RESEARCH ARTICLE OPEN ACCESS

FIRE: Fuzzy-logic based Theoretic Intrusion Response and Recovery Engine

Deepika Irugu [1], P. Balaji [2], P. Nirupama [3]
M.Tech Student [1], Associate Professor [2], Head of the Department [3]
Department of Computer Science and Engineering
Siddharth Institute of Engineering and Technology
Siddharth Nagar, Puttur, Chittoor
Andhra Pradesh - India

ABSRACT

Preserving the supply and integrity of networked computing systems within the face of fast-spreading intrusions needs advances not solely in detection algorithms, however additionally in automatic response techniques. During this paper, we tend to propose a brand new approach to automatic response referred to as the response and recovery engine (RRE). Our engine employs a game-theoretic response strategy against adversaries sculptural as opponents in a very two-player Stackelberg random game. The RRE applies attack-response trees (ART) to research unwanted system-level security events among host computers and their countermeasures victimization symbolic logic to combine lower level attack consequences. Additionally, the RRE accounts for uncertainties in intrusion detection alert notifications. The RRE then chooses optimum response actions by determination a partly evident competitive mathematician call method that's automatically derived from attack-response trees. To support network-level multiobjective response choice and contemplate presumably conflicting network security properties, we tend to use symbolic logic theory to calculate the network-level security metric values, i.e., security levels of the system's current and doubtless future states in every stage of the sport. Especially, inputs to the network level game-theoretic response choice engine, square measure initial fed into the fuzzy system that's accountable of a nonlinear illation and quantitative ranking of the potential actions victimization it's antecedently outlined fuzzy rule set. Consequently, the optimum network-level response actions square measure chosen through a game-theoretic improvement method. Experimental results show that the RRE, victimization Snort's alerts, will defend massive networks that attack-response trees have quite five hundred nodes.

Keywords:- Intrusion response systems, network state estimation, Andre Mark off call processes, random games, and symbolic logic and management.

I. INTRODUCTION

The severity and variety of intrusions on pc networks square measure quickly increasing. Generally, incident handling techniques square measure classified into 3 broad classes. First, there square measure intrusion bar ways that take actions to forestall incidence of attacks, for instance, network flow secret writing to forestall man-in-the-middle attacks. Second, there square measure intrusion detection systems (IDSes), that attempt to observe inappropriate, incorrect, or abnormal network activities, for instance, perceiving CrashIIS attacks by police work deformed packet payloads. Finally, there square measure intrusion response techniques that take responsive actions supported received IDS alerts to stop attacks before they will cause vital injury and to ensure safety of the computing atmosphere. So far, most research has centered on rising techniques for intrusion prevention and detection, whereas intrusion response typically remains a manual method performed by network directors who square measure notified by IDS alerts and reply to the intrusions. This manual response method inevitably introduces some delay between notification and response, which could be simply exploited by the offender to attain his or her goal and considerably increase the injury. Therefore, to cut back the severity of attack injury ensuing from delayed response, an automatic intrusion response is required that gives instant response to intrusion. In this paper, we have a tendency to gift an automatic

cost-sensitive intrusion response system referred to as the response and recovery engine (RRE) that models the protection battle between itself and also the attacker as a multi-step, sequential, ranked, nonzero sum, two-player random game. In every step of the sport, RRE leverages a replacement extended attack tree structure, called the attack-response tree (ART), and received IDS alerts to evaluate varied security properties of the individual host systems among the network. ARTs give a proper thanks to describe host system security supported doable intrusion and response situations for the offender and response engine, severally. a lot of significantly, ART's modify RRE to consider inherent uncertainties in alerts received from IDS'es (i.e., false positive and false negative rates), when estimating the system's security and preferring response actions. Then, the RRE mechanically converts the attack response trees into part noticeable competitive Markov decision processes that are solved to search out the optimum response action against the aggressor, within the sense that the maximum discounted accumulative injury that the attacker will cause later within the game is reduced. It is noteworthy that despite the mathematical price minimisation in RRE that itself needs a while to finish in follow, RRE's final objective is to save/reduce intrusion response costs and also the system damages because of attacks compared to existing intrusion response solutions. mistreatment this gametheoretic approach, RRE adaptively adjusts its behavior according to the attacker's potential future reactions, thus preventing the wrongdoer from inflicting important injury to the system by taking Associate in Nursing showing intelligence chosen sequence of actions. To wear down security problems with totally different granularities, RRE's two-layer design consists of native engines that reside in individual host computers and the global engine that resides within the response and recovery server and decides on international response actions once the system isn't recoverable by the native engines. what is more, the hierarchal design improves measurability, ease of design, and performance of RRE, in order that it will defend computing assets against attackers in large-scale laptop networks. To support network-level intrusion response where the world security level is usually a perform of different specific properties and business objectives, RRE employs a fuzzy control-based technique that may take into account many objective functions at the same time. In particular, reports from native engines area unit fed into the world response engine's fuzzy system as inputs. Then, the RRE calculates quantitative various the potential network-level response actions mistreatment its antecedently outlined fuzzy rule set. The fuzzy rule set is outlined victimisation fuzzy numbers, and hence, varied input parameters will defy qualitative values like high or low; so, the real-world challenge that correct crisp values of the concerned parameters aren't invariably notable is self-addressed fully.

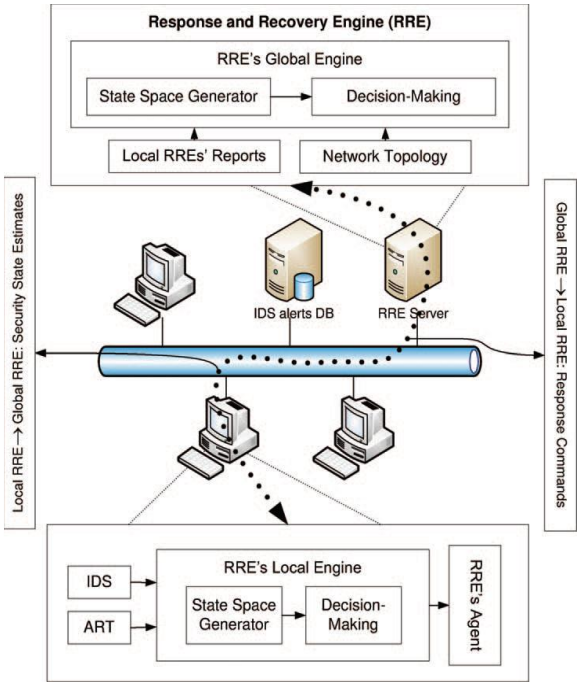

Fig. 1. High-level architecture of the RRE

RRE extends the state of the art in intrusion response in three elementary ways in which. First, RRE accounts for planned adversarial behavior within which attacks occur little by little in which adversaries execute well-planned ways and address defense measures taken by system directors along the method. It will therefore by applying theory of games and seeking responses that optimize on semi-permanent gains. Second, RRE at the same time accounts for inherent uncertainties in IDS alert notifications with attack-response trees converted to a partly evident Andrei Markov call method that computes optimum responses despite these

uncertainties. This is necessary as a result of IDS'es these days and within the close to future are going to be unable to get alerts that match utterly to triple-crown intrusions, and response techniques should, therefore, give this imperfectness to be sensible. Third, for easy style functions, RRE permits network security administrators to outline high-level network security properties through easy-to-understand linguistic terms for the particular target network. This is a vital facility that RRE provides, as a result of not like system-level security properties, for example, the online server convenience, which may be reused across networks, the network-level security properties usually ought to be outlined specifically for every network by the safety directors manually. RRE achieves the above 3 goals with a unified modeling approach in which theory of games and Andre Markoff call processes are combined. We have a tendency to demonstrate that RRE is computationally efficient for comparatively massive networks via prototyping and experimentation, demonstrate that it's sensible by studying usually found grid important infrastructure networks. However, we tend to believe that RRE has wide applicability to any or all sorts of networks.

II. RELATED WORK

EMERALD a dynamic cooperative response system, introduces a stratified approach to deploy monitors through different abstract layers of the network. Analyzing IDS alerts and coordinative response efforts, the response components also are able to communicate with their peers at different network layers. AAIRS provides adaptation through a confidence metric related to IDS alerts and through successful metric appreciate response actions. Although EMERALD and AAIRS provide nice infrastructure for automatic federal agency, they are doing not arrange to balance intrusion damage and recovery value.

III. PROBLEM FORMULATION

We formulate the best response choice as a decision making problem within which the goal is to decide on the cost optimal response action at whenever instant. The best action m is picked out of the set of all attainable response actions m2M, as well as the No-OPeration (NOP) action. For example, associate degree intrusion response system will reply to SQL's buffer overflow exploitation by closing its communications protocols connection. The improvement downside is solved within the response system, given the subsequent inputs:
W: a collection of the computing assets w 2W, as an example, an SQL server, that square measure to be protected by the response engine.
O: a collection of IDS alerts o a pair of O that specifically indicate associate degree adversarial plan to exploit the prevailing specific vulnerabilities of the assets, alerts from Snort warning a couple of packet transferring the Slammer worm that exploits a buffer overflow vulnerability in associate degree SQL server.
G: a collection of ART graphs g a pair of G that consistently outline how intrusive (responsive) eventualities regarding the aggressor (response engine) have an effect on system security. The following sections square measure dedicated to an answer to the response choice problem; in different words, we'll focus on however the RRE finds the best response action supported given input arguments.

IV. RRE HIGH-LEVEL ARCHITECTURE

Before giving theoretical style and implementation details, we provide a high-level design of RRE, as illustrated in Fig. 1. it's 2 kinds of decision-making engines at 2 totally different layers, i.e., native and international. This hierarchical structure of RRE's design, as mentioned later, makes it capable of handling terribly frequent IDS alerts, and choosing optimum response actions. Moreover, the two-layer architecture improves its quantifiability for large-scale computer networks, within which RRE is meant to safeguard a large number of host computers against malicious attackers.

Finally, separation of high- and low-level security issues considerably simplifies the correct style of response engines. At the first layer, RRE's local engines are distributed in host computers. Their main inputs consist of IDS alerts and attack-response trees. All IDS alerts are sent to and stored in the alert database (see Fig. 1) to which each local engine

subscribes to be notified when any of the alerts related to its host computer is received. It is noteworthy that the current RRE design assumes that the triggered alerts are trusted. Using the mentioned local information, local engines compute local response actions and send them to RRE agents that are in charge of enforcing received commands and reporting back the accomplishment status, i.e., whether the command was successfully carried out. The internal architecture of engines includes two major components: the state space generator, and the decision engine. Once inputs have been received, all possible cyber security states, which the host computer could be in, are generated. The state space might be intractably large; therefore, RRE partially generates the state space so that the decision-making unit can quickly decide on the optimal response action. The decision-making unit employs a game-theoretic algorithm that models attacker-RRE interaction as a two-player game in which each player tries to maximize his or her overall benefit. This implies that, once a system is under attack, immediate greedy response decisions are not necessarily the best choices, since they may not guarantee the minimum total accumulative cost involved in complete recovery from the attack.

Although individual native engines arrange to defend their corresponding host computers, they will become malicious themselves if they get compromised. moreover, it could become terribly difficult, even not possible, for native engines to choose and take a world network-level response action, due to their restricted native information. To handle these problems, RRE's world engine, as its second layer, obtains high-level info from all host computers within the network, decides on best world response actions to require, and coordinates RRE agents to accomplish the actions by sending them relevant response commands. in addition, if a local engine is detected to be compromised or doesn't respond, the RRE's world engine takes network level actions to prevent any injury, for instance, to quarantine the compromised node, and/or probably live through the attack, for instance, to modify to the secondary repilca of the compromised node within the network. additionally to native security estimates from host computers, topology as well because the world network access management policies are fed into the world engine. RRE converts the network topology and access management policies into the competitive Markov call method (CMDP) model mechanically. Moreover, security directors outline the network security properties as a perform of the safety of the network's essential assets victimisation easy-to-understand linguistic terms.

RRE employs the defined network-level security properties as security metrics to select the optimal network level response action by solving the generated network CMDP model. The ART model in the global server within RRE formulates the high-level organizational objectives that are subjective and require human involvement by the security administrators to capture the attack consequences that affect those objectives. For instance, confidentiality of a logging server in a financial institute may be considered as a critical security property while it could be ignored in a process control network. Consequently, the single global ART model in RRE's global server needs to be designed manually; however, the local ART models within individual hosts, such as the Apache web server, capture the system level consequences, for example, the web server availability. Hence, the local ART models can be reused across systems in different networks as they are not dependent on the high-level objectives. The reusability of the ART models reduces the manual endeavor requirement for the overall system deployment.

V. LOCAL RESPONSE AND RECOVERY

Starting with very cheap level modules in RRE, we explain how native engines, residing in host computers, defend native computing assets exploitation security-related data, i.e., IDS alerts, about them. Attack-response tree to shield a neighborhood computing plus, its corresponding native engine initial tries to work out what security properties of the plus are profaned as results of an attack, given a received set of alerts. Attack trees provide a convenient thanks to consistently reason the various ways within which AN plus is attacked, native engines build use of a brand new extended attack tree structure, known as AN attack response tree (ART), that creates it doable 1) to include possible measure (response) actions against attacks, and 2) to contemplate intrusion detection uncertainties because of false positives and negatives

in sleuthing in intrusions, while estimating the present security state of the system.

The attack-response trees are designed offline by experts for each computing asset, for example, an SQL server, residing in a host computer. It is important to note that, unlike the attack tree that is designed according to all possible attack scenarios, the ART model is built based on the attack consequences, for example, an SQL crash; thus, the designer does not have to consider all possible attack scenarios that might cause those consequences.
A node decomposition scheme could be based on either 1) an AND gate, where all of the subconsequences must happen for the abstract consequence to take place, or 2) an OR gate, where occurrence of any one of the subconsequences will result in the abstract consequence. For a gate, the underlying subconsequence(s) and the resulting abstract consequence are called input(s) and output, respectively. Being at the lowest level of abstraction in the attack-response tree structure, every leaf node consequence is mapped to its related subset of IDS alerts, each of which represents a specific vulnerability exploitation attempt by the attacker.

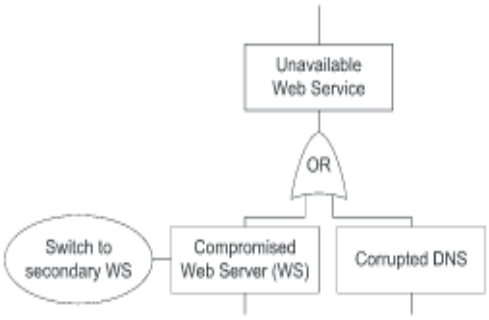

Fig. 2. Node decomposition in ART.

Some of the consequence nodes in associate degree ART graph tagged by response boxes that represent measure (response) actions m2M against the implications to which they're connected. Fig. two illustrates however a sample abstract consequence node (output), i.e., associate degree unprocurable net service, is rotten into 2 sub-consequences (inputs) using associate degree OR gate; this implies that the online service becomes unavailable if either the online server is compromised or the domain name server is corrupted. moreover, if a web service is unprocurable because of the compromised net server, the response engine will switch to the secondary net server. Fig. three shows however a typical ART would finally look. For every ART graph, a significant goal is to probabilistically verify whether or not the protection property specific by ART's root node has been profaned, given the sequence of 1) the received alerts, and 2) the success taken response actions. Boolean values are allotted to all or any nodes within the attack-response tree every leaf node consequence is initially zero, and is about to one once any alert from its corresponding alert set (defined earlier) is received from the IDS. These values for alternative consequence nodes, including the basis node, are merely determined bottom-up according to leaf nodes' values within the subtree whose root is the consequence node into consideration. Response boxes are triggered once with success taken by the response engine; as a result, all nodes within their subtree reset to zero, and also the corresponding received alerts cleared. As a case in purpose, if the response box, that is connected to ART's root node is triggered, all nodes in the ART graph reset to zero.

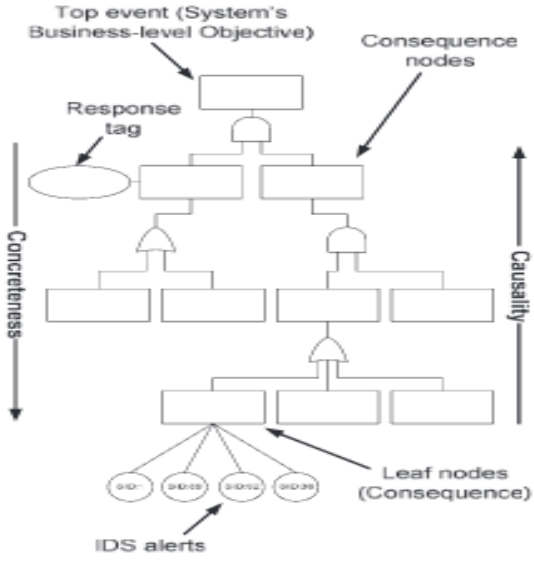

Fig. 3. Attack response tree.

Dealing with uncertainties: In reality, determining Boolean values of the leaf node consequences in ART is more complicated, due to the uncertainty about whether 1) the received alerts actually represent some consequence occurrence, and 2) no consequence has happened if no alert has been received. Taking such uncertainties into account,

RRE makes use of a naive Bayes binary classifier, that uses Bernouli variables, i.e., alerts, to determine the value of each leaf consequence node l, given the set of its related received alerts.

Stackelberg game: RRE versus aggressor. Reciprocal interaction between the soul and response engine in a very computer system could be a game within which every player tries to maximize his or her own profit. The response choice process in RRE is modeled as a ordered Stackelberg stochastic game within which RRE acts because the leader whereas the attacker is that the follower; but, in our infinite-horizon game model, their roles could modification while not touching the final solution to the matter.

Specifically, the sport could be a finite set of security states S that cover all potential security conditions that the system might be in. The system is in one among the safety states s at every time instant. RRE, the leader, chooses and takes a response action admittable in s, that ends up in a probabilistic security state transition to s0. The aggressor, that is that the follower, observes the action designated by the leader, and then chooses Associate in Nursingd takes an soul action s0 two O admissible in s0, leading to a probabilistic state transition to s00. At every transition stage, players could receive some reward per a present operate for every player. The reward function for an attacker is usually not known to RRE, because an attacker's reward depends on his final malicious goal, which is also not known; therefore, assuming that the attacker takes the worst possible adversary action, RRE chooses its response actions based on the security strategy, i.e., maximin, as discussed later. It is also important to note here that although S is a finite set, it is possible for the game to revert back to some previous state; therefore, the RRE-adversary game can theoretically continue forever. This stochastic game is essentially an antagonistic multi-controller Markov decision process, called a competitive Markov decision process (CMDP).

Automatic conversion: ART-to-MDP. Using the ART graphs, RRE's local engines automatically construct response Markov decision process (MDP) models, where security states are defined as a binary vector whose variables are actually the set of satisfied/unsatisfied (1/0) leaf consequence nodes in the ART graph under consideration. In other words, as a binary string, each MDP security state vector represents the ART leaf node consequences that have already been set to 1 according to the received alerts from IDS systems. For instance, an ART graph with n leaf nodes results in a generated MDP model with 2n security states, i.e., n-bit vectors. For ART graphs with a large number of leaf nodes, this exponential growth of the security state space usually results in the state space explosion problem, which RRE deals with by making use of approximation techniques.

Uncertainty in updating inputs, i.e., IDS alerts, converts our Markovian decision process into a higher level model, called a partially observable competitive Markov decision process (POCMDP). Indeed, states b 2 B, in this higher level model, are probability distributions over a set of states S in the underlying Markovian decision process model. It is noteworthy that the rationale behind having the ART models within RRE rather than having the security administrators to design the state-based Markov decision processes manually is that the ART trees are easier to understand and hence to design manually partially because of their tree structure and recursive design process for individual subtrees. As discussed earlier, RRE uses the manually designed ART model to construct the CMDP state space automatically. In addition to ARTs' easier understandability, to design the Markov decision process requires more effort than designing its corresponding ART tree because the number of security states in a Markov decision process is exponentially more compared to the number of nodes (more accurately number of leaf nodes) in its corresponding ART tree.

Optimal response strategy: As the last step in the decision making process in local engines, RRE solves the POCMDP to find an optimal response action from its action space, and sends an action command to its agents that are in charge of enforcing received commands. Action optimization in RRE is accomplished by trying to maximize the accumulative long-run reward measure received while taking sequential response actions. To accumulate sequential achieved rewards, here, we use the infinite-horizon discounted cost technique, which gives more weight to nearer future rewards. In other words, in each step, the game value is computed by recursively adding up the immediate reward after both players take their next actions and the discounted expected game value from then on.

Agents: In above-named security battle between RRE and also the somebody, agents play a key role in accomplishing every step of the sport. they're to

blame of taking response actions selected by RRE engines. Actually, having received commands from engines, agents try to carry them out with success and report the result, whether they were winning or not, back to the commander, i.e., the engine. If the agent's report indicates that some response action has been taken with success, the engines update their ART trees' corresponding variables, which are leaf node values within the subtree for the with success taken response action node. Consequently, as explained on top of leaf node variables in ART trees area unit updated by 2 sorts of messages: IDS alerts and agents' reports.

VI. GLOBAL RESPONSE AND RECOVERY

Although host-based intrusion response is taken into account by RRE's local engines using local ART graphs and the IDS rule-set for computing assets, for example, the SQL server, maintenance of global network-level security requires information about underlying network topology and profound understanding about what different combinations of secure assets are necessary to guarantee network security maintenance. As discussed, in the distributed local response engines, most of the security properties (ARTs' root nodes) are (objective) system-level concepts, for example, Is the apache process available?, and can be measured simply using the Boolean logic expressions (ART trees) and the triggered IDS alerts. In RRE, global network intrusion response is resolved in the central server. Unlike in local engines, in the global intrusion response engine, global network-level (possibly subjective) security properties, for example, Is the network currently secure?, are to be determined. Such global security properties do not always take on only binary values. As a case in point, in a large scale enterprise network, a web server compromise affects the network's current security level, but it does not mean that the network is completely insecure. Additionally, various network assets often have different levels of criticality and impact on accomplishment of the enterprise's overall business objective, and hence, affect the global security level differently.

The global engine's fuzzy controller is composed of the following four elements:
1. A rule-base (a set of If-Then rules), which contains a fuzzy logic quantification of the experts linguistic description of how to *achieve* accurate global network-level security measure estimates.
2. An inference module, which emulates the experts' decision-making in interpreting and applying knowledge about how best to estimate the global network-level security measure values.
3. A fuzzification interface, which converts the controller inputs from local response engines into information that the inference mechanism can easily use to activate and apply rules.
4. A defuzzification interface, which converts the conclusions of the inference mechanism into real number values as inputs to the game-theoretic intrusion response system to pick the cost-optimal response action.

VII. CONCLUSIONS

A game-theoretic intrusion response engine, called the response and recovery engine, was conferred. We modeled the security maintenance of pc networks as a Stackelberg random two-player game during which the attacker and response engine attempt to maximize their own benefits by taking best soul and response actions, respectively. Experiments show that RRE expeditiously takes appropriate step actions against in progress attacks that save system injury and intrusion response value compared to existing static and dynamic government agency solutions.

REFERENCES

[1] R. Rehman, Intrusion Detection Systems with Snort. Prentice-Hall, 2003.

[2] J. Filar and K. Vrieze, Competitive Markov Decision Processes. Springer-Verlag, 1997.

[3] B. Foo, M. Glause, G. Howard, Y. Wu, S. Bagchi, and E. Spafford, Information Assurance: Dependability and Security in Networked Systems. Morgan Kaufmann, 2007.

[4] B. Schneier, Secrets and Lies: Digital Security in a Networked World. John Wiley & Sons, 2000.

[5] E. Sondik, "The Optimal Control of Partially Observable Markov Processes," PhD thesis: Standford Univ., 1971.

[6] M. Bloem, T. Alpcan, and T. Basar, "Intrusion Response as a Resource Allocation Problem," Proc. Conf. Decision and Control, pp. 6283-6288, 2006.

[7] D. Ragsdale, C. Carver, J. Humphries, and U. Pooch, "Adaptation Techniques for Intrusion Detection and Intrusion Response System," Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics, pp. 2344-2349, 2000.

[8] O.P. Kreidl and T.M. Frazier, "Feedback Control Applied to Survivability: A Host-Based Autonomic Defense System," IEEE Trans. Reliability, vol. 53, no. 1, pp. 148-166, Mar. 2004.

[9] I. Balepin, S. Maltsev, J. Rowe, and K. Levitt, "Using Specification-Based Intrusion Detection for Automated Response," Proc. Int'l Symp. Recent Advances in Intrusion Detection, pp. 136-154, 2003.

[10] K. Lye and J. Wing, "Game Strategies in Network Security," Int'l J.Information Security, vol. 4, pp. 71-86, 2005.

[11] R.C. Berkan and S. Trubatch, Fuzzy System Design Principles, first ed. Wiley-IEEE Press, 1997.

[12] S.-J. Chen and S.-M. Chen, "Fuzzy Risk Analysis Based on Similarity Measures of Generalized Fuzzy Numbers," IEEE Trans. Fuzzy Systems, vol. 11, no. 1, pp. 45-56, Feb. 2003.

[13] P.Porras and P. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," Proc. Information Systems Security Conf., pp. 353-65,1997.