimizing overall loss or maximizing overall gain. Minimax (Minimizing the maximum gain from the defender/user side) will be equal to Maximin (Maximizing the minimum loss from the offender/attacker side) in a zero-sum two-player game. The optimal equilibrium value obtained with the LP lies between Minimax and Maximin by Neumann's Mixed Strategy [4, 46–49]. Threats of  $n = 20$  counts in Appendix A's example 2 are from the original Table A4 which merged into Table A5 in terms of attackers and users, only.

The 20 non-negativity constraints require all variables to be nonnegative.  $LCM_{i,j}(\text{optimal}) = 1 - CM_{i,j}$  in Eq. (5) is the

unknown vector for an optimization problem, i.e. the optimized lack of countermeasure for the vulnerability  $V_i$ , and the threat  $T_j$ . The Objective Function is Min LOSS s.t. constraints of Eqs. (5)–(49) or (5)–(50) by skipping Eq. (49).

$$0 < LCM_{i,j}(\text{optimal}) \leq 1; i = 1, \dots, 10; j = 1, 2 \quad (5)$$

A constraint for the improvement of the lack of countermeasure is to optimize or minimize the given  $LCM_{i,j}$  vector for the vulnerability  $V_i$  and the threat  $T_j$ ; that is,  $V_i T_j$ , expressed in the constraint of Eq. (6):

$$LCM_{i,j}(\text{optimal}) \leq LCM_{i,j}(\text{original}); i = 1, \dots, 10; j = 1, 2 \quad (6)$$

Follow the constraints of Eqs. (7)–(26) (i.e., 20 constraints) per Appendix A and Appendix C:

$$LCM_{11} < 0.9180795 \quad (7)$$

$$LCM_{12} < 0.2520777 \quad (8)$$

$$LCM_{21} < 0.0311604 \quad (9)$$

$$LCM_{22} < 0.8917579 \quad (10)$$

$$LCM_{31} < 0.6319202 \quad (11)$$

$$LCM_{32} < 0.2626705 \quad (12)$$

$$LCM_{41} < 0.5665642 \quad (13)$$

$$LCM_{42} < 0.0786773 \quad (14)$$

$$LCM_{51} < 0.8208223 \quad (15)$$

$$LCM_{52} < 0.9718796 \quad (16)$$

$$LCM_{61} < 0.4384332 \quad (17)$$

$$LCM_{62} < 0.6243588 \quad (18)$$

$$LCM_{71} < 0.5990462 \quad (19)$$

$$LCM_{72} < 0.2798739 \quad (20)$$

$$LCM_{81} < 0.996674 \quad (21)$$

$$LCM_{82} < 0.1558227 \quad (22)$$

$$LCM_{91} < 0.3038783 \quad (23)$$

$$LCM_{92} < 0.08006754 \quad (24)$$

$$LCM_{10,1} < 0.7718623 \quad (25)$$

$$LCM_{10,2} < 0.6743795 \quad (26)$$

The game-theoretic constraints are to minimize the LOSS. In constraint of Eq. (27),  $P(V_i)$  and  $P(T_j | V_i)$  are the probability of vulnerability and the conditional probability of threat given vulnerability, respectively, as follows:

$$P(V_i) \times P(T_j | V_i) \times LCM_{i,j} - \text{LOSS} < 0; i = 1, \dots, 10; j = 1, 2 \quad (27)$$

The game-theoretic 20 constraints of  $LCM_{i,j}(\text{opt.})$  from the constraint of Eq. (28) on to Eq. (47) are as follows:

$$0.18141960 \times LCM_{11} - \text{LOSS} < 0 \quad (28)$$

$$0.03658040 \times LCM_{12} - \text{LOSS} < 0 \quad (29)$$

$$0.03508485 \times LCM_{21} - \text{LOSS} < 0 \quad (30)$$

$$0.01021515 \times LCM_{22} - \text{LOSS} < 0 \quad (31)$$

$$0.00706024 \times LCM_{31} - \text{LOSS} < 0 \quad (32)$$

$$0.00003976 \times LCM_{32} - \text{LOSS} < 0 \quad (33)$$

$$0.00375375 \times LCM_{41} - \text{LOSS} < 0 \quad (34)$$

$$0.00874625 \times LCM_{42} - \text{LOSS} < 0 \quad (35)$$

$$0.28869096 \times LCM_{51} - \text{LOSS} < 0 \quad (36)$$

$$0.00150904 \times LCM_{52} - \text{LOSS} < 0 \quad (37)$$

$$0.21235830 \times LCM_{61} - \text{LOSS} < 0 \quad (38)$$

$$0.00954170 \times LCM_{62} - \text{LOSS} < 0 \quad (39)$$

$$0.08186191 \times LCM_{71} - \text{LOSS} < 0 \quad (40)$$

$$0.00023809 \times LCM_{72} - \text{LOSS} < 0 \quad (41)$$

$$0.01322541 \times LCM_{81} - \text{LOSS} < 0 \quad (42)$$

$$0.00507459 \times LCM_{82} - \text{LOSS} < 0 \quad (43)$$

$$0.08342112 \times LCM_{91} - \text{LOSS} < 0 \quad (44)$$

$$0.00697888 \times LCM_{92} - \text{LOSS} < 0 \quad (45)$$

$$0.00180056 \times LCM_{10,1} - \text{LOSS} < 0 \quad (46)$$

$$0.01239944 \times LCM_{10,2} - \text{LOSS} < 0 \quad (47)$$

The additional two constraints are the non-negativity constraints for the  $\text{LOSS} > 0$  variable and the constraint of Eq. (49) or (50) to mitigate the total risk from the current percentage to a certain desirable percentage of a lesser value.

The non-negativity of the 21<sup>st</sup> variable,  $\text{LOSS} > 0$ , which is the 41<sup>st</sup> constraint, is acceptable by default of non-negativity.

The 42<sup>nd</sup> constraint instead is the sum of the optimized risks when the OTRR is the final constraint of Eq. (49) or (50) for  $\text{OTRR} \leq 0.2$  and  $\text{OTRR} \leq 0.3$ , respectively.

For all 42 constraints, i.e.,  $2n$  (=size of LCM vector) + 2 =  $2(20) + 2 = 42$ , and nonnegative  $\text{LOSS} > 0$  to achieve a risk mitigation from 62.95% to 20% while reaching Eq. (49) for  $\text{OTRR} \leq 0.2$  and similarly  $\text{OTRR} \leq 30\%$  from 62.95% while reaching Eq. (50) for  $\text{OTRR} \leq 0.3$ , as follows:

$$\text{LOSS} > 0 \quad (48)$$

$$\begin{aligned} \text{OTRR (Optimized Total Residual Risk)} &= 0.1814196 \times \\ &LCM_{11} + 0.036580 \times LCM_{12} + 0.03508485 \times LCM_{21} + \\ &0.01021515 \times LCM_{22} + 0.0070602 \times LCM_{31} + 0.00003976 \times \\ &LCM_{32} + 0.00375375 \times LCM_{41} + 0.00874625 \times LCM_{42} + \\ &0.28869096 \times LCM_{51} + 0.00150904 \times LCM_{52} + 0.2123583 \times \\ &LCM_{61} + 0.0095417 \times LCM_{62} + 0.08186191 \times LCM_{71} + \\ &0.00023809 \times LCM_{72} + 0.01322541 \times LCM_{81} + 0.00507459 \\ &\times LCM_{82} + 0.08342112 \times LCM_{91} + 0.00697888 \times LCM_{92} + \\ &0.00180056 \times LCM_{10,1} + 0.01239944 \times LCM_{10,2} \leq 0.2 \quad (49) \end{aligned}$$

$$\begin{aligned} \text{OTRR (Optimized Total Residual Risk)} &= 0.1814196 \times \\ &LCM_{11} + 0.036580 \times LCM_{12} + 0.03508485 \times LCM_{21} + \\ &0.01021515 \times LCM_{22} + 0.0070602 \times LCM_{31} + 0.00003976 \times \\ &LCM_{32} + 0.00375375 \times LCM_{41} + 0.00874625 \times LCM_{42} + \\ &0.28869096 \times LCM_{51} + 0.00150904 \times LCM_{52} + 0.2123583 \times \\ &LCM_{61} + 0.0095417 \times LCM_{62} + 0.08186191 \times LCM_{71} + \\ &0.00023809 \times LCM_{72} + 0.01322541 \times LCM_{81} + 0.00507459 \\ &\times LCM_{82} + 0.08342112 \times LCM_{91} + 0.00697888 \times LCM_{92} + \\ &0.00180056 \times LCM_{10,1} + 0.01239944 \times LCM_{10,2} \leq 0.3 \quad (50) \end{aligned}$$

The reader is urged to explore Appendix A to observe input and output tables, and how the two competing methods, SM and XL, are executed, cross-validated, and compared for investment cost-efficiency. Section V contains no results, only the methodology and a battery of fundamental governing



equations to cover the game-theoretic LP optimization from the constraints of Eqs. (5)–(49) or Eqs. (5)–(50) by skipping Eq. (49) for example 2. These solutions are in Appendix A.

## VI. CONCLUSIONS AND DISCUSSIONS

The risk assessment system helps organizations or enterprises decide on the necessary security investments in terms of security measures that are most effective. However, introducing a new vulnerability and threat management process within an organization or enterprise can also be administratively challenging. To ensure a successful vulnerability management process, the organization needs to filter vulnerabilities and threats that suit the needs of the organization. Finally, when starting with vulnerability and threat assessment, it is important to limit the scope of the initial vulnerabilities and threats to avoid unnecessary overcrowding and information deluge.

This application-oriented and computationally intensive research article is based on the quantitative assessment of security risks, and it allows for the evaluation of different investment options in an information security framework. The risk assessment system leads an organization from the initial input of data to those final recommendations for the selection of an optimal measure that reduces prevalent security risk. By using the risk assessment and management systemic procedure, enterprises can avoid, or track any new or existing threats and vulnerabilities that can pose a risk. The proposed SM algorithm helps to effectively prioritize the mitigation of threats and vulnerabilities of importance from the organization's perspective.

In the process of evaluating the optimal level of investment in the information security roadmap, it is necessary to quantify the threats and vulnerabilities that are related to an information asset as well as any measures to reduce these risks. By using a quantitative analysis approach for the evaluation of vulnerabilities and threats, one calculates the optimal solutions using ECL, TRR, TNRR, OTRR, and OTNRR, and constraints as formulated in example 2's governing equations from constraints of Eqs. (5)–(49) and (5)–(50) by skipping Eq. (49) relevant to the binding risk mitigation constraints. The application software proposed will enable the user to calculate the probability of occurrence of each threat and vulnerability based on those existing incidents to intercept future anticipated attacks.

The incentives for evaluating security risks are so compelling and indispensable that one should, rather than not, make reasonable estimates [1, 4, 5, 14]. In this article, one examines new scientific ways to estimate and infer probabilities, empirically by observing the frequencies of outcomes and calculating the associated losses. In this way, one is kept informed about the extent of the cost of bringing hardware and software systems to a desirable percentage of security from an unwanted adverse insecurity level. The difficulty in data collection parameter estimation poses a challenge to practitioners in the testing field. The author has employed the concept of a simple relative frequency approach, otherwise known as a simple counting technique [1, 4, 5, 14]. Although one may not predict the outcome of a random experiment with Big Data, one could by the law of large numbers predict the relative frequency, which denotes the ratio of desirable events to the sample size. Thus, the

outcome will rest within a desirable statistical confidence interval. In Ref. [5]'s Fig. 7 which corresponds to Table A1 supported by Tables A2 and A3 for example 1 in a different but smaller case scenario, the author outlines the uniquely feasible solution to outsmart all other contenders using a game-theoretic approach. Whereas employing an alternative Nonlinear Minimization Solution of the Portfolio Variance by LINGO Software; the proposed game-theoretic SM approach is observed to yield better economic results than the Portfolio by Markowitz for the same SM scenario, i.e., 90.52% cumulative change (which no other feasible solution can outsmart) vs. a relatively higher, 309% cumulative change by Markowitz, as presented in various invited seminars [44–50].

In the risk assessment system of vulnerabilities and threats, one should compute the overall risk by randomizing, or simulating the uniform random variates for the  $LCM = 1 - CM$  probability, if not provided apriori. For an accurate calculation of risk by the industry standards as laid out by the NIST, and as created by the U.S. Computer Emergency Readiness Team (CERT), it is highly recommended to provide LCM values from the respective organizations and enterprises that supply data to complete the picture. The goal is to obtain the weights of the diversity of threats so that a particular threat may be considered more influential and costly than the others in the pool. As the data keeps updating, one needs to obtain these values from organizations and enterprises timely for accurate calculations. One should be able to allow the users to enter and store raw data from the NVD, and extract information regarding threats and vulnerabilities for security risk management [4, 51, 52].

To allow different commercial enterprises as well as state agencies, and users to store the vulnerability and threat data according to the specifications and standards established by their organization, a new table may be created to store the data as per their needs. Since the agencies were not able to provide the researchers with the details of the particular threat being intercepted or not, one has no choice but to justify the LCM values in Appendix A and Appendix B to be simulated or randomly drawn between 0 and 1 through the EXCEL uniform random generator, `RANDC()`. The re-scalable SM software can be adapted to cover larger input data to accommodate more than the current 10 counts of rows for risk mitigation. Moreover, the COVID-19 pandemic highlighted the need for more cyber security in a wider world [53–55].

Security Meter has applications to the Electric Power industry analytically using the associated software, beyond which Discrete Event Simulations (DES) are also applicable using the Cloud Computing framework on an annual basis of 8760 h [56–58]. The SM method, beyond cybersecurity, has had a wide range of applications from National Defense to Healthcare, and from Business to On/Off Shore Oil Drilling, and from Ecology to Digital Forensics and Defense Acquisition to Wireless Security [4, 59–66]. Computationally intensive Monte Carlo simulations validated SM outputs [67].

As a key takeaway, the Microsoft EXCEL Solver alternatively validated the SM risk management stage although the SM Risk Management software was more cost-effective toward a more prudent investment plan to achieve risk goals. Namely, to quantify the risk of an NVD, and manage the associated risk, a culminating and up-to-date-not-encountered innovative method is introduced. This was a



long-expected application-oriented, user-friendly, practical, and scalable breakthrough in the cybersecurity research arena regarding risk quantification and risk management.

Rather than this proposed application-oriented practical approach supported by a two-decade-long background work by the author toward a universally accepted Cybersecurity Risk Assessment and Risk Management goal, Digital Information Security experts in the commercial business or government circles may be revolving around rules and regulations bound by strict policy matters as opposed to identifying the elephant in a China shop, and finding a clear-cut scalable and universally applicable end-solution with the core purpose in mind: How does the security analyst assess the NVD risk under scrutiny given the proper data, and how does one remedy or countermeasure the problem situation most scientifically and accurately to reach an optimally feasible solution in the least costly manner? The proposed quantitative approach-centered philosophy underlying the SM method, as opposed to the popular Confidentiality-Integrity-Availability-centered [1] conventional, descriptive, and qualitative routines resulting in non-numerical limited comparisons, is clarified and elaborated. What this proposed work demonstrates in summary is as follows:

A) A scalable SM procedure can be successfully applied to an acceptably representative target NVD covering a whole nation, although the said database may not be a unique or the only NVD taken from CVE, such as [nvd.nist.gov](https://nvd.nist.gov) [27–30].

B) SM is compared for the critical stage of the Risk Management to the EXCEL Solver's LP-feasible solutions, which although generating the identical OTRR  $\approx 0.2$  and OTRR  $\approx 0.3$  (in the article's example 2 comprising attackers and users in the data sets of Tables A4 and A5) will be outsmarted by the SM's cumulative change in the unknown vector solutions of  $LCM_{ij}$  (optimal) illustrated in Tables A12–A14 and B1–B4. This implies that outputs by various software-generated LP-feasible solutions other than those of the SM will yield higher investment costs to redeem the NVD problem under scrutiny to mitigate and reduce the overall bad

risk percentage to a desirable and tolerable percentage.

C) Economic conditions are of primary importance since without cost and benefit, the investments proposed following the risk management stage may not be justified by the national or commercial enterprises.

D) Once the tree diagram's limbs are pruned implying that the related threats have been  $\sim 100\%$  annulled to perfection following the risk management stage as a consequence of investments, the analyst can carry on reducing the risk percentage to new lows if the allocated budget permits.

E) Targeting critical contributions to the vast potential of Computer Theory and Engineering, this article carries a certain weight and responsibility as underlined in the synopsis of the IJCTE.org Special Issue (2) which it was written for. The dissemination of the know-how and application mode is also imperative. The proposed innovative and practical solution is shown to uniquely and timely contribute to the Cybersecurity Informatics Sciences and Computer Engineering at large.

## APPENDIX A

### A. SM Risk Assessment with Optimal Risk Management Solutions in Tables A1–A14 and Figs. A4–A15

The Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information security vulnerabilities and exposures or threats [27–30]. MITRE Corporation maintains the system, with funding from the National Cyber Security Division of the United States Department of Homeland Security. CVE is used by the Security Content Automation Protocol (SCAP), and CVE Identifiers (IDs) are listed on Content Automation Protocol, listed on MITRE's system as well as the U.S. National Vulnerability Database. Moreover, CVE provides an easy-to-use web interface for the vulnerability data. One can browse for vendors, products, and versions and view CVE entries, and vulnerabilities related to descriptive attributes as illustrated in Figs. A1–A3.

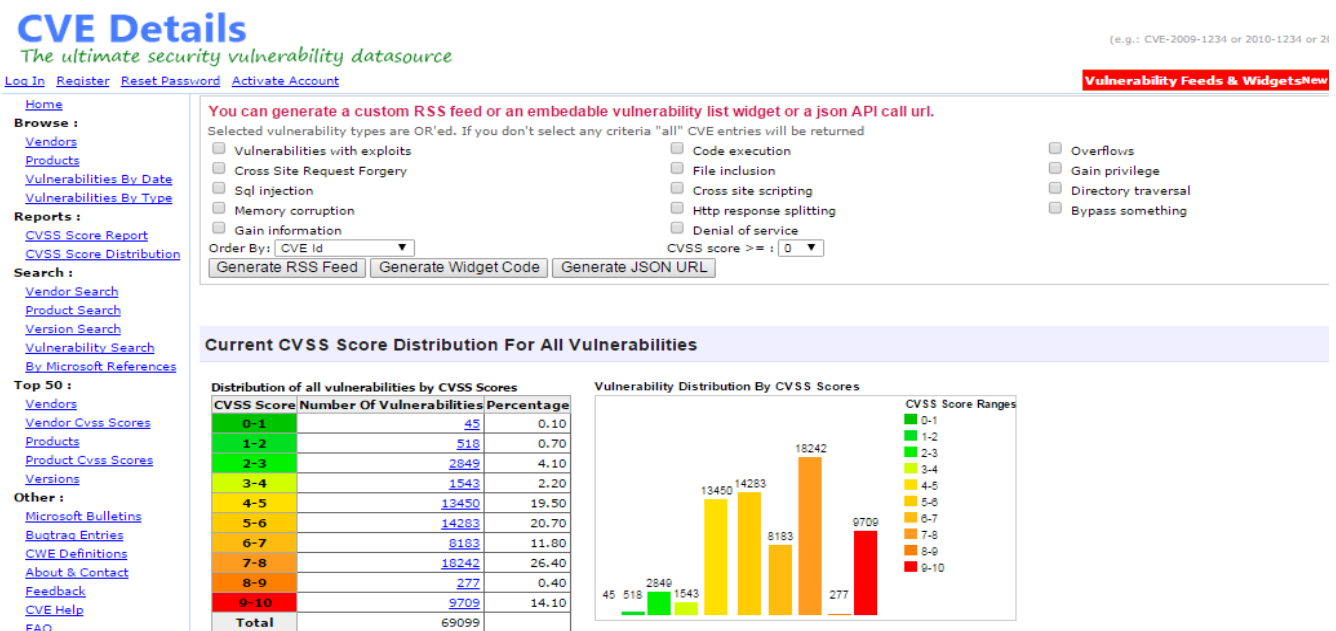


Fig. A1. Products and statistics screenshot for vulnerability count and severity levels (USG source <https://nvd.nist.gov/cvss.cfm?calculator&version=2>).

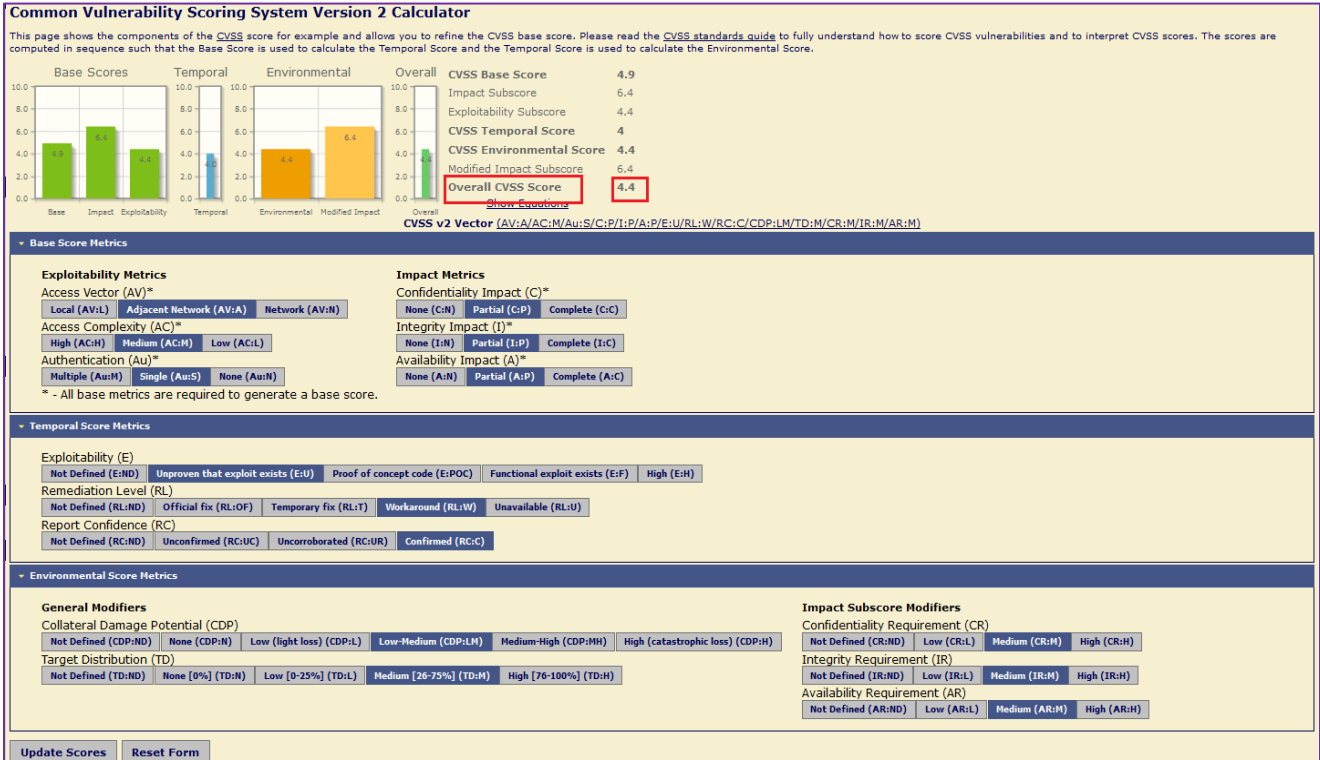


Fig. A2. NVD/CVSS case of descriptive input (CIA triplet) to demo how CVSS version 2 calculator operates (USG source <https://nvd.nist.gov/cvss.cfm?calculator&version=2>).

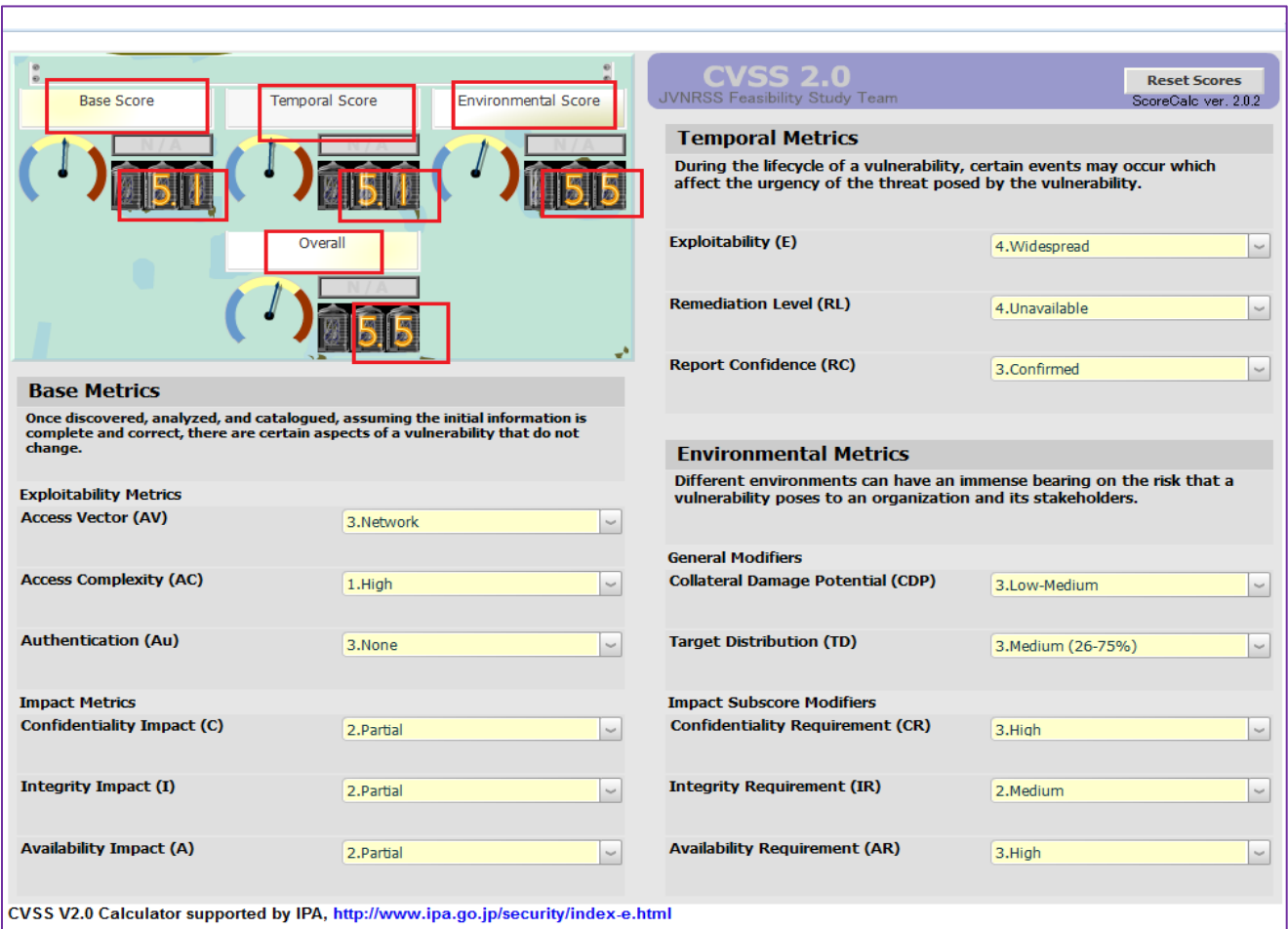


Fig. A3. NVD/CVSS version 2.0 (JVN) case of descriptive input (CIA triplet) to demo how its calculator operates (USG source <https://nvd.nist.gov/cvss.cfm?calculator&version=2>).

Tables A4–A14 and Figs. A4–A15 will show the quantitative route by which the SM algorithmic framework can be facilitated to arrive at a solution and will be justified

in detail owing to the tabulated spreadsheets of Tables A6 and A7 by the SM software for example 2, whereas example 1 solely covers input and output information in Tables A1–A3.

Table A1. SM Input data tabulated for example 1 cited in [5] to indicate how to mitigate risk

Vulnerability (V)	Threat (T)	Countermeasure (CM)	Lack of Countermeasure (LCM = 1 – CM)
V <sub>1</sub> = 0.35 (Internal Security Breach Only)	T <sub>11</sub> = 0.48 (Internal Abuse of Network Access)	CM <sub>11</sub> = 0.70 (Security Awareness Policy Training)	LCM <sub>11</sub> = 0.30 (by Subtraction)
	T <sub>12</sub> = 0.16 (System Penetration)	CM <sub>12</sub> = 0.42 (Smart Cards/Other One-Time Password Tokens)	LCM <sub>12</sub> = 0.30 (by Subtraction)
	T <sub>13</sub> = 0.32 (Denial of Service)	CM <sub>13</sub> = 0.97 (Firewalls)	LCM <sub>13</sub> = 0.03 (by Subtraction)
	T <sub>14</sub> = 0.04 (Financial/Telecom Fraud)	CM <sub>14</sub> = 0.80 (Security Audits)	LCM <sub>14</sub> = 0.20 (by Subtraction)
V <sub>2</sub> = 0.26 (External Security Breach Only)	T <sub>21</sub> = 0.32 (Denial of Service)	CM <sub>21</sub> = 0.35 (Public Key Infrastructure)	LCM <sub>21</sub> = 0.65 (by Subtraction)
	T <sub>22</sub> = 0.02 (Sabotage)	CM <sub>22</sub> = 0.35 (Intrusion Prevention Systems)	LCM <sub>22</sub> = 0.65 (by Subtraction)
	T <sub>23</sub> = 0.66 (Virus)	CM <sub>23</sub> = 0.96 (Anti-Virus Software)	LCM <sub>23</sub> = 0.04 (by Subtraction)
	T <sub>31</sub> = 0.32 (Unauthorized Access to Information)	CM <sub>31</sub> = 0.72 (Intrusion Detection Systems)	LCM <sub>31</sub> = 0.28 (by Subtraction)
V <sub>3</sub> = 0.39 (Both Internal and External Security Breaches Combined)	T <sub>32</sub> = 0.59 (Malicious Code)	CM <sub>32</sub> = 0.70 (Server-Based Control)	LCM <sub>32</sub> = 0.30 (by Subtraction)
	T <sub>33</sub> = 0.09 (Theft of Proprietary Information)	CM <sub>33</sub> = 0.46 (Encrypted Files)	LCM <sub>33</sub> = 0.54 (by Subtraction)

Table A2. Example 1 cited in Table A1 of [5] demonstrates how to reduce risk from ~26% to ~10% with SM with EXCEL Spreadsheet for a Cumulative Change of 90.52% to reach ~\$513.38 Investment Cost where the Breakeven Cost is \$5.67 per 1%, i.e. \$5.67 per 1% ≈ \$513.38 / 90.52%

Vulnerab.	Threat	CM & LCM	Res. Risk	CM & LCM	Res. Risk	Change	Cost	C = C * COST	per 1%
0.35	0.48	0.7	0.0504	1	0	0.3	\$170.10		\$5.67
	0.16	0.42	0.03248	0.58	0.03248	0	\$0.00		
	0.32	0.97	0.00336	0.97	0.00336	0	\$0.00		
	0.04	0.8	0.0028	0.8	0.0028	0	\$0.00		
0.26	0.22	0.35	0.03718	0.35	0.03718	0	\$0.00		
	0.02	0.35	0.00338	0.35	0.00338	0	\$0.00		
	0.76	0.96	0.007904	1	0	0.04	\$22.68		
0.39	0.32	0.72	0.034944	0.9852	0.00184704	0.2652	\$150.37		
	0.59	0.7	0.06903	0.148	0	0.3	\$170.10		
	0.09	0.46	0.018954	0	0	0	\$0.00		
		0.54	0.018954	0.54	0.018954				
		Total Risk	0.260432	Total Risk	0.10000104	0.9052	\$513.25		
		Percentage	26.04%	Percentage	10.00%				
BASE	SERVER	Final Risk	0.1041728	Final Risk	0.040000416		IMPROVED	SERVER	
Asset=	\$8000	ECL	\$833.38	ECL	\$320.00				
Criticality=	0.40			Delta ECL	-\$513.38				

Table A3. \*Referring to example 1's Table A1 input spreadsheet, this output following the preceding EXCEL Table A2 to SM's output Table A3 generates the following advice tips to mitigate the total risk from ~26% to ~10%: **1)** For the V<sub>1</sub>, the vulnerability of Internal Security Breach Only: Improve the Security Awareness Policy Training's CM from 70% to 100% by investing \$170 out of \$513 (the total investment cost) per \$8K lump sum for the asset of the university server. **2)** For the V<sub>2</sub>, the vulnerability of External Security Breach Only: Improve the Anti-Virus Software's CM from 96% to 100% by investing \$23 out of \$513 (the total investment cost) per \$8K lump sum for the asset of the university server. **3)** For the V<sub>3</sub>, the vulnerability of Both Internal and External Security Breaches Combined: Improve the Intrusion Detection Systems' CM from 72% to 98.54% by investing \$150 out of \$513 (the total investment cost) per \$8K lump sum for the asset of the university server. **4)** For the V<sub>3</sub>, the vulnerability of Both Internal and External Security Breaches Combined: Improve the Server-Based Control's CM from 70% to 99.99% by investing \$170 out of \$513 (the total investment cost) per \$8K lump sum for the asset of the university server. If the lump sum rises from \$8K to \$80K, each of the investment values proportionately rises 10-fold.

Vuln.	Threat	CM & LCM	Res. Risk	CM & LCM	Res Risk	Change	Opt Cost	Unit Cost	Final Cost	Advice
0.350000	0.480000	0.700000	0.050400	1.000000	0.000000	0.300000	\$170.13	\$170.00	\$170.00	Increase the CM capacity for threat "v1.t1" for the vulnerability of "v1" from 70.00% to 100.00% for an improvement of 30.00%.
	0.160000	0.420000	0.032480	0.580000	0.032480					
	0.320000	0.970000	0.003360	0.970000	0.003360					
	0.040000	0.800000	0.002800	0.800000	0.002800					
0.260000	0.220000	0.350000	0.037180	0.350000	0.037180					
	0.020000	0.350000	0.003380	0.350000	0.003380					
	0.760000	0.960000	0.007904	1.000000	0.000000	0.040000	\$22.68	\$23.00	\$23.00	Increase the CM capacity for threat "v2.t3" for the vulnerability of "v2" from 96.00% to 100.00% for an improvement of 4.00%.
0.390000	0.320000	0.720000	0.034944	0.985410	0.001821	0.265410	\$150.51	\$150.00	\$150.00	Increase the CM capacity for threat "v3.t1" for the vulnerability of "v3" from 72.00% to 98.54% for an improvement of 26.54%.
	0.590000	0.700000	0.069030	0.999890	0.000025	0.299890	\$170.06	\$170.00	\$170.00	Increase the CM capacity for threat "v3.t2" for the vulnerability of "v3" from 70.00% to 99.99% for an improvement of 29.99%.
	0.090000	0.460000	0.018954	0.540000	0.018954					
						Total Change	Total Cost	Break Even Cost	Total Final Cost	
						90.53%	\$513.38	\$5.67	\$513.00	

Criticality 0.40 Total Risk 0.260432 Percentage 26.043200 Final Risk 0.104173 ECL \$833.38

Capital Cost \$8,000.00 Total Threat Costs N/A

Show where you are in Security Meter

Optimize

Change Cost

Total Risk 0.100000 Percentage 10.000004 Final Risk 0.040000 ECL \$320.00 ECL Delta \$513.38

Change Unit Cost

Calculate Final Cost

Print Summary

Print Results Table

Table A4. The original data of 10 vulnerabilities and 40 threats modified from Common Vulnerabilities and Exposures, www.cvedetails.com of example 2

Vulnerability	Vulnerability Count	Probability	Threats	Threat Count
Buffer overflow (V <sub>1</sub> )	5466	0.218	Remote attacker (T <sub>1</sub> )	4090
				0.7483
			Local user (T <sub>2</sub> )	766
				0.1401
			User-assisted remote attacker (T <sub>3</sub> )	349
				0.0638
			Remote authenticated user (T <sub>4</sub> )	151
				0.0276
			Context-dependent attacker (T <sub>5</sub> )	110
				0.0202
Total Vulnerabilities (V <sub>1</sub> )				5466
Total Probability (V <sub>1</sub> )				1
Web Server (V <sub>2</sub> )	1135	0.0453	Remote attacker (T <sub>1</sub> )	877
				0.7727
			Remote user (T <sub>2</sub> )	134
				0.1180
			Local Users (T <sub>3</sub> )	54
				0.0476
			User-assisted remote attacker (T <sub>3</sub> )	2
				0.0018
			Remote authenticated user (T <sub>5</sub> )	68
				0.0599
Total Vulnerabilities (V <sub>2</sub> )				1135
Total Probability (V <sub>2</sub> )				1
JavaScript (V <sub>3</sub> )	178	0.0071	Remote attacker (T <sub>1</sub> )	169
				0.9495
			Local User (T <sub>2</sub> )	1
				0.0056
			User-assisted remote attacker (T <sub>3</sub> )	8
				0.0449
Total Vulnerabilities (V <sub>3</sub> )				178
Total Probability (V <sub>3</sub> )				1
Race condition (V <sub>4</sub> )	313	0.0125	Remote attacker (T <sub>1</sub> )	87
				0.2779
			Local User (T <sub>2</sub> )	214
				0.6838
			Physically proximate attacker (T <sub>3</sub> )	7
				0.0224
			Remote authenticated user (T <sub>4</sub> )	5
				0.0159
Total Vulnerabilities (V <sub>4</sub> )				313
Total Probability (V <sub>4</sub> )				1
Cross-Site (V <sub>5</sub> )	7274	0.2902	Remote attacker (T <sub>1</sub> )	6798
				0.9346
			Remote user (T <sub>2</sub> )	4
				0.0005
			Local Users (T <sub>3</sub> )	34
				0.0047
			User-assisted remote attacker (T <sub>4</sub> )	438
				0.0602
Total Vulnerabilities (V <sub>5</sub> )				7274
Total Probability (V <sub>5</sub> )				1
SQL Injection (V <sub>6</sub> )	5564	0.2219	Remote attacker (T <sub>1</sub> )	5321
				0.9563
			Local User (T <sub>2</sub> )	3
				0.0005
			User-assisted remote attacker (T <sub>3</sub> )	2
				0.0004
			Remote authenticated user (T <sub>4</sub> )	236
				0.0424
			Context-dependent attacker (T <sub>5</sub> )	2
				0.0004
Total Vulnerabilities (V <sub>6</sub> )				5564
Total Probability (V <sub>6</sub> )				1
File Inclusion (V <sub>7</sub> )	2060	0.0821	Remote attacker (T <sub>1</sub> )	2054
				0.9971
			Remote authenticated user (T <sub>2</sub> )	6
				0.0029
Total Vulnerabilities (V <sub>7</sub> )				2060
Total Probability (V <sub>7</sub> )				1
Format string (V <sub>8</sub> )	458	0.0183	Remote attacker (T <sub>1</sub> )	331
				0.7227
			Local User (T <sub>2</sub> )	80
				0.1747
			User-assisted remote attacker (T <sub>3</sub> )	22

			0.0480
	Remote authenticated user (T <sub>4</sub> )		16
			0.0349
	Context-dependent attacker (T <sub>5</sub> )		9
			0.0197
Total Vulnerabilities (V <sub>8</sub> )			458
Total Probability (V <sub>8</sub> )			1
Directory traversal (V <sub>9</sub> )	2267	Remote attacker (T <sub>1</sub> )	2092
	0.0904		0.9228
	Remote authenticated user (T <sub>2</sub> )		154
			0.0679
	Remote authenticated administrator (T <sub>3</sub> )		21
			0.0093
Total Vulnerabilities (V <sub>9</sub> )			2267
Total Probability (V <sub>9</sub> )			1
Untrusted search path (V <sub>10</sub> )	355	Remote attacker (T <sub>1</sub> )	43
	0.0142		0.1211
	Local user (T <sub>2</sub> )		309
			0.8704
	User-assisted remote attacker (T <sub>3</sub> )		2
			0.0057
Remote authenticated user (T <sub>4</sub> )		1	
			0.0028
Total Vulnerabilities (V <sub>10</sub> )			355
Total Probability (V <sub>10</sub> )			1
Grand Total		25070	

Table A5. Vulnerabilities merged (Attacker-User) 20 threats: (T<sub>1</sub>)' and (T<sub>2</sub>)' modified for V<sub>1</sub> to V<sub>10</sub> per original Table A4, www.cvedetails.com of example 2

Vulnerability	Vulnerability Count Probability	Threats	Threat Count
Buffer overflow (V <sub>1</sub> )	5466	Attacker (T <sub>1</sub> )'	4549
	0.218		0.8322
		User (T <sub>1</sub> )'	917
			0.1678
Total Vulnerabilities (V <sub>1</sub> )			5466
Total Probability (V <sub>1</sub> )			1
Web Server (V <sub>2</sub> )	1135	Attacker (T <sub>1</sub> )'	879
	0.0453		0.7745
		User (T <sub>1</sub> )'	256
			0.2255
Total Vulnerabilities (V <sub>2</sub> )			1135
Total Probability (V <sub>2</sub> )			1
JavaScript (V <sub>3</sub> )	178	Attacker (T <sub>1</sub> )'	177
	0.0071		0.9944
		User (T <sub>1</sub> )'	1
			0.0056
Total Vulnerabilities (V <sub>3</sub> )			178
Total Probability (V <sub>3</sub> )			1
Race condition (V <sub>4</sub> )	313	Attacker (T <sub>1</sub> )'	94
	0.0125		0.3003
		User (T <sub>1</sub> )'	219
			0.6997
Total Vulnerabilities (V <sub>4</sub> )			313
Total Probability (V <sub>4</sub> )			1
Cross-Site (V <sub>5</sub> )	7274	Attacker (T <sub>1</sub> )'	7236
	0.2902		0.9948
		User (T <sub>1</sub> )'	38
			0.0052
Total Vulnerabilities (V <sub>5</sub> )			7274
Total Probability (V <sub>5</sub> )			1
SQL Injection (V <sub>6</sub> )	5564	Attacker (T <sub>1</sub> )'	5325
	0.2219		0.9570
		User (T <sub>1</sub> )'	239
			0.0430
Total Vulnerabilities (V <sub>6</sub> )			5564
Total Probability (V <sub>6</sub> )			1
File Inclusion (V <sub>7</sub> )	2060	Attacker (T <sub>1</sub> )'	2054
	0.0821		0.9971
		User (T <sub>1</sub> )'	6
			0.0029
Total Vulnerabilities (V <sub>7</sub> )			2060
Total Probability (V <sub>7</sub> )			1
Format string (V <sub>8</sub> )	458	Attacker (T <sub>1</sub> )'	331
	0.0183		0.7227
		User (T <sub>1</sub> )'	127
			0.2773
Total Vulnerabilities (V <sub>8</sub> )			458

<b>Total Probability (V<sub>8</sub>)</b>		<b>1</b>
<b>Directory traversal (V<sub>9</sub>)</b>	<b>2267</b>	<b>2092</b>
	<b>0.0904</b>	<b>0.9228</b>
	<b>Attacker (T<sub>1</sub>)'</b>	
		175
		0.0772
<b>Total Vulnerabilities (V<sub>9</sub>)</b>		<b>2267</b>
<b>Total Probability (V<sub>9</sub>)</b>		<b>1</b>
<b>Untrusted search path(V<sub>10</sub>)</b>	<b>355</b>	<b>45</b>
	<b>0.0142</b>	<b>0.1268</b>
	<b>Attacker (T<sub>1</sub>)'</b>	
		310
		0.8732
<b>Total Vulnerabilities (V<sub>10</sub>)</b>		<b>355</b>
<b>Total Probability (V<sub>10</sub>)</b>		<b>1</b>
<b>Grand Total</b>		<b>25070</b>

Table A6. Risk assessment of the original Table A4's input data yielding TRR  $\approx 0.63$  by using EXCEL uniform random number generator RANDC()

<b>Vulnerability</b>	<b>Threat</b>	<b>Vulnerability *Threat</b>	<b>LCM from RANDC()</b>	<b>Residual Risk</b>
0.218	0.7483	0.1631294	0.372463	0.060759662
0.218	0.1401	0.0305418	0.668684	0.020422816
0.218	0.0638	0.0139084	0.342764	0.004767297
0.218	0.0276	0.0060168	0.778985	0.004686996
0.218	0.0202	0.0044036	0.669135	0.002946601
0.0453	0.7727	0.0350033	0.923419	0.032322715
0.0453	0.118	0.0053454	0.245885	0.001314355
0.0453	0.0476	0.0021563	0.711052	0.001533226
0.0453	0.0018	8.154E-05	0.948677	7.73551E-05
0.0453	0.0599	0.0027135	0.522707	0.00141835
0.0071	0.9495	0.0067415	0.299165	0.002016804
0.0071	0.0056	3.976E-05	0.082268	3.27097E-06
0.0071	0.0449	0.0003188	0.682852	0.000217686
0.0125	0.2779	0.0034738	0.457385	0.001588841
0.0125	0.6838	0.0085475	0.664137	0.005676712
0.0125	0.0224	0.00028	0.959153	0.000268563
0.0125	0.0159	0.0001988	0.955010	0.000189808
0.2902	0.9346	0.2712209	0.918751	0.249184568
0.2902	0.0005	0.0001451	0.154792	2.24603E-05
0.2902	0.0047	0.0013639	0.301131	0.000410724
0.2902	0.0602	0.01747	0.418016	0.007302748
0.2219	0.9563	0.212203	0.851019	0.18058886
0.2219	0.0005	0.000111	0.978707	0.000108587
0.2219	0.0004	8.876E-05	0.688031	6.10697E-05
0.2219	0.0424	0.0094086	0.277037	0.00260652
0.2219	0.0004	8.876E-05	0.261302	2.31932E-05
0.0821	0.9971	0.0818619	0.083686	0.006850667
0.0821	0.0029	0.0002381	0.637028	0.00015167
0.0183	0.7227	0.0132254	0.688656	0.009107754
0.0183	0.1747	0.003197	0.583031	0.001863955
0.0183	0.048	0.0008784	0.453523	0.000398374
0.0183	0.0349	0.0006387	0.576169	0.000367982
0.0183	0.0197	0.0003605	0.807661	0.00029117
0.0904	0.9228	0.0834211	0.216282	0.018042476
0.0904	0.0679	0.0061382	0.311403	0.001911442
0.0904	0.0093	0.0008407	0.11518	9.68339E-05
0.0142	0.1211	0.0017196	0.435736	0.000749301
0.0142	0.8704	0.0123597	0.737971	0.009121088
0.0142	0.0057	8.094E-05	0.812291	6.57468E-05
0.0142	0.0028	3.976E-05	0.645450	2.56631E-05
<b>SUMS→</b>		<b>1.00</b>		<b>TRR <math>\approx 0.630</math></b>

Table A7. Risk assessment of the merged Table A5's input data yielding TRR  $\approx 0.629$  by EXCEL uniform random number generator RANDC()

<b>Vulnerability</b>	<b>Threat</b>	<b>Vulnerability *Threat</b>	<b>LCM from RANDC()</b>	<b>Residual Risk</b>
0.218	0.8322	0.1814196	0.91807948	0.166557612
0.218	0.1678	0.0365804	0.25207774	0.009221105
0.0453	0.7745	0.03508485	0.03116043	0.001093259
0.0453	0.2255	0.01021515	0.89175789	0.009109441
0.0071	0.9944	0.00706024	0.63192018	0.004461508
0.0071	0.0056	0.00003976	0.26267045	1.04438E-05
0.0125	0.3003	0.00375375	0.56656422	0.00212674
0.0125	0.6997	0.00874625	0.07867729	0.000688131
0.2902	0.9948	0.28869096	0.82082296	0.236964169
0.2902	0.0052	0.00150904	0.97187958	0.001466605
0.2219	0.957	0.2123583	0.43843319	0.093104927
0.2219	0.043	0.0095417	0.62435882	0.005957445
0.0821	0.9971	0.08186191	0.59904621	0.049039067
0.0821	0.0029	0.00023809	0.27987393	6.66352E-05
0.0183	0.7227	0.01322541	0.99667400	0.013181422
0.0183	0.2773	0.00507459	0.15582265	0.000790736
0.0904	0.9228	0.08342112	0.30387833	0.025349871



0.0904	0.0772	0.00697888	0.08006754	0.000558782
0.0142	0.1268	0.00180056	0.77186225	0.001389784
0.0142	0.8732	0.01239944	0.67437953	0.008361929
<b>SUMS→</b>		<b>1.00</b>		<b>TRR=0.629</b>

Next, enter step by step into the SM algorithmic software, the 20 ( $= 10 \times 2$ ) vulnerability-threat pairs for the merged input data set in Table A5 from the original Table A4 with 40 threat counts through Fig. A4 to set the stage, and counting 10 pairs of data entries through Figs. A5–A14: i) Table A8 (Risk Assessment interface solution  $TRR \approx 0.629$  to be

optimized to  $OTRR = 0.2$ ). ii) Table A9 (Risk Management of Table A8 to be optimized with cost-optimal investment advice rows), iii) Table A10 (Risk Assessment interface solution  $TRR = 0.629$  to be optimized to  $OTRR = 0.3$ ), and iv) Table A11 (Table A10 to be cost-optimized with investment-savvy advice rows).

Name of Vulnerability	Actual	Threats
1 v1	0.218	2
2 v2	0.0453	2
3 v3	0.0071	2
4 v4	0.0125	2
5 v5	0.2902	2
6 v6	0.2219	2
7 v7	0.0821	2
8 v8	0.0183	2
9 v9	0.0904	2
10 v10	0.0142	2

Fig. A4. The data entry for the merged Table A5 from the original Table A4 to enter the SM input toward the TRR assessment result.

Name of Threat	Threat	LCM
1 v1.t1	0.8322	0.9181
2 v1.t2	0.1678	0.2521

Fig. A5.  $V_1$  input vulnerability-threat pairs  $V_1 T_1'$  and  $V_1 T_2'$  with  $LCM_{ij}$  from EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
1 v2.t1	0.7745	0.0312
2 v2.t2	0.2255	0.8918

Fig. A6.  $V_2$  input vulnerability-threat pairs  $V_2 T_1'$  and  $V_2 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
1 v3.t1	0.9944	0.6319
2 v3.t2	0.0056	0.2627

Fig. A7.  $V_3$  input vulnerability-threat pairs  $V_3 T_1'$  and  $V_3 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() from Table A7.

Name of Threat	Threat	LCM
1 v4.t1	0.3003	0.5666
2 v4.t2	0.6997	0.0787

Fig. A8.  $V_4$  input vulnerability-threat pairs  $V_4 T_1'$  and  $V_4 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.



Name of Threat	Threat	LCM
v5.t1	0.9948	0.8208
v5.t2	0.0052	0.9719

Fig. A9.  $V_5$  input vulnerability-threat pairs  $V_5 T_1'$  and  $V_5 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
v6.t1	0.9570	0.4384
v6.t2	0.043	0.6244

Fig. A10.  $V_6$  input vulnerability-threat pairs  $V_6 T_1'$  and  $V_6 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
v7.t1	0.9971	0.5990
v7.t2	0.0029	0.2799

Fig. A11.  $V_7$  input vulnerability-threat pairs  $V_7 T_1'$  and  $V_7 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
v8.t1	0.7227	0.9967
v8.t2	0.2773	0.1558

Fig. A12.  $V_8$  input vulnerability-threat pairs  $V_8 T_1'$  and  $V_8 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
v9.t1	0.9228	0.3039
v9.t2	0.0772	0.0801

Fig. A13.  $V_9$  input vulnerability-threat pairs  $V_9 T_1'$  and  $V_9 T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Name of Threat	Threat	LCM
v10.t1	0.1268	0.7719
v10.t2	0.8732	0.6744

Fig. A14.  $V_{10}$  input vulnerability-threat pairs  $V_{10} T_1'$  and  $V_{10} T_2'$  with  $LCM_{ij}$  of EXCEL uniform random generator: RANDC() by Table A7.

Criticality	1.0
Equipment cost	1000.0
Production cost	0.0
Repair cost to mitigate =	\$629.49

Fig. A15. Final TRR=0.629 following Figs. A4–A14 to get Expected Cost of Loss (ECL)  $\approx$  \$629.49 (Table A7) out of an Equipment Cost = \$1000.

Table A8. SM solution of  $10 \times 2 = 20$  residual risks cumulates to TRR  $\approx 0.629$  ready to optimize to OTRR = 0.2 (Eq. 49) for risk management.

Vuln.	Vuln. Risk	Threat	Threat Risk	LCM	Res. Risk	Post Res. Risk	Post Vuln. Risk	>
v1	0.218000	v1.t1	0.832200	0.918100	0.166561	0.26		
		v1.t2	0.167800	0.252100	0.009222	0.01	0.279246	!
v2	0.045300	v2.t1	0.774500	0.031200	0.001095	0.00		
		v2.t2	0.225500	0.891800	0.009110	0.01	0.016211	
v3	0.007100	v3.t1	0.994400	0.631900	0.004461	0.01		
		v3.t2	0.005600	0.262700	0.000010	0.00	0.007104	!
v4	0.012500	v4.t1	0.300300	0.566600	0.002127	0.00		
		v4.t2	0.699700	0.078700	0.000688	0.00	0.004472	
v5	0.290200	v5.t1	0.994800	0.820800	0.236958	0.38		
		v5.t2	0.005200	0.971900	0.001467	0.00	0.378757	!
v6	0.221900	v6.t1	0.957000	0.438400	0.093098	0.15		
		v6.t2	0.043000	0.624400	0.005958	0.01	0.157358	
v7	0.082100	v7.t1	0.997100	0.599000	0.049035	0.08		
		v7.t2	0.002900	0.279900	0.000067	0.00	0.078002	
v8	0.018300	v8.t1	0.722700	0.996700	0.013182	0.02		
		v8.t2	0.277300	0.155800	0.000791	0.00	0.022196	!
v9	0.090400	v9.t1	0.922800	0.303900	0.025352	0.04		
		v9.t2	0.077200	0.080100	0.000559	0.00	0.041161	
v10	0.014200	v10.t1	0.126800	0.771900	0.001390	0.00		
		v10.t2	0.873200	0.674400	0.008362	0.01	0.015492	!

Criticality 1.00  
Capital Cost \$1,000.00  
Total Threat Costs N/A  
Res-Risk \* Criticality 0.629492  
Total Res-Risk 0.629492  
Expected Cost of Loss \$629.49

Optimize

Input  
Enter target Risk (must be less than 0.629492)  
0.20  
OK Cancel

Table A9. The SM risk management solution to optimize Table A8 from TRR  $\approx 0.629$  to OTRR  $\approx 0.2$  with three investment actions advised\*.

Vuln.	Threat	CM & LCM	Res. Risk	CM & LCM	Res Risk	Change	Opt Cost	Unit Cost	Final Cost	Advice
0.218000	0.832200	0.081900	0.630264	0.369736	0.067077	0.548364	\$130.31	\$130.00	\$130.00	Increase the CM capacity for threat "v1.t1" for the vulnerability of "v1" from 8.19% to 63.03% for an improvement of 54.84%.
	0.167800	0.747900	0.747900	0.747900						
	0.252100	0.009222	0.252100	0.009222						
0.045300	0.774500	0.968800	0.968800							
	0.031200	0.001095	0.031200	0.001095						
	0.225500	0.108200	0.108200							
	0.891800	0.009110	0.891800	0.009110						
0.007100	0.994400	0.368100	0.368100							
	0.631900	0.004461	0.631900	0.004461						
	0.005600	0.737300	0.737300							
	0.262700	0.000010	0.262700	0.000010						
0.012500	0.300300	0.433400	0.433400							
	0.566600	0.002127	0.566600	0.002127						
	0.699700	0.921300	0.921300							
	0.078700	0.000688	0.078700	0.000688						
0.290200	0.994800	0.179200	0.999835	0.999835		0.820635	\$195.01	\$195.00	\$195.00	Increase the CM capacity for threat "v5.t1" for the vulnerability of "v5" from 17.92% to 99.98% for an improvement of 82.06%.
	0.820800	0.236958	0.000165	0.000048						
	0.005200	0.028100	0.028100							
0.221900	0.971900	0.001467	0.971900	0.001467		0.438400	\$104.18	\$104.00	\$104.00	Increase the CM capacity for threat "v6.t1" for the vulnerability of "v6" from 56.16% to 100.00% for an improvement of 43.84%.
	0.561600	1.000000	1.000000							
	0.438400	0.093098	0.000000	0.000000						
	0.375600	0.375600								

Criticality 1.00  
Capital Cost \$1,000.00  
Total Threat Costs N/A

Total Risk 0.629492  
Percentage 62.949171  
Final Risk 0.629492  
ECL \$629.49

Show where you are in Security Meter  
Optimize

Change Cost

Total Risk 0.200000  
Percentage 19.999998  
Final Risk 0.200000  
ECL \$200.00  
ECL Delta \$429.49

Change Unit Cost  
Calculate Final Cost  
Print Summary  
Print Results Table

\*1) Increase the CM capacity for threat  $V_1T_1$  for Vulnerability  $V_1$ 's Buffer Overflow Attacker  $T_1$  from 8.19% to 63.03% for an improvement of 54.84% while investing  $\leq \$130.31$  (out of \$429.49 total investment budget) for \$1K capital cost. 2) Increase the CM capacity for threat  $V_5T_1$  for Vulnerability  $V_5$ 's Cross-Site Attacker  $T_1$  from 17.92% to 99.98% for an improvement of 82.06% while investing  $\leq \$195.01$  (out of \$429.49 total investment budget) for \$1K capital costs. 3) Increase the CM capacity for threat  $V_6T_1$  for Vulnerability  $V_6$ 's SQL Injection Attacker  $T_1$  from 56.16% to 100% for an improvement of 43.59% while investing  $\leq \$104$  (out of \$429.49 total investment budget) for \$1K capital costs. Results are  $1 \times 10^6$ -fold scalable if in case \$1K  $\rightarrow$  \$1 Billion of total assets for the entire National Vulnerability Database as an example.

Table A10. SM solution of  $10 \times 2 = 20$  residual risks cumulate to TRR  $\approx 0.629$  ready to optimize to OTRR = 0.3 (Eq. (50) for risk management.

Vuln.	Vuln. Risk	Threat	Threat Risk	LCM	Res. Risk	Post Res. Risk	Post Vuln. Risk	>
v1	0.218000	v1.t1	0.832200	0.918100	0.166561	0.26		
		v1.t2	0.167800	0.252100	0.009222	0.01	0.279246	!
v2	0.045300	v2.t1	0.774500	0.031200	0.001095	0.00		
		v2.t2	0.225500	0.891800	0.009110	0.01	0.016211	
v3	0.007100	v3.t1	0.994400	0.631900	0.004461	0.01		
		v3.t2	0.005600	0.262700	0.000010	0.00	0.007104	!
v4	0.012500	v4.t1	0.300300	0.566600	0.002127	0.00		
		v4.t2	0.699700	0.078700	0.000688	0.00	0.004472	
v5	0.290200	v5.t1	0.994800	0.820800	0.236958	0.38		
		v5.t2	0.005200	0.971900	0.001467	0.00	0.378757	!
v6	0.221900	v6.t1	0.957000	0.438400	0.093098	0.15		
		v6.t2	0.043000	0.624400	0.005958	0.01	0.157358	
v7	0.082100	v7.t1	0.997100	0.599000	0.049035	0.08		
		v7.t2	0.002900	0.279900	0.000067	0.00	0.078002	
v8	0.018300	v8.t1	0.722700	0.996700	0.013182	0.02		
		v8.t2	0.277300	0.155800	0.000791	0.00	0.022196	!
v9	0.090400	v9.t1	0.922800	0.303900	0.025352	0.04		
		v9.t2	0.077200	0.080100	0.000559	0.00	0.041161	
v10	0.014200	v10.t1	0.126800	0.771900	0.001390	0.00		
		v10.t2	0.873200	0.674400	0.008362	0.01	0.015492	!

Criticality 1.00  
Capital Cost \$1,000.00  
Total Threat Costs N/A  
Res-Risk \* Criticality 0.629492  
Total Res-Risk 0.629492  
Expected Cost of Loss \$629.49

Optimize

Input  
Enter target Risk (must be less than 0.629492)  
0.30  
OK Cancel

Table A11. SM risk management to optimize Table A10 from  $TR \approx 0.629$  to optimize to  $OTRR \approx 0.3$  with two investment actions advised\*.

Vuln.	Threat	CM & LCM	Res. Risk	CM & LCM	Res Risk	Change	Opt Cost	Unit Cost	Final Cost	Advice
		0.252100	0.009222	0.252100	0.009222					
0.045300	0.774500	0.968800		0.968800						
		0.031200	0.001095	0.031200	0.001095					
	0.225500	0.108200		0.108200						
		0.891800	0.009110	0.891800	0.009110					
0.007100	0.994400	0.368100		0.368100						
		0.631900	0.004461	0.631900	0.004461					
	0.005600	0.737300		0.737300						
		0.262700	0.000010	0.262700	0.000010					
0.012500	0.300300	0.433400		0.433400						
		0.566600	0.002127	0.566600	0.002127					
	0.699700	0.921300		0.921300						
		0.078700	0.000688	0.078700	0.000688					
0.290200	0.994800	0.179200		0.999853		0.820653	\$215.18	\$215.00	\$215.00	Increase the CM capacity for threat "v5.t1" for the vulnerability of "v5" from 17.92% to 99.99% for an improvement of 82.07%.
		0.820800	0.236958	0.000147	0.000043					
	0.005200	0.028100		0.028100						
		0.971900	0.001467	0.971900	0.001467					
0.221900	0.957000	0.561600		0.997546		0.435946	\$114.31	\$110.00	\$110.00	Increase the CM capacity for threat "v6.t1" for the vulnerability of "v6" from 56.16% to 99.75% for an improvement of 43.59%.
		0.438400	0.093098	0.002454	0.000521					
	0.043000	0.375600		0.375600						

Criticality 1.00 Total Risk 0.629492  
Capital Cost \$1,000.00 Percentage 62.949171  
Total Threat Costs N/A Final Risk 0.629492  
ECL \$629.49

Show where you are in Security Meter  
Optimize

Change Unit Cost  
Change Cost

Total Risk 0.300000  
Percentage 30.000000  
Final Risk 0.300000  
ECL \$300.00  
ECL Delta \$329.49

Change Unit Cost  
Calculate Final Cost  
Print Summary  
Print Results Table

\*1) Increase the CM capacity for threat  $V_5T_1$  for Vulnerability  $V_5$ 's Cross-Site Attacker  $T_5$ , from 17.92% to 99.99% for an improvement of 82.08% while investing  $\leq \$215.18$  (out of a total investment cost of \$329.49) for \$1K capital costs. 2) Increase the CM capacity for threat  $V_6T_1$  for Vulnerability  $V_6$ 's SQL Injection  $T_6$  from 56.16% to 99.75% for an improvement of 43.59% while investing  $\leq \$114.3$  (out of a total investment cost of \$329.49) for \$1K capital costs. Results are  $1 \times 10^6$ -fold scalable if in case  $\$1K \rightarrow \$1$  Billion of total assets for the entire National Vulnerability Database as an example.

Table A12. SM risk management of merged threat data from Tables A5 and A7 where Eq. (49)  $\rightarrow$  OTRR (Optimized)  $\approx 0.2$ ; Cum.  $\Delta\% \approx 181\%$ 

Vulnerability	Threat	Vulnerability*Threat	LCM (SM)	Residual Risk	Difference from Original Input LCM Vector
0.218	0.8322	0.1814196	0.369736	0.067077176	0.548364
0.218	0.1678	0.0365804	0.25207774	0.009221105	
0.0453	0.7745	0.03508485	0.03116043	0.001093259	
0.0453	0.2255	0.01021515	0.89175789	0.009109441	
0.0071	0.9944	0.00706024	0.63192018	0.004461508	
0.0071	0.0056	0.00003976	0.26267045	1.04438E-05	
0.0125	0.3003	0.00375375	0.56656422	0.00212674	
0.0125	0.6997	0.00874625	0.07867729	0.000688131	
0.2902	0.9948	0.28869096	0.000165	4.24376E-05	0.820635
0.2902	0.0052	0.00150904	0.97187958	0.001466605	
0.2219	0.957	0.2123583	0.000001	0.000521127	0.438400
0.2219	0.043	0.0095417	0.62435882	0.005957445	
0.0821	0.9971	0.08186191	0.59904621	0.049039067	
0.0821	0.0029	0.00023809	0.27987393	6.66352E-05	
0.0183	0.7227	0.01322541	0.996674	0.013181422	
0.0183	0.2773	0.00507459	0.15582265	0.000790736	
0.0904	0.9228	0.08342112	0.30387833	0.025349871	
0.0904	0.0772	0.00697888	0.08006754	0.000558782	
0.0142	0.1268	0.00180056	0.77186225	0.001389784	
0.0142	0.8732	0.01239944	0.67437953	0.008361929	
SUMS $\rightarrow$		1.00	OTRR = 0.20		1.807 (=180.7%) $\approx$ Cum. $\Delta\%$

Table A13. SM risk management of merged threat data from Tables A5 and A7 where Eq. (50)  $\rightarrow$  OTRR (Optimized)  $\approx 0.3$ ; Cum.  $\Delta\% \approx 126\%$ .

Vulnerability	Threat	Vulnerability*Threat	LCM (SM)	Residual Risk	Difference from Original Input LCM Vector
0.218	0.8322	0.18142	0.9180795	0.055709061	
0.218	0.1678	0.03658	0.2520777	0.009221105	
0.0453	0.7745	0.03508	0.0311604	0.001093259	
0.0453	0.2255	0.01022	0.8917579	0.009109441	
0.0071	0.9944	0.00706	0.6319202	0.004461508	
0.0071	0.0056	4E-05	0.2626705	1.04438E-05	
0.0125	0.3003	0.00375	0.5665642	0.00212674	
0.0125	0.6997	0.00875	0.0786773	0.000688131	
0.2902	0.9948	0.28869	0.000147	0.055709053	0.820676
0.2902	0.0052	0.00151	0.9718796	0.001466605	
0.2219	0.957	0.21236	0.002454	0.055709036	0.435946
0.2219	0.043	0.00954	0.6243588	0.005957445	
0.0821	0.9971	0.08186	0.5990462	0.049039067	
0.0821	0.0029	0.00024	0.2798739	6.66352E-05	
0.0183	0.7227	0.01323	0.996674	0.013181422	
0.0183	0.2773	0.00507	0.1558227	0.000790736	
0.0904	0.9228	0.08342	0.3038783	0.025349871	
0.0904	0.0772	0.00698	0.0800675	0.000558782	
0.0142	0.1268	0.0018	0.7718623	0.001389784	
0.0142	0.8732	0.0124	0.6743795	0.008361929	
SUMS $\rightarrow$		1.00	OTRR = 0.30		1.257 (=125.7%) $\approx$ Cum. $\Delta\%$

### B. Summary of Tables A6–A13 All Tabulated in Table A14

Regarding example 2, in following Table A4 (original number of 40 counts of varying threats per 10 vulnerabilities) and Table A5 (merged equivalent simple  $10 \times 2$  topology for 20 threat counts), the author tabulated EXCEL spreadsheets of Tables A6 and A7, resulting from Figs. A4–A15, for input and output values. Regarding the SM's OTRR target of 20%, the SM risk assessment of Table A8 was followed by Table A9, where the initially undesirable risk of  $TRR \approx 62.95\%$  was mitigated to a desirable  $OTRR \approx 20\%$  in observance of Eqs. (5)–(49).

This implies that there is no better feasible solution minimizing the solution vector,  $LCM_{ij}$  (optimal), compared to that of Security-Meter's LP-feasible solution governed by the previous Eqs. (1)–(49). Consequently, regarding the SM's OTRR target of 30%; the SM risk assessment of Table A10, followed by Table A11 was undertaken where the initially undesirable risk of  $TRR \approx 62.95\%$  was mitigated to a desirable  $OTRR \approx 30\%$  in observance of Eqs. (5)–(50) by skipping Eq. (49). This implies that there is no better feasible solution minimizing the solution vector,  $LCM_{ij}$ , compared to that of Security-Meter's LP-feasible solution governed by the previous Eqs. (1)–(50). The following itemized investments and sub-investments of  $\$629.49 - \$300 = \$329.49$  and  $\$629.49 - \$200 = \$429.49$  must be carried out for the NVD respectively, whose data was collected from [www.cvedetails.com](http://www.cvedetails.com) given the Objective Function, Min LOSS, per Eqs. (5)–(49) and (5)–(50) by skipping Eq. (49):

**A)** For Risk Mitigation down to  $\sim 20\%$  from  $\sim 62.9\%$  per Table A9:

i) Regarding the  $CM_{11}$  of  $V_1T_1$ ' (Buffer Overflow Attacker of Table A4): Invest for a change of 54.84% on  $CM_{11}$  from 8.1% to 63.63%. Note,  $\Delta ECL / (\text{Total \% Change})$  gives the "Breakeven cost per 1%" =  $\$429.49 / 180.74\% \approx \$2.376$  per 1%. If the 54.84% change for  $CM_{11}$ (Buffer Overflow) is multiplied by  $\$2.376$  (Breakeven Cost), one gets  $\approx \$130.31$  (Opt. Cost) as in column 8 of Table A9.

ii) Regarding the  $CM_{51}$  of  $V_5T_1$ ' (Cross-Site Attacker of Table A4): Invest for a change of 82.06% on  $CM_{51}$  from 17.92% to 99.98%. Note,  $\Delta ECL / (\text{Total \% Change})$  gives the "Breakeven cost per 1%" =  $\$429.49 / 180.74\% \approx \$2.376$ . If the 82.06% change for  $CM_{51}$  (Cross-Site) is multiplied by  $\$2.376$  (Breakeven Cost), one gets  $\approx \$195.01$  (Opt. Cost) as in column 8 of Table A9.

iii) Regarding the  $CM_{61}$  of  $V_6T_1$ ' (SQL Injection Attacker of Table A4): Invest for a change of 43.84% on  $CM_{61}$  from 56.16% to 100%. Note,  $\Delta ECL / (\text{Total \% Change})$  gives the "Breakeven cost per 1%" =  $\$429.49 / 180.74\% \approx \$2.376$ . If the 43.84% change for  $CM_{61}$  (SQL Injection) is multiplied by  $\$2.376$  (Breakeven Cost), one gets  $\approx \$104.18$  as in column 8 of Table A9.

**B)** For Risk Mitigation down to  $\sim 30\%$  from  $\sim 62.9\%$  per Table A10, similarly:

i) Regarding the  $CM_{51}$  of  $V_5T_1$ ' (Cross-Site Attacker of Table A5): Invest for 82.07% on  $CM_{51}$  from 17.92% to 99.99%. Note,  $\Delta ECL / (\text{Total \% Change})$  gives the "Breakeven cost per 1%" =  $\$329.49 / 125.66\% \approx \$2.622$ . If 82.07% change for  $CM_{51}$  (Cross-Site) is multiplied by  $\$2.622$ . (Breakeven Cost), one gets  $\approx \$215.18$  as in column 8

of Table A11.

ii) Regarding the  $CM_{61}$  of  $V_6T_1$ ' (SQL Injection Attacker of Table A2): Invest for 43.59% on  $CM_{61}$  from 56.16% to 99.75%. Note,  $\Delta ECL / (\text{Total \% Change})$  gives the "Breakeven cost per 1%" =  $\$329.49 / 125.66\% \approx \$2.622$ . If the 43.59% change for  $CM_{61}$  (SQL Injection) is multiplied by  $\$2.622$  (Breakeven Cost), one gets  $\approx \$114.31$  as in column 8 of Table A11.

To realize a back substitution from the merged Table A5 regressing to the original Table A4 in example 2 for all vulnerabilities and threats, and then, to interpret the risk management results, observe the following:

**A')** Risk Mitigation to  $\sim 20\%$  to break down the threats:

i) a) Buffer Overflow Remote Attacker ( $V_1T_1$ ) of Table A4 ( $4090/4549 = 89.9\%$  of all Attacker types) deserves  $0.899 \times \$130.31$  (investment cost):  $\$117.16$

i) b) Buffer Overflow User-Assisted Remote Attacker ( $V_1T_3$ ) of Table A4 ( $349/4549 = 7.67\%$  of all Attackers types) deserves  $0.0767 \times \$130.31$  (investment cost):  $\$9.99$ .

i) c) Buffer Overflow Context-Dependent Attacker ( $V_1T_5$ ) of Table A4 ( $110/4549 = 2.43\%$  of all Attacker types) deserves  $0.0242 \times \$130.31$  (investment cost):  $\$3.16$ .

Recap of  $V_1$ 's three Threat-Itemized Sub-Investments:  $\$117.16 + \$9.99 + \$3.16 = \$130.31$ , identical to  $\$130.31$  in Table A9 (row 1, column 8) regarding  $V_1$ .

ii) a) Cross-Site Remote Attacker ( $V_5T_1$ ) of Table A4,  $6798/7236 = (93.95\%$  of all Attacker types) deserves  $0.9395 \times \$195.01$  (investment cost):  $\$183.2$ .

ii) b) Cross-Site User-Assisted Remote Attacker ( $V_5T_4$ ) of Table 4, ( $438/7236 = 6.05\%$  of all Attacker types) deserves  $0.0605 \times \$195.01$  (investment cost):  $\$11.81$ .

Recap of  $V_5$ 's two Threat-Itemized Investments:  $\$183.2 + \$11.81 = \$195.01$ , identical to  $\$195.01$  in Table A9 (row 17, column 8) regarding  $V_5$ .

iii) a) SQL Injection Remote Attacker ( $V_6T_1$ ) of Table A4 ( $5321/5325 = 99.92\%$  of all Attackers types) deserves  $0.9992 \times \$104.18$  (investment cost):  $\$104.1$ .

iii) b) SQL Injection User-Assisted Remote Attacker ( $V_6T_3$ ) of Table A4 ( $2/5325 = 0.04\%$  of all Attackers types) deserves  $0.00038 \times \$104.18$  (investment cost):  $\$0.039$ .

iii) c) SQL Injection Context-dependent Attacker ( $V_6T_5$ ) of Table A4 ( $2/5325 = 0.04\%$  of all Attackers types) deserves  $0.00038 \times \$104.18$  (investment cost):  $\$0.039$ .

Recap of  $V_6$ 's three Threat-Itemized Investments:  $\$104.1 + \$0.039 + \$0.039 = \$104.18$ , equal to  $\$104.18$  in Table A9 (row 20, column 8) regarding  $V_6$ .

**B')** Risk Mitigation to  $\sim 30\%$  to break down the threats:

i) a) Cross-Site Remote Attacker ( $V_5T_1$ ) of Table A4,  $6798/7236 = (93.95\%$  of all Attacker types) deserves  $0.9395 \times \$215.18$  (investment cost):  $\$202.16$ .

i) b) Cross-Site User-Assisted Remote Attacker ( $V_5T_4$ ) of Table A4, ( $438/7236 = 6.05\%$  of all Attacker types) deserves  $0.0605 \times \$215.08$  (investment cost):  $\$13.02$ .

Recap of  $V_5$ 's two Threat-Itemized Investments:  $\$202.16 + \$13.02 = \$215.18$ , which is identical to  $\$215.18$  in Table A11 (row 14, column 8) regarding  $V_5$ .

ii) a) SQL Injection Remote Attacker ( $V_6T_1$ ) of Table A4 ( $5321/5325 = 99.92\%$  of all Attackers types) deserves  $0.9992 \times \$114.31$  (investment cost):  $\$114.22$

ii) b) SQL Injection User-Assisted Remote Attacker ( $V_6T_3$ ) of Table A4 ( $2/5325 = 0.04\%$  of all Attackers types) deserves

$$0.00038 \times \$114.31 \text{ (investment cost): } \$0.043$$

ii) c) SQL Injection Context-dependent Attacker ( $V_6T_5$ ) of Table A4 ( $2/5325 = 0.04\%$  of all Attackers types) deserves  $0.00038 \times \$114.31$  (investment cost): \$0.043.

Recap of  $V_6$ 's three Threat-Itemized Investments:  $\$114.22 + \$0.043 + \$0.043 = \$114.31$ , identical to  $\$114.31$  in Table A11 (row 18, column 8) regarding  $V_6$ .

Then, Table A12 will display the SM Risk Management of Merged Threat Data from Table A5 where the constraint of Eq. (49)  $\rightarrow$  OTRR (Optimized TRR)  $\approx 0.2$  yielding (Cum.  $\Delta$  %)  $\approx 181\%$ .

Next, Table A13 will serve the same purpose for the

constraint of Eq. (50)  $\rightarrow$  OTRR (Optimized TRR)  $\approx 0.3$  yielding (Cum.  $\Delta$  %)  $\approx 126\%$ . Both of these Cum.  $\Delta$  % results of APPENDIX A of the Security Meter algorithm are less than those accumulated by an alternate EXCEL (XL) Solver with slightly different LP-feasible solutions in Appendix B.

The following comprehensive and all-inclusive Table A14 tabulates and computationally verifies the content of this subsection for a follow-up from Appendix A's input and output Tables A4–A13 and Figs A4–A14. This constitutes the NVD's example 2 of initial Table A4 which got merged to Table A5, which was summarized in Tables A12 and A13, and culminated in Table A14 spreadsheet for  $OTRR = 0.2, 0.3$ .

Table A14. SM tabulation of example 2 from Tables A4 to A5 and A6 to A13 with  $\text{Cum.}\Delta\% \approx 181\%(\text{O2})$  for Eq. (49) and  $\text{Cum.}\Delta\% \approx 126\%(\text{J9})$  for Eq. (50)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	SUMMARY OF TABLE A6 TO A13		V <sub>1</sub> T <sub>1</sub>					V <sub>6</sub> T <sub>1</sub>					V <sub>6</sub> T <sub>1</sub>		SUM Δ%	Tot. Investment
2	Constraint of Eq. (49): OTRR<0.2		\$4.84%					82.06%					43.84%		180.74%	for OTRR<0.2
3			\$130.31					\$195.01					\$104.18			\$429.5
4		V <sub>1</sub> T <sub>1</sub>	V <sub>1</sub> T <sub>3</sub>	V <sub>1</sub> T <sub>5</sub>	SUM CHECK		V <sub>6</sub> T <sub>1</sub>		V <sub>6</sub> T <sub>4</sub>	SUM CHECK		V <sub>6</sub> T <sub>1</sub>	V <sub>6</sub> T <sub>3</sub>	V <sub>6</sub> T <sub>5</sub>	SUM	SUM CHECK
5		89.90%	7.67%	2.43%	100.00%		93.95%		6.05%	100.00%		99.92%	0.04%	0.04%	100.00%	
6		\$117.16	\$9.99	\$3.16	\$130.31		\$183.20		\$11.81	\$195.01		\$104.10	\$0.039	\$0.039	\$104.18	\$429.5
7																
8	SUMMARY OF TABLE A6 TO A13		V <sub>6</sub> T <sub>1</sub>					V <sub>6</sub> T <sub>1</sub>		SUM Δ%	Tot. Investment					
9	Constraint of Eq. (50): OTRR<0.3		82.07%					43.59%			for OTRR<0.3					
10			\$215.18					\$114.31			\$329.5					
11		V <sub>6</sub> T <sub>1</sub>		V <sub>6</sub> T <sub>4</sub>	SUM CHECK		V <sub>6</sub> T <sub>1</sub>	V <sub>6</sub> T <sub>3</sub>	V <sub>6</sub> T <sub>5</sub>	SUM CHECK	SUM CHECK					
12		93.95%		6.05%	100.00%		99.92%	0.04%	0.04%	100.00%						
13		\$202.16		\$13.02	\$215.18		\$114.22	\$0.043	\$0.043	\$114.31		\$329.5				

## APPENDIX B: MICROSOFT EXCEL SOLVER COMPETITIVE-TO-SM (SECURITY METER) RISK MANAGEMENT SOLUTIONS IN TABLES B1 TO B4

In APPENDIX B, the authors display the alternative game-theoretic EXCEL (XL) Solver LP-feasible solutions for example 2 regarding the input Tables A4 and A5 in Appendix A. It is noteworthy to remark that the EXCEL (XL) Solver LP-feasible solutions, since there may be more than one LP-feasible solution [43–46] in the end, will generate identical outputs of  $OTRR \approx 0.2$  and  $OTRR \approx 0.3$ , which are those final constraints explored in APPENDIX A to mitigate to. As clarified in Table B1 to Table B4, the cumulative percentage change (Cum.  $\Delta\%$ ) in the optimal vector solution of  $LCM_{ij}$ ,  $i = 1, \dots, 10$ ;  $j = 1, 2$ , via EXCEL Solver using a game-theoretic algorithm via Eqs. (1)–(49) and (1)–(50) respectively exceed the cumulative percentage change (Cum.  $\% \Delta$ ) of the previous Appendix A’s Security Meter software algorithm.

XL's Cum.  $\Delta\%$   $\approx 203\%$  vs SM's Cum.  $\Delta\%$   $\approx 181\%$  for the

OTRR = 0.2 (i.e., OTNRR =  $1 - 0.2 = 0.8$ ) constraint of Eq. (49) is compared as favorable to SM. Also, XL's Cum.  $\% \Delta \approx 142\%$  vs SM's Cum.  $\Delta \approx 126\%$  for the OTRR = 0.3 (i.e., OTNRR =  $1 - 0.3 = 0.7$ ) constraint of Eq. (50) perform the same, as favorable to SM. This supports, as claimed in the Summary of Tables A4–A14, no other LP-feasible solution by any other commercial software can outsmart the SM algorithm in terms of minimizing the cumulative percent change, Cum.  $\Delta$ , of the LCM<sub>ij</sub> vector. LOSS  $\approx 0.029$  and LOSS  $\approx 0.056$  for OTRR = 0.2 and OTRR = 0.3, as minimized by the LP objective function, are hence generated in Tables B1 and B2. Regarding SM for Eq. (49), the article's takeaway is  $202.6\% - 180.7\% \approx 22\%$  less investment cost, and  $141.96\% - 125.7\% \approx 16\%$  less for Eq. (50) in example 2 of Tables A6–A7. For a simple demonstration, if the cost of investment is \$1m (million) per 1%, by the constraint of Eq. (49), ~\$22m is saved by the SM. Similarly, for the constraint of Eq. (50), ~\$16m is saved.

Table B1. XL solver LP-feasible solution for  $\text{LCM}_{ij}$  (B11:U11),  $\text{LOSS} \approx 0.029$ (V11),  $\text{TRR} \approx 0.629$ (V9),  $\text{OTRR} = 0.2$ (V14),  $\text{Cum. } \Delta\% \approx 203\%$ (V16)

The screenshot displays an Excel spreadsheet with a Solver Parameters dialog box overlaid. The spreadsheet contains numerical data across columns A-W and rows 1-26. The Solver Parameters dialog box shows the following settings:

- Solver Parameters:**
  - Set Objective:  $V_{11}$
  - To: ☐ Max ☒ Min ☐ Value Of: 0
  - By Changing Variable Cells:  $B_{11}:S_{11}$
  - Subject to the Constraints:
    - $B_{11} \geq S_{11}$
    - $B_{11} + S_{11} = 0$
    - $S_{11} = S_{16}$
    - $D_{11} = S_{16}$
    - $F_{11} = S_{16}$
    - $H_{11} = S_{16}$
    - $J_{11} = S_{16}$
    - $L_{11} = S_{16}$
    - $N_{11} = S_{16}$
    - $P_{11} = S_{16}$
    - $R_{11} = S_{16}$
    - $T_{11} = S_{16}$
    - $V_{11} = S_{16}$
    - $X_{11} = S_{16}$
  - ☒ Make Unconstrained Variables Non-Negative
- Select a Solving Method: GRG Nonlinear
- Solving Method: Select the LP Simplex engine for linear Solver Problems, and select the Evolutionary engine for Solver.



Table B2. EXCEL solver LP-feasible solution for LCMij (B11:U11), LOSS  $\approx 0.056$ (V11), TRR  $\approx 0.629$ (V9), OTRR = 0.3(V14), Cum.  $\Delta\% \approx 142\%$ (V16)

[illegible]

Table B3. XL solver risk management of merged threat data from Tables A5 and B1 for (49)  $\rightarrow$  OTRR (Optimized)  $\approx 0.2$ , Cum.  $\Delta\%$   $\approx 203\%$

Vulnerability	Threat	Vulnerability*Threat	LCM (XL)	Optimized Residual Risk	Difference from Original Input LCM Vector
0.218	0.8322	0.1814196	0.1576665	0.028603793	0.7604130
0.218	0.1678	0.0365804	0.25207774	0.009221105	
0.0453	0.7745	0.03508485	0.03116043	0.001093259	
0.0453	0.2255	0.01021515	0.89175789	0.009109441	
0.0071	0.9944	0.00706024	0.63192018	0.004461508	
0.0071	0.0056	0.00003976	0.26267045	1.04438E-05	
0.0125	0.3003	0.00375375	0.56656422	0.00212674	
0.0125	0.6997	0.00874625	0.07867729	0.000688131	
0.2902	0.9948	0.28869096	0.100597	0.029041531	0.7202250
0.2902	0.0052	0.00150904	0.97187958	0.001466605	
0.2219	0.957	0.2123583	0.1367573	0.029041548	0.3016759
0.2219	0.043	0.0095417	0.62435882	0.005957445	
0.0821	0.9971	0.08186191	0.3547626	0.029041544	0.2442836
0.0821	0.0029	0.00023809	0.27987393	6.66352E-05	
0.0183	0.7227	0.01322541	0.996674	0.013181422	
0.0183	0.2773	0.00507459	0.15582265	0.000790736	
0.0904	0.9228	0.08342112	0.30387833	0.025349871	
0.0904	0.0772	0.00697888	0.08006754	0.000558782	
0.0142	0.1268	0.00180056	0.77186225	0.001389784	
0.0142	0.8732	0.01239944	0.67437953	0.008361929	
SUMS→		1.00		OTRR ≈ 0.20	2.026(=202.6%) ≈ 203% = Cum. Δ %

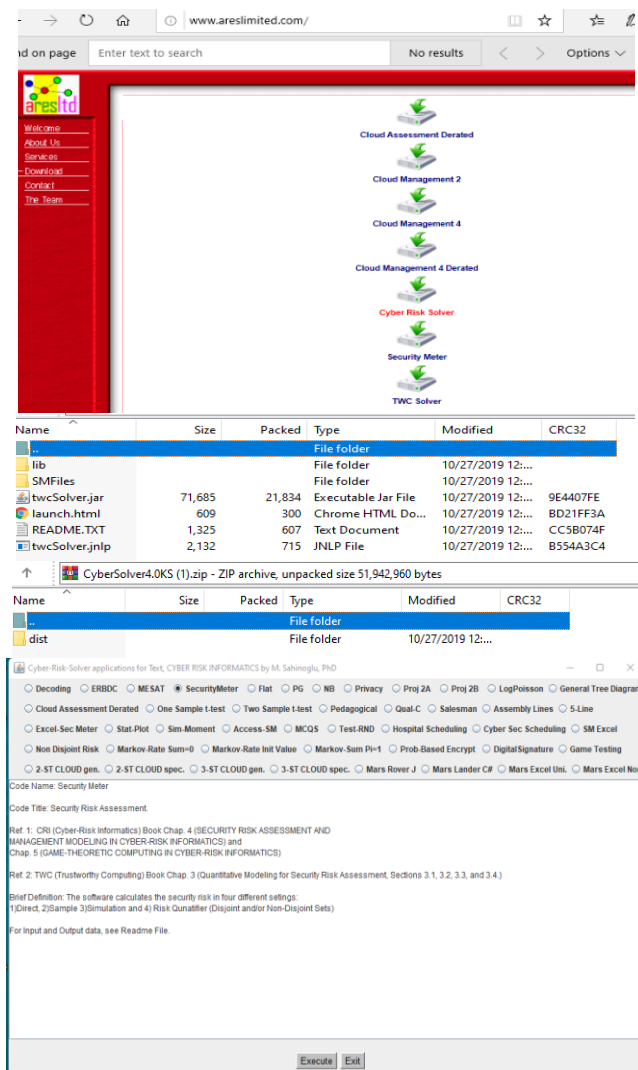
Table B4. XL solver risk management of merged threat data from Tables A5 and B2 for (50)  $\rightarrow$  OTRR (Optimized)  $\approx 0.3$ , Cum.  $\Delta\% \approx 142\%$

Vulnerability	Threat	Vulnerability*Threat	LCM (XL)	Optimized Residual Risk	Difference from Original Input LCM Vector
0.218	0.8322	0.1814196	0.3024443	0.054869324	0.6156352
0.218	0.1678	0.0365804	0.2520777	0.009221105	
0.0453	0.7745	0.03508485	0.0311604	0.001093259	
0.0453	0.2255	0.01021515	0.8917579	0.009109441	
0.0071	0.9944	0.00706024	0.6319202	0.004461508	0.6278511
0.0071	0.0056	0.00003976	0.2626705	1.04438E-05	
0.0125	0.3003	0.00375375	0.5665642	0.00212674	
0.0125	0.6997	0.00874625	0.0786773	0.000688131	
0.2902	0.9948	0.28869096	0.1929712	0.055709041	0.1760981
0.2902	0.0052	0.00150904	0.9718796	0.001466605	
0.2219	0.957	0.2123583	0.2623351	0.055709036	
0.2219	0.043	0.0095417	0.6243588	0.005957445	
0.0821	0.9971	0.08186191	0.5990462	0.049039067	0.000000000
0.0821	0.0029	0.00023809	0.2798739	6.66352E-05	
0.0183	0.7227	0.01322541	0.996674	0.013181422	
0.0183	0.2773	0.00507459	0.1558227	0.000790736	
0.0904	0.9228	0.08342112	0.3038783	0.025349871	0.000000000
0.0904	0.0772	0.00697888	0.0800675	0.000558782	
0.0142	0.1268	0.00180056	0.7718623	0.001389784	
0.0142	0.8732	0.01239944	0.6743795	0.008361929	
SUMS→		1.00		OTRR ≈ 0.30	1.4196(=141.96%) ≈ 142% = Cum. Δ %

These arguments were readily verified in Tables B1–B4 in Appendix B. The computations illustrate that the SM algorithm is monetarily the most cost-optimal to invest in due to the minimum competing cumulative change percentages, i.e. Cum.  $\Delta\%$ , to reach the OTRR = 0.2 and OTRR = 0.3, respectively, using the optimizing LP constraints from Eqs. (1)–(49) and (1)–(50) skipping (49). OTRR is defined as the Optimal Total Residual Risk. OTNRR is equal to the Optimal Total Non-Residual Risk as the complement of the OTRR.

#### APPENDIX C: HOW TO INSTALL CYBERRISKSOLVER APP TO RUN SM

See the detailed instructions after the following screen captures:



#### Instructions:

- 1) Type [www.areslimited.com](http://www.areslimited.com). Type in the user name: *mehmetstuna*, and password: *Mehpareanne*, and click OK.
- 2) Go to DOWNLOAD on [www.areslimited.com](http://www.areslimited.com) for the left-hand side menu's 4<sup>th</sup> from the top.
- 3) Click on the *Cyber Risk Solver* in red and download the application which is a ZIP file. Unzip or extract the downloaded application into the C:\myapp folder. See C:\myapp\dist. Open a Command Prompt and go to C:\myapp\dist>java SecurityB.jar folder and run the command: //For Cyber Risk Solver, java -jar twcSolver.jar.

Right-click on the twcSolver.jar to create a shortcut. Use the license code: **EFE28SEP1986** when it is requested.

- 4.) Click the Security Meter app (checked). Click Open. Use the license code: **HAKAN07MAR1995** when requested.

#### CONFLICT OF INTEREST

The author declares no conflict of interest in this work.

#### ACKNOWLEDGMENT

The author (MS) thanks various U.S. State Agencies for providing data support through personal communications. MS also appreciates Rasika K. Balasuriya, a PFW (Purdue Fort Wayne) Math Science instructor and a former M.S. graduate of the CSIS (Cyber Systems and Information Security) founded by the author at AUM (2009–) under the author's supervision (2015–2016), toward his selfless assistance while the author was working on the backbone of the EXCEL Solver calculations. This manuscript was considerably improved, thanks to the two anonymous reviewers' first-class constructive and reader-friendly suggestions, while they appreciated and helped clarify innovative research findings that go back ~20 years to 2005.

#### REFERENCES

- [1] M. Sahinoglu, *Trustworthy Computing: Analytical and Quantitative Engineering Evaluation*, J. Wiley & Sons Inc., Hoboken, NJ, 2007.
- [2] SANS Information Security Resources. [Online]. Available: <http://www.sans.org/information-security/>
- [3] M. McLean. (January 4, 2024). 2023 Must-Know Cyber Attack Statistics and Trends. Embroker\_Blog Business Advice & Research. [Online]. Available: <https://www.embroker.com/blog/cyber-attack-statistics/>
- [4] M. Sahinoglu, *Cyber-Risk Informatics: Engineering Evaluation with Data Science*, John Wiley and Sons, Hoboken, New Jersey, 2016.
- [5] M. Sahinoglu, "An input-output measurable design for the security meter model to quantify and manage software security risk," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 6, pp. 1251–1260, June 2008.
- [6] Federal Information Security Management Act (FISMA). Bing.com. [Online]. Available: [https://cn.bing.com/search?q=Federal+Information+Security+Management+Act+\(FISMA\).&cid=d743cc85d9e9450b88a534c6693bdcce&aqs=edge..69i57j0.3561j0j4&FORM=ANAB01&PC=HCTS](https://cn.bing.com/search?q=Federal+Information+Security+Management+Act+(FISMA).&cid=d743cc85d9e9450b88a534c6693bdcce&aqs=edge..69i57j0.3561j0j4&FORM=ANAB01&PC=HCTS)
- [7] Cybersecurity. NIST. [Online]. Available: <https://www.nist.gov/cybersecurity>
- [8] M. Sahinoglu, "Quantitative risk assessment of software security and privacy, and risk management with game theory," presented at CERIAS, the Purdue University Annual Symposium, February 2009.
- [9] M. Sahinoglu, "Quantitative risk assessment for dependent vulnerabilities," in *Proc. RAMS-06, The International Symposium on Product Quality and Reliability (52nd Year)*, New Port Beach, CA, 2006.
- [10] M. Sahinoglu, "Quantitative risk assessment for software maintenance with Bayesian principles," in *Proc. International Conference on Software Maintenance*, Budapest, Hungary, 2005, pp. 67–70.
- [11] M. Sahinoglu, "An empirical Bayesian stopping rule in testing and verification of behavioral models," *IEEE Trans. Instrumen. and Measurement*, vol. 52, no. 5, pp. 1428–1443, 2003.
- [12] M. Sahinoglu, D. Libby, and S. Das, "Measuring availability indices with small samples for component and network reliability using the Sahinoglu-Libby probability model," *IEEE Trans. Instrumen. and Measurement*, vol. 54, no. 3, pp. 1283–1295, June 2005.
- [13] M. Sahinoglu, "Security meter: A probabilistic framework to quantify security risk," Certificate of Registration, US Copyright Office, TXu 1-134-116, 5 Dec. 2003.
- [14] M. Sahinoglu, "Security meter: A practical decision tree model to quantify risk," *IEEE Security and Privacy*, vol. 3, issue 3, pp. 18–24, 2005.
- [15] E. Forni. (2002). Certification and accreditation. AUM Lecture Notes, Data Systems Design Laboratories. [Online].



- [16] Definition of Information Risk. IT Auditing. [Online]. Available: <http://it-auditing.wikidot.com/information-risks>.
- [17] Information Systems Audit and Control Association (ISACA). [Online]. Available: [https://www.bing.com/search?q=Information+System+Audit+and+Control+Association+\(ISACA\)&cv=480875a832274a0aa9a409f9d045bd2d&aqs=edge..69i57.12930j0j1&pglt=41&FORM=ANN TA1&PC=HCTS](https://www.bing.com/search?q=Information+System+Audit+and+Control+Association+(ISACA)&cv=480875a832274a0aa9a409f9d045bd2d&aqs=edge..69i57.12930j0j1&pglt=41&FORM=ANN TA1&PC=HCTS)
- [18] M. Sahinoglu, Y.-L. Yuan, and D. Banks, "Validation of a security and privacy risk metric using the triple uniform product rule," *International Journal of Computers, Information Technology and Engineering*, vol. 4, no. 2, pp. 125–135, December 2010.
- [19] R. Bojanc and B. Jerman-Blazic, "Quantitative model for information security risk management," *Engineering Management Journal*, vol. 25, no. 2, pp. 25–37, 2015.
- [20] S. Ashokan, "Quantitative risk assessment of vulnerabilities and threats," M.S. thesis, Auburn University Montgomery, 2015.
- [21] V. Evrin, *Risk Assessment and Analysis Methods: Qualitative and Quantitative*, CISA, CRISC, COBIT 2019 Foundation, 2021.
- [22] W. G. Meyer, "Quantifying risk: Measuring the invisible," presented at PMI@ Global. Congress—EMEA, London, England, 10 October 2015.
- [23] D. Tan. (2002). Quantitative risk analysis is step-by-step. SANS White Papers. [Online]. Available: <https://www.sans.org/white-papers/849/>
- [24] R. Schmittling and A. Munns. (2010). Performing a security risk assessment. [Online]. Available: <https://www.isaca.org/resources/isaca-journal/past-issues/2010/performing-a-security-risk-assessment>
- [25] M. Sahinoglu, "Method for cyber-security risk assessment and cost-efficient management in wireless sensor networks," in *Proc. the 3rd Cyberspace Research Workshop, CRW'10*, Shreveport, LA., 2010, pp. 8–13. <http://csc.test.latech.edu/crw10/proceedings.pdf>
- [26] S. Bansal. (2019). Quantitative vs qualitative risk analysis by iZenBridge. [Online]. Available: <https://www.izenbridge.com/blog/differentiating-quantitative-risk-analysis-and-qualitative-risk-analysis/>
- [27] National Vulnerability Database. [Online]. Available: <https://nvd.nist.gov/>
- [28] CVE Terminology. [Online]. Available: <https://cve.mitre.org/about/terminology.html>
- [29] Common Vulnerabilities and Exposures. [Online]. Available: [http://en.wikipedia.org/wiki/Common\\_Vulnerabilities\\_and\\_Exposures](http://en.wikipedia.org/wiki/Common_Vulnerabilities_and_Exposures)
- [30] Common Vulnerabilities and Exposures. CVE Detail [Online]. Available: <http://www.cvedetails.com/>
- [31] What is a Buffer Overflow? Learn About Buffer Overrun Vulnerabilities, Exploits & Attacks. [Online]. Available: <https://www.veracode.com/blog/2012/04/what-is-a-buffer-overflow-learn-about-buffer-overrun-vulnerabilities-exploits-attacks>
- [32] Java Script Security. [Online]. Available: <http://www.veracode.com/security/javascript-security>
- [33] Race Condition. [Online]. Available: [http://en.wikipedia.org/wiki/Race\\_condition](http://en.wikipedia.org/wiki/Race_condition)
- [34] Cross-site scripting. [Online]. Available: [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting)
- [35] SQL Injection. [Online]. Available: [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)
- [36] File Inclusion. [Online]. Available: [http://en.wikipedia.org/wiki/File\\_inclusion\\_vulnerability](http://en.wikipedia.org/wiki/File_inclusion_vulnerability)
- [37] Denial of Service Attack. [Online]. Available: [http://en.wikipedia.org/wiki/Denial-of-service\\_attack](http://en.wikipedia.org/wiki/Denial-of-service_attack)
- [38] Format String Attack. [Online]. Available: [https://www.owasp.org/index.php/Format\\_string\\_attack](https://www.owasp.org/index.php/Format_string_attack)
- [39] Directory Traversal. [Online]. Available: [http://en.wikipedia.org/wiki/Directory\\_traversal\\_attack](http://en.wikipedia.org/wiki/Directory_traversal_attack)
- [40] Untrusted Search Paths. [Online]. Available: <https://cwe.mitre.org/data/definitions/426.html>
- [41] Risk Optimization. [Online]. Available: <http://www.iadclexicon.org/risk-optimization/>
- [42] D. Anderson, D. Sweeney, T. Williams, and K. Martin, *An Introduction to Management Science. Quantitative Approaches to Decision Making*, 13th ed. South-Western Cengage Learning, 2011.
- [43] G. B. Dantzig, *Linear Programming and Extensions*, Princeton, NJ, Princeton University Press, 1963.
- [44] M. Sahinoglu, "Market-centric quantifiable risk management of cloud computing by operational planning," AUM Research Paper, Invited Seminar, Computer Science, Colorado State U, Ft. Collins, 2017.
- [45] M. Sahinoglu, "Cloud computing risk assessment and management in modern power systems—Application to Microgrids," Invitational Seminar at Department of Electrical Engineering and Computer Science and CASE, Syracuse University, 2017.
- [46] M. Sahinoglu, "Alternative game-theoretical solutions to the quantitative risk assessment algorithm," in *Proc. INTERFACE'11*, Cary, NC, June 2011.
- [47] M. Sahinoglu, "Cloud computing risk assessment and management—An application to cyber- and power systems," Invited Seminar at Lawrence Livermore National Lab, February 5, 2018.
- [48] M. Sahinoglu, "Generalized game theory applications to computer security risk," in *Proc. the IEEE Symposium on Security and Privacy*, Oakland, CA, May, 2008, pp. 18–21.
- [49] M. Sahinoglu, L. Cueva-Parra, and D. Ang, "Game-theoretic computing in risk analysis," *WIREs Comput. Stat.*, vol. 4, pp. 227–248, 2012.
- [50] M. Sahinoglu, "A universal quantitative risk assessment design to manage and mitigate," in *Proc. International Conference on the Digital Information Industry*, Seoul, Korea, pp. 333–340, November 2006.
- [51] M. Benini and S. Sicari, "Risk assessment in practice: A real case study," *Computer Communications*, vol. 31, no. 15, pp. 3691–3699, 2008.
- [52] S. Fenz and A. Ekelhart, "Verification, validation and evaluation in information security risk management," *IEEE Security and Privacy*, pp. 58–65, 2011.
- [53] About ENISA—The European Union Agency for Cybersecurity. Towards a Trusted and Cyber Secure Europe. [Online]. Available: <https://www.enisa.europa.eu/about-enisa>
- [54] M. Sahinoglu, "Security-meter model—a simple probabilistic model to quantify risk," in *Proc. 55th Session of the International Statistical Institute*, Sydney, Australia, Conference Abstract Book, 2005, p. 163.
- [55] M. Sahinoglu, Statistical inference to quantify and manage the risk of privacy, in *Proc. the 56th Session of the International Statistical Institute (ISI), Session 22 (S80:Risk)*, Lisbon, Portugal; ISI Book of Abstracts, August 2007, p. 506.
- [56] M. Sahinoglu, "Optimal risk management of electric power systems with cloud simulation and security meter algorithms," in *Handbook of Smart Energy Systems*, M. Fathi, E. Zio, P. M. Pardalos, Eds. Springer, Cham, 2021. [https://doi.org/10.1007/978-3-030-72322-4\\_85-1](https://doi.org/10.1007/978-3-030-72322-4_85-1)
- [57] M. Sahinoglu, "Risk assessment of electric power generation systems by stochastic simulation using cloud computing," in *Handbook of Smart Energy Systems*, M. Fathi, E. Zio, P. M. Pardalos, Eds. Springer, Cham, 2022. [https://doi.org/10.1007/978-3-030-72322-4\\_118-1](https://doi.org/10.1007/978-3-030-72322-4_118-1)
- [58] M. Sahinoglu and L. Cueva-Parra, "Cloud computing," *Wiley Interdisciplinary Review Series Computational Statistics*, vol. 3, no. 1, pp. 47–68, 2011.
- [59] M. Sahinoglu, "Security risk-meter: Software for quantitative risk & cost optimal management in national cyber-security," presented at Invitational Seminar of Cybersecurity Research and Education Institute, Eric Jonsson School of Engineering & Computer Science, University of Texas at Dallas, 2018.
- [60] M. Sahinoglu and K. Wool, "Risk assessment and management to estimate and improve the hospital credibility score of a patient's healthcare quality," *Book Chapter (Society of Design and Process Science) Cyber-Physical Systems of the Future: Transdisciplinary Convergence in the 21st Century*, S. Suh et al. (eds.), Springer Publishing New York, 2013.
- [61] M. Sahinoglu, S. Morton, D. Ang, P. Vasudev, and W. Kramer, "Quantitative metrics to assess and manage business contracting risk using Risk-O-Meter software," *IJBI (International Journal of Business and Information)*, 2016.
- [62] M. Sahinoglu, S. Simmons, L. Cahoon, "Ecological Risk-O-Meter: A risk assessor and manager software for decision-making in ecosystems," *Environmetrics*, vol. 23, pp. 729–737, 2012.
- [63] M. Sahinoglu, S. Stockton, R. Barclay, and S. Morton, "Metrics-based risk assessment and management of digital forensics," *Defense A.R.J.*, vol. 23, no. 2, pp. 152–177, April 2016.
- [64] M. Sahinoglu, D. Marghitu, and V. Phoha, *Risk Studies of Operational Variations for Onshore & Offshore Oil-Rigs*, Lambert Academic Publishing, ISBN: 978-3-659-97152, 2017.
- [65] M. Sahinoglu, S. Morton, and N. Gokul, *How to Quantify Metrics to Assess and Manage Wireless Security Risk*, Lambert Academic Publishing, ISBN: 978-3-330-33712, 2017.
- [66] M. Sahinoglu and J. C. Petty, "Quantitative risk assessment and management of national defense acquisition with a game-theoretic security risk meter tool," *International Journal of Computer Theory and Engineering*, vol. 15, no. 4, pp. 152–177, 2023. doi: 10.763/IJCTE.V15.1344
- [67] M. Sahinoglu, L. Cueva-Parra, and D. Ang, "The modeling and simulation in engineering," *WIREs Computational Statistics*, vol. 5, pp. 239–266, 2013. doi: 10.1002/wics.1254

Copyright © 2024 by the authors. This is an open-access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).