© 2016 IJSRSET | Volume 2 | Issue 1 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

A Review - Anomaly Based Network Security Using Response Recovery Engine

Ganesh Ghodke, Vaibhav Sarode, Sagar Valmiki, Prof. Patil S. S., Prof. Kothawale G. S.

Al-Ameen College of Engineering, Koregaon Bhima, Savitribai Phule Pune University, Pune, India

ABSTRACT

The security of the network reduces due to increase in the size of the network, there are many intrusion detection and intrusion response strategies which are carried on the basis to find and stop the intruders in the network such as local and global. Preserving the availability and integrity of networked computing systems in the face of fast-spreading intrusions requires advances not only in detection techniques and also in automated response techniques. Preserving the availability and integrity of networked computing systems in the face of fast-spreading intrusions requires advances not only in detection algorithms, but also in automated response techniques. In this paper, we propose a new approach to automated response called the response and recovery engine (RRE). Our engine employs a game-theoretic response strategy against adversaries modeled as opponents in a two-player Stackelberg stochastic game. The RRE applies attack-response trees(ART) to analyze undesired system-level security events within host computers and their countermeasures using Boolean logic to combine lower level attack consequences. In addition, the RRE accounts for uncertainties in intrusion detection alert notifications. The RRE then chooses optimal response actions by solving a partially observable competitive Markov decision process that is automatically derived from attack-response trees. To support network-level multiobjective response selection and consider possibly conflicting network security properties, we employ fuzzy logic theory to calculate the network-level security metric values, i.e., security levels of the system's current and potentially future states in each stage of the game. In particular, inputs to the networklevelgame-theoretic response selection engine, are first fed into the fuzzy system that is in charge of a nonlinear inference and quantitative ranking of the possible actions using its previously defined fuzzy rule set. Consequently, the optimal network-levelresponse actions are chosen through a game-theoretic optimization process. Experimental results show that the RRE, using Snort's alerts, can protect large networks for which attack-response trees have more than 500 nodes.

Keywords: Stackelberg game, ART trees, RRE engine, Markov Decision making, fuzzy rule set. Intrusion response systems, network state estimation.

I. INTRODUCTION

The network is in the order of increasing size in day to day life hence the security of the network is to be affected in great manner. IP fragmentation, SMTP mass mailing, DoS attacks, flood attacks, spoofing, buffer overflow are some of the attacks that occur in the network.There is other serious threat in network considered to be Intrusion. Intrusion is an action or instance of intruding or an unwelcome visit or a set of actions aimed to compromise integrity, confidentiality, or availability, of a computing as well as networking resource. that is an intrusion on one's privacy.in order to detect the intrusions the systems of intrusion detection, prevention and response systems are needed.

This paper is built upon our previous work [4]. In this paper, we present an automated cost-sensitive intrusion response system called the response and recovery engine (RRE) that models the security battle between itself and the attacker as a multistep, sequential, hierarchical,nonzerosum,two-player stochastic game. In each step of the game, RRE leverages a new extended attack tree structure, called the attack-response tree (ART), and received IDS alerts to evaluate various security properties of the individual host systems within the network. ARTs provide a formal way to describe host system security based on possible intrusion and response scenarios for the attacker and response engine, respectively. More importantly, ARTs enable RRE to consider inherent uncertainties in alerts received from IDSes (i.e., false positive and false negative rates), when estimating the system's security and deciding on response actions. Then, the RRE automatically converts the attackresponse trees into partially observable competitive Markov decision processes that are solved to find the optimal response action against the attacker, in the sense that the maximum discounted accumulative damage that the attacker can cause later in the game is minimized. It isnoteworthy that despite the mathematical. In RRE that itself requires some time to complete in practice,RRE's ultimate objective is to save/reduce intrusion response costs and the system damages due to attacks compared to existing intrusion response solutions. Using this gametheoretic pproach, RRE adaptively adjusts its behaviour according to the attacker's possible future reactions, thus preventing the attacker from causing significant damage to the system by taking an intelligently chosen sequence of actions. To deal with security issues with different granularities, RRE's two-layer architecture consists of local engines, which reside in individual host computers, and the global engine, which resides in the response and recovery server and decides on global response actions once the system is not recoverable by the local engines. Furthermore, the hierarchical architecture improves scalability, ease of design, and performance of RRE, so that it can protect computing assets against attackers in large-scale computer networks. To support network-level intrusion response where the global security level is often a function of different specific properties and business objectives, RRE employs a fuzzy control-based technique that can take into account several objective functions simultaneously. In particular, reports from local engines are fed into the global response engine's fuzzy system as inputs. Then, the RRE calculates quantitative scores of the possible network-level response actions using its previously defined fuzzy rule set.The fuzzy rule set is defined using fuzzy numbers, and hence, various input parameters can take on qualitative values such as high or low; therefore, the real-world challenge that accurate crisp values of the involved parameters are not always known is addressed completely.

The IDS is used in order to improve the security of the network by finding suspicious activities, whether the network is of local or global, the security should be provided in a great manner. In the case of local network the size of the network is small hence the detection can be done with the incoming and outgoing data packets effectively. But in the case of the global network, the size increases hence the IDSis to be performed in the deep manner. Intrusion detection has been made automated in the network that finds whether the user is authorized or an intruder by the default characterises and details. As the network grows larger the intrusion response is also needed to be automated in order to provide the response as soon as possible.

RRE extends the state of the art in intrusion response in three fundamental ways. First, RRE accounts for planned adversarial behavior in which attacks occur in stages in which adversaries execute well-planned strategies and address defense measures taken by system administrators along the way. It does so by applying game theory and seeking responses that optimize on long-term gains.Second, RRE concurrently accounts for inherent uncertainties in IDS alert notifications with attack-response trees converted to a partially observable Markov decision process that computes optimal responses despite these uncertainties.

This is important because IDSes today and in the near future will be unable to generate alerts that match perfectly to successful intrusions, and response techniques must,therefore, allow for this imperfection to be practical. Third,for ease of design purposes, RRE allows network security administrators to define high-level network security properties through easy-to-understand linguistic terms for the particular target network. This is a crucial facility that RRE provides, because unlike system-level security properties,for example, the web server availability, which can be reused across networks, the network-level security properties usually should be defined specifically for each network by the security administrators manually. RRE achieves the above three goals with a unified modeling approach in which game theory and Markov decision processes are combined. We demonstrate that RRE is computationally efficient for relatively large networks via prototyping and experimentation, demonstrate that it is practical by studying commonly found power grid critical infrastructure networks. However, we believe that RRE has wide applicability to all kinds of networks.

II. METHODS AND MATERIAL

A. Existing System

There are many detection techniques used in the network in order to find the misbehaviour and the intruder .The unauthorized login and the usage of the network lead to loss of the information and the blocking of the information in the needed time.EMERALD [11], a dynamic cooperative response system, introduces a layered approach to deploy monitors through different abstract layers of the network. Analysing IDS alerts and coordinating response efforts, the response components are also able to communicate with their peers at other network layers. AAIRS [12] provides adaptation through a confidence metric associated with IDS alerts and through a success metric corresponding to response actions. Though EMERALD, AAIRS and other offer great infrastructure for automatic IRS, they failed to balance intrusion damage and recovery cost. LADS [5], a host-based automated defense system, uses a partially observable Markov decision process to account for imperfect state information; however, LADS cannot be applicable in general-purpose distributed systems due to their reliance on local responses and specific rofilebased IDS. Balepin et al. [13] address an automated response-enabled system that is based on a resource type hierarchy tree and a directed graph model called a system map. Both LADS and the IRS in [13] can be exploited since none of them takes into account the malicious attacker's potential next actions while choosing response actions. Lye and Wing [16] use a game-theoretic method to analyze the security of computer networks. The interactions between an attacker and the administrator are modelled as a two-player simultaneous game in which each player makes decisions without the knowledge of the strategies being chosen by the other player; however, in reality, IDSes help administrators probabilistically figure out what the attacker has done before they decide upon response actions, as in sequential games. AOAR [14], created by Bloem et al., is used to decide whether each attack should be forwarded to the administrator or taken care of by the automated response system. Thus the use of a single step game model makes the AOAR vulnerable to multistep security attacks in which the attacker significantly damages the system with an intelligently chosen sequence of individually negligible adversarial actions.There are many limitations in the above techniques which that include more cost of the systems and the decisions and response are done by the predefined rules hence the intruder with a new strategy are cannot be guessed. To overcome the above disadvantages the concept of RRE engine is developed with the game theory.

B. Proposed System

We formulate the optimal response selection as a decisionmaking problem in which the goal is to choose the costoptimal response action at each time instant. The optimal action m is picked out of the set of all possible response actions m2M, including the No-OPeration (NOP) action.For example, an intrusion response system can respond to SQL's buffer verflow exploitation by closing its TCP connection. The optimization problem is solved in the response system, given the following inputs:

W: a set of the computing assets w 2W, for example, an SQL server, that are to be protected by the response engine.

O: a set of IDS alerts o 2 O that specifically indicate an adversarial attempt to exploit the existing specific vulnerabilities of the assets, for example, alerts from Snort warning about a packet transferring the Slammer worm that exploits a buffer overflow vulnerability in an SQL server.

G: a set of ART graphs g 2 G that systematically define how intrusive (responsive) scenarios about the attacker (response engine) affect system security.

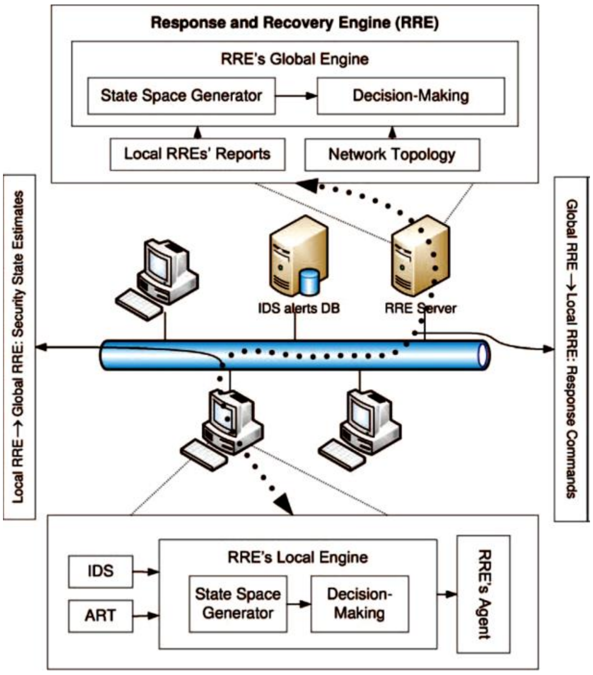
The following sections are devoted to a solution to the response selection problem; in other words, we will focus on how the RRE finds the optimal response action based on given input arguments.

III. RESULTS AND DISCUSSION

A. RRE Architecture

It has two types of decision-making engines at two different layers, i.e., local and global. This hierarchical structure of RRE's architecture, as discussed later, makes it capable of handling very frequent IDS alerts,and choosing optimal response actions. Moreover, the twolayer architecture improves its scalability for large-scalecomputer networks, in which RRE is supposed to protect a large number of host computers against malicious attackers.Finally, separation of high- and low-level security issues significantly simplifies the accurate design 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.

The security maintenance of computer networks is given by Stackelberg stochastic two-player game in which the leader and follower try to maximize their own benefits by taking optimal responses and actions. The system provides more security by the means of the game. The game type called sliding puzzle is used. The uthentication process is made of with the double iteration, in the sense of both the password and the game are considered for the authentication purpose.

If the user needs to access the server for first time the server provides with registration process. The process includes the details of the user that to be filled for the security purpose and the process asks the user to solve the puzzle game that provided with the list of sequence hints which is to be followed by the user in order to solve the puzzle. The game will be provided by the administrator of the server. After the successful registration process the password and the game sequence are mailed to the client's email which makes the reduction of the remembrance of the password.

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 requirementfor the overall system deployment.

1. Local Response And Recovery Engine

Attack-response tree. To protect a local computing asset, its corresponding local engine first tries to figure out what security properties of the asset have been violated as result of an attack, given a received set of alerts. Attack trees [6] offer a convenient way to systematically categorize the different ways in which an asset can be attacked. Local engines make use of a new extended attack tree structure, called an attackresponse tree (ART), that makes it possible 1) to incorporate possible countermeasure (response) actions against attacks,and 2) to consider intrusion detection uncertainties due to false positives and negatives in detecting successful intrusions,while estimating the current security state of thesystem. 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.

The purpose of an attack-response tree gw 2 G for an asset w 2 W is to define and analyze possible combinations of attack consequences that lead to violation of some security property of the asset. This security property, for example, integrity, is assigned to the root node of the tree that is also called the top-event node. In the current implementation of RRE's local engines, there are at most three ART graphs Gw ¼ fgc w; gi w; ga wg for each asset w, which are typically concerned with confidentiality, integrity, and availability of assets; Gw _ G can be expanded to include other security properties. An attack-response tree's structure is expressed in the node hierarchy, allowing one to decompose an abstract attack goal (consequence) into a number of more concrete consequences called subconsequences.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 l 2 L is mapped to (reported by) its related subset of IDS alerts Ol _ O, each of which represents a specific vulnerability exploitation attempt by the attacker.

Starting from the root node and recursively using ART, it is simple to obtain _g, i.e., that is the probability that the security property of the root node in ART graph g has been compromised. This value, as a local security estimate, is reported by the local engine to the RRE server, where optimal global response actions are decided upon according to received local estimates . Next, we will explain how ART graphs and their nodes' satisfaction probabilities are used in a game-theoretic algorithm to decide on the optimal response action. Specifically, the game is a finite set of security states S that cover all possible security conditions that the system could be in.

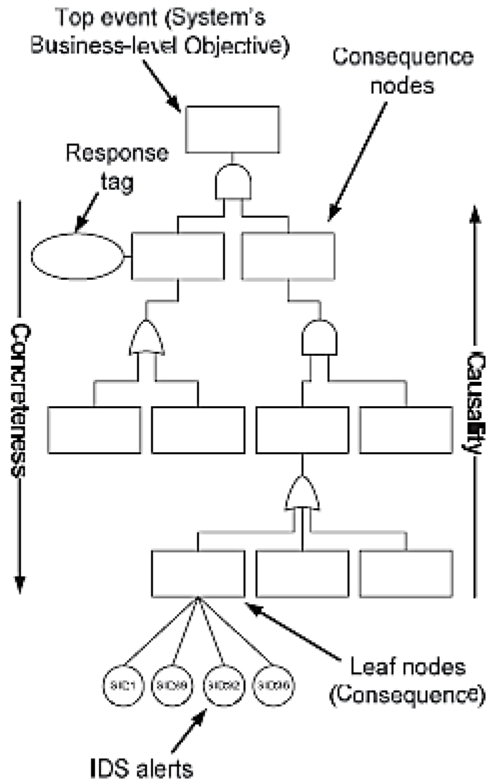


Figure 1. Attack Response Tree

The system is in one of the security states s at each time instant. RRE, the leader, chooses and takes a response action ms 2M admissible in s, which leads to a probabilistic security state transition to s0. The attacker, which is the follower, observes the action selected by the leader, and then chooses and takes an adversary action os0 2 O admissible in s0, resulting in a probabilistic state transition to s00. At each transition stage, players may receive some reward according to a reward function for each 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 multicontroller Markov decision process, called a competitive Markov decision process (CMDP)

So far, we have discussed how RRE's local engine estimates local security state and decides upon and takes local response actions following alerts received from the IDS. Next, we will address how RRE's server makes use of local information received from local engines to estimate the security status of the whole network, and then decide what global response actions to take. The information that are sent by local engines to RRE's server consist of root probabilities _g, as computed in (3), of local ART graphs. In the current implementation of RRE, these include three root node probabilities of three ART trees reflecting confidentiality, integrity, and availability of local host systems.

Agents. In the above-mentioned security battle between RRE and the adversary, agents play a key role in accomplishing each step of the game. They are in charge of taking response actions decided on by RRE engines Actually, having received commands from engines, agents try to carry them out successfully and report the result,whether they were successful or not, back to the commander, i.e., the engine. If the agent's report indicates that some response action has been taken successfully, the engines update their ART trees' corresponding variables, which are leaf node values in the subtree for the successfully taken response action node. Consequently, as explained above, leaf node variables in ART trees are updated by two types of messages: IDS alerts and agents' reports.

2. Global Response And Recovery Engine

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.

➢ Automatic CMDP Generation

To generate the CMDP model, RRE analyzes the network topology input to find out about the set of known system vulnerabilities and individual host computers, i.e., privilege domains. Given the set of system vulnerabilities, the connectivity matrix is updated accordingly to encode adversarial paths only. In particular, RRE automaticallygenerates a CMDP by traversing the connectivity matrix and concurrently updating the CMDP. First, RRE creates the CMDP's initial state ð_Þ and starts the CMDP generation with the network's entry point (Internet) node in the connectivity matrix. Considering the connectivity matrix as a directed graph, RRE runs a depth-first search (DFS) on the graph. While DFS is recursively traversing the graph, it keeps track of the current state in the CMDP, i.e., the set of privileges already gained through the path traversed so far by DFS. When DFS meets a graph edge ½i; j_ that crosses over privilege domains wi to wj, a state transition in CMDP is created if the current state in CMDP does not include the privilege domain of the host to which the edge leads, i.e.,wj. The transition in CMDP is between the current state and the state that includes exactly the same privilege set as the current state plus the host wj directed by the graph edge ½i; j_. The CMDP's current state in the algorithm is then updated to the latter state, and the algorithm proceeds until no further updates to CMDP are possible according to the connectivity matrix.

➢ Multiobjective System Security Reward Function

Local engines send their local security estimates, i.e., root node probabilities _g of their ART graphs, to the RRE server.RRE considers the etwork's global security as a multiobjective reward function for the response selection procedure. Each objective is represented by a specific system-level security property, and quantified by the _g values, which are calculated in the local engines. In our multiobjective game scheme, there is usually not a single solution that simultaneously minimizes each objective to its fullest. In each case, we are looking for a solution for which each objective has been optimized to the extent that if we try to optimize it any further, then the other objective(s) will suffer as a result. RRE makes use of a fuzzy-logic based controller that merges the involved objective function values using an information fusion algorithm according to the network security definition, and consequently, result in a single scalar reward value.

Fuzzy logic is a form of multivalued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. In contrast with binary sets which follow the binary logic, the fuzzy logic variables may have a membership value of not only 0 or 1. Just as in fuzzy set theory, with fuzzy logic, the set membership values can range inclusively) between 0 and 1, and the degree of truth of a statement, for example, The network is currently secure.,can range between 0 : false and 1 : true and is not constrained to only two digital values as in classic propositional logic. In particular, RRE calculates the global network security level, i.e., the truth degree of the "The network is currently secure" predicate, using a fuzzy control system [15] that analyzes analog input values in terms of logical variables (system-level security properties) from local response engines that take on continuous values _g,and produces the network-level security measure values.

Formally, inputs to the fuzzy controller, that is in charge of calculating the global network-level security measure values for individual network security states _n : S ! ½0; 1_, represent root node values of the ART trees within the local response engines _g g 2 G. Before getting into technical details, as a clarifying example scenario, let us consider that the fuzzy controller defines the global network-leve l security as a function of two inputs: AðWSÞ: availability of the web server and IðDBÞ: integrity of the database server. So, given degrees of the system availability, for example, AðWSÞ ¼ high, and system integrity, for example, IðDBÞ ¼ low, in a sample network belief state b, the fuzzy controller computes the security status of the system, for example, _nðbÞ ¼ medium.

IV. CONCLUSION

The proposed system improves the performance of intrusion response. Using the proposed system the system can yield the advantages as scalability in which the system can be applied to any global area and the security in the manner of prevention, detection and response for intrusion are increased. The user can access the server with easy accessibility. The server can be reached easily and the main importance and advantage of the game is that it avoids the password remembrance.

V. REFERENCES

[1] Devi Parikh, Tsuhan Chen,"Data Fusion and Cost Minimization for Intrusion Detection". IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 3, NO. 3, SEP 2008 pp 381-389

[2] Fu-Wen Chen and Jung-Chun Kao "Game-Based Broadcast over Reliable and Unreliable Wireless Links in Wireless Multihop Networks" IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 12, NO. 8, AUG 2013 pp 1613-1624

[3] Kai Hwang, Min Cai, Ying Chen,and Min Qin"Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes. "IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 4, NO. 1, JAN-MAR 2007. Pp 41-55

[4] Nicola Basilico, Nicola Gatti, Mattia Monga, and Sabrina Sicari2014 "Security Games for Node Localization through Verifiable Multilateration " IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 11, NO. 1, JAN / FEB pp 72-85

[5] O. Patrick Kreidl, and Tiffany M. Frazier, "Feedback Control Applied to Survivability: A Host-Based Autonomic Defense System."IEEE TRANSACTIONS ON RELIABILITY, VOL. 53, NO. 1, MAR 2004. pp.148-166,

[6] Paul C. van Oorschot, Amirali Salehi-Abari, and Julie Thorpe" Purely Automated Attacks on PassPoints Style Graphical Passwords "IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,VOL. 5, NO. 3, SEP 2010 pp 393-405

[7] Shi-Jay Chen and Shyi-Ming Chen,"Fuzzy Risk Analysis Based on Similarity Measures of Generalized Fuzzy Numbers." IEEE TRANSACTIONS ON FUZZY SYSTEMS, VOL. 11, NO. 1, FEB 2003. Pp 45-56

[8] Tatyana Ryutov, Clifford Neuman, Dongho Kim, and Li Zhou "Integrated Access Control and Intrusion Detection for Web Servers." IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,VOL. 14, NO. 9, SEP 2003.pp 841-841

[9] Vivek Raghunathan and P.R. Kumar "Wardrop Routing in Wireless Networks ", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 8, NO. 5, MAY 2009 pp 636-652

[10] Zhenxin Zhan, Maochao Xu, and Shouhuai Xu"Characterizing Honeypot-Captured Cyber Attacks:Statistical Framework and Case Study" IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 11, NOV 2013 pp 1775-118,

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

[12] 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, 2000. pp. 2344-2349,

[13] 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, 2003.14M. Bloem, T. Alpcan, and T. Basar, "Intrusion Response as a Resource Allocation Problem," Proc. Conf Decision and Control, pp. 6283-6288, 2006

[14] saman a. zonouz, himanshu khurana, william h. Sanders and timothy m. yardley"RRE: a game-theoretic intrusion response and recovery engine" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, vol. 25, no. 2, february 2014 pp 395-406.

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