Journal of Computer Sciences and Applications, 2013, Vol. 1, No. 2, 27-32
Available online at http://pubs.sciepub.com/jcsa/1/2/3
© Science and Education Publishing
DOI:10.12691/jcsa-1-2-3

Modeling Attacker-Defender Interaction as a Zero-Sum Stochastic Game

Ibidunmoye E.O.[*], Alese B.K., Ogundele O.S.

Department of Computer Science, Federal University of Technology, Akure, Nigeria
*Corresponding author: oloruntoba78@yahoo.com

Received January 18, 2013; Revised March 13, 2013; Accepted April 17, 2013

Abstract Game-theoretic modeling of computer security views security attack scenarios as an optimization game comprising of multiple players notably the attackers and the defenders (system administrators). This paper first presents theoretically, a two-player zero-sum stochastic game model of the interaction between malicious users and network administrators and secondly introduces a hypothetical network of a typical scenario to show the applicability of our model within that scenario. State games are encoded using a binary scheme in order to properly capture components of the underlying network environment. Our solution involves reducing each state game into a min and max linear programming problems for both the defender and attacker respectively. Game costs, rewards and outcomes are modeled to closely match real world measurements. We propose the use of a combination of the pivotal algorithm and a custom stochastic algorithm to compute the optimal (best-response) strategies for the players at each state. We also describe how the results can be analyzed to show how the optimal strategies can be used by the network administrators to predict adversary's actions, determine vulnerable network assets and suggest optimal defense strategies.

Keywords: *security games, strategies, attackers, defenders, stochastic games, game theory*

1. Introduction

Information technology, having assumed the position of being the driving force behind modern society, is constantly changing the way we live, share and communicates with others. The new paradigms of ubiquitous computing and high capacity data transfer have turned the Internet into today's main medium for information interchange and electronic commerce [8]. As a result, our strongly-connected world has continually been plagued with myriads of security threats due to the pervasiveness of computer networks spurred by the Internet.

In order to study the mechanism of attack propagation, a number of closely related attack modeling techniques have been developed. Attack graphs are one of such techniques, used to study how an attacker can combine vulnerabilities to stage an attack [14]. Central to attack graphs analysis are the attacker's goal and methods, and so can easily be used to reveal the true scope of threats by mapping the sequences of attacker's exploits that can penetrate the network [17]. Though attack graphs encourages informed risk assessment process and form the basis for optimal network defense, their growth can be exponential and lack the capability to predict attackers set of moves and possible counter-measures.

Network security, when viewed from a game theoretic perspective, can be seen as a game comprising multiple players; the attackers (malicious users) and the defenders (network/system administrators). The benefit of quantifying network security using game-theoretic approach is enormous. Most importantly it may help network administrator to find the optimal defense strategies of a system and to calculate the expected loss associated with different defense strategies [10].

Security games provide a quantitative framework for modeling the interaction between attackers and defenders. These games and their solutions could serve as a basis for security decision making and algorithm development as well as to predict attacker's behavior [3]. Security games vary from simple deterministic ones to more complex stochastic ones and are applicable to security problems in a variety of areas ranging from intrusion detection to social, wireless, and vehicular networks. In stochastic games the play proceeds by steps from position to position, according to transition probabilities controlled by the two players [16]. Stochastic games aims both to capture the unknown and uncontrollable parameters in security problems and analyze the behavior of rational attackers which is usually represented as a probability distribution over the possible attacks [3].

The paper is organized as follows. We first introduce the theoretical basis of our framework in different sections. Section 2 provides an overview and state-of-the-art of the use of game-theoretic techniques in security analysis. Section 3 introduces the game model, the definition and some useful properties. In section 4 we introduce a simple network environment and possible components of such network. Section 5 documents the process of determining the cost/rewards for strategies chosen by game players and the overall outcomes of the game play. In section 6 we describe how to identify the major game actors, the functions of such actors, and the nature and format of the set of actions available to them. Sections 7 and 8 describe the stateful nature of the stochastic game, state encoding and how to compute state transition probabilities. In section 9 we describe the structure of game matrices, how they can be generated. Section 10 explains how game matrices are transformed into linear programs that can be solved to derive game values and optimal strategies for both players. In section 11, we introduce a simple scenario where our model can be used. This scenario is proposed to serve as a basis for future simulation of our model. Section 12 describes expected results of the simulation of the stochastic game model and how these results can inform the defender on how best to protect the network. Section 13 concludes the paper with relevant future outlook and recommendations.

2. Related Works

Network security has gained significant attention in research and industrial communities as a result of the global connectivity provided by the Internet [4]. This has led to a variety of traditional defense mechanisms ranging from cryptography, firewalls, antivirus software, to intrusion detection systems.

Security decisions have recently been investigated analytically. Analytical approaches present a number of advantages compared to heuristic and adhoc approaches [3]. Many mathematical models have been used to model and analyze the decision making problems in security. Machine Learning [2], Control Theory [11], and Data Mining [1] are mathematical models that have been utilized to model security problems. However, these attempts fail to capture the ability of attackers to intelligently choose their targets and alter their attack strategies based on the defensive schemes that are put in place by defenders [6] and so are not suitable for modeling the interaction with dynamic, pro-active, and cognitive adversaries [5,15] provides a formal way of describing the security of a system via his attack trees which are, though novel, often exponentially explosive.

The use of game-theoretic approaches to quantifying security has gained enormous research attention. More recently, Game Theory has been used to study network security problems [3,10,12,13]. Recently there has been increased interest in probabilistic method for quantifying the operational security of networked computer systems [9]. Security games provides the capability of examining hundreds of attack scenarios and offers methods for suggesting several potential course of actions with accompanying predicted outcomes [13]. Computer implementations of those methods can result in intelligent and automated security decision engines that are fast and at the same time scalable. The work of [12] and many others view stochastic games as a non-linear programming problem that could be solved using dynamic programming algorithms, the value iteration algorithm or any other similar approaches. Also, existing works [3,10] nsider attacker-defender interactions as general-sum games. In this paper we investigate how attack scenarios can be analysed as a zero-sum two-player games and the possibility of viewing such as linear programming problems that could be solved using common linear algorithms.

3. The Stochastic Game Model

Consider a two-player zero-sum game played on a finite state space, where each player has a finite number of actions to choose from. We formally define our two-player stochastic game as a tuple as defined in (1).

$$G = (S, P, (A_i, \alpha_i, U_i)_{1 \le i \le |p|}, Q) \tag{1}$$

Table 1. Definition of parameters

	Expression	Description
S	$S = \{S_1, S_1, S_1, \dots S_t\}_{1 \le t \le \lvert s \rvert}$	A finite set of states
P	$p = \{p_k\}_{k=1,2}$	A finite set of players
A_i	$\forall (p_k \in p) \exists A_i$ $= \{a_1, a_2, \dots a_n\}$	For every player p_k there exist a finite set of actions for that player.
a_i	$a : S \to A_i;$ $i = 1, 2$	A mapping that assigns to each state $s \in S$ the set of actions $a_i(s)$ that are available to player i in state s
Sa	$Sa = \{(s, a) : s \in$ $S, a = (a_i), a_i \in$ $a_i(s); 1 \le i \le \lvert p \rvert\}$	The set of all possible action profiles for each player
U_i	$Ui : Sa \to R$ $i = 1, 2$	For every player, U_1 assigns a stage payoff to player P_1 when the corresponding action profile is played
Q	$Q : SA \to P(S)$	Q:is the probability distribution over S

4. The Network Environment

A typical stochastic security game scenario is played over a computer network environment made up of several interconnected components (assets) and game actors. Network assets may include firewalls, database, file/print, application servers, routers, cryptographic devices etc. The game actors often are network/virtual users, normal users attempting to accomplish a task, attackers who exploit vulnerabilities and defenders whose responsibility is to secure the network from malicious threats to both internal and external factors.

5. Rewards, Costs, and Outcomes

Attacker’s actions are mostly associated with rewards measured in the amount of damage done to any network asset, while defenders mostly have loss in terms of cost. When an attacker successfully wreck havoc on a network component, it may take the *defender* say *X* to *Y* minutes to figure out which service or component is affected and restore it to operation. Meanwhile, the *attacker* may use the same period of probing to propagate or exploit another vulnerability. Thus, this amount of time is a loss to the defender and a gain to the attacker. Therefore, in this work, the attacker's rewards are defined in terms of the amount of time required by the defender to put the affected asset to a working state.

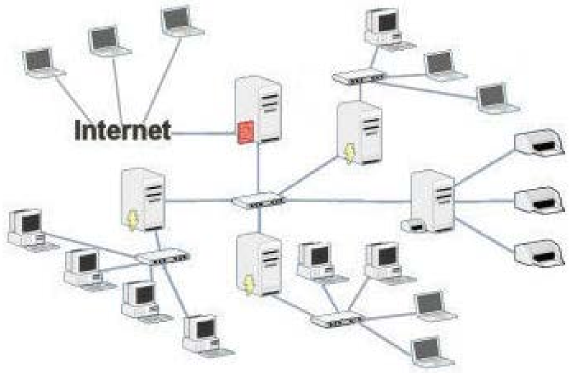

Figure 1. Typical Network Environment

In our case we assign cost to network components depending on their perceived value. That is the higher the perceived importance of an asset the higher it's assigned cost. The value of an asset as aforementioned is valued according to the amount of time required to restore such to an operative state by the defender.

For our two-person finite zero-sum game is considered the magnitude of the reward to the *attacker* is the same as the loss of the *defender*. That is, if it takes asset Z an average X minutes to be restored after a breakdown then the cost of asset C is minutes while the defender's loss and attacker's gain are is –*X* and +*X* minutes respectively. Thus satisfying the zero-sum property X+(–*X*)=0.

6. Modeling Game Actors and Actions

Actors in a game are the players whose intents are to either maximize gains or minimize losses. The game actors in this model are the attacker and the defender. The attacker abstracts one or multiple entities with malicious intent to compromise a computer network. Such encompasses professional hackers, disgruntled staffs, malicious users, and malicious nodes while the defender abstracts one or more entities such as system administrators and intelligent nodes entrusted with the responsibilities to protect the network and its assets and make timely security decisions.

We represent players of the game as p_1, the *defender*, and p_2, the *attacker*. The action spaces of the players are the sets of possible attack moves and defense counter measures respectively. This model encapsulates each attack or defense as a single action achieving a specific goal. Therefore, the finite action spaces for both the *defender* (p_1) and *attacker* (p_2) defined as in equations (2) and (3) respectively

$$A_1 = P_1^a = \{a_1, a_2 \ldots a_n\} \tag{2}$$

$$A_2 = p_2^a = \{a_1, a_2, \ldots a_n\} \tag{3}$$

At every state of the game, players have at their disposal a finite set of actions to choose from, the nature of the configuration of the network determines if this actions are unique across states or not.

For example, an attacker action space may be

$$\{scan_{webserver}, attack_{httpd}, attack_{database-service}\}$$

while for a *defender* could be

$$\{monitor_{firewall}, patch_{httpd}, restart_{database-service}\}$$

7. Modeling Game States

Stochastic security games are played between players on a finite state space (representing the environment upon which the game is played) that moves probabilistically from state to state. We adopt [3] idea of state as an operational mode of the networked system, in which units are fully, partially operational, or completely out of operation. [12] modeled the state of a network as one containing various kinds of information or features such as type of hardware, software, connectivity, bandwidth and user privileges. Our game transits from one state to another according to a probability distribution. The state transition probability is a function of both the players' actions, and the current state. These probabilities do not only determine state movements they are also incorporated into a solution method to influence both the value of the game and the optimal mixed strategies for the players.

A Stochastic game G, consists of a finite set of states or positions

$$S = \{s_1, s_2, s_3 \ldots, s_t\}_{1 \le t \le |s|}$$

that represent the underlying network environment, one of which is assigned the start state. Associated with each state S_k is a matrix game G^k. Transitions from state S_k to another S_l depends on the outcome of G^k and a probability $P(S_l)^k$ interpreted as, at state S_k, the game transit to state S_1 with a probability $P(S_l)^k$. Where P^k is a probability distribution over the state space and so it holds that $0 \le P^k \le 1, \ \sum_{k=1}^{|s|} P^k \forall k$.

In practical cases, there exist some state transitions that are infeasible. For instance, it may not be possible for the network to move from a normal operation state to a completely shutdown state without traversing some intermediate states. Such infeasible states are assigned zero probabilities and are ignored in this model. Therefore, given state space *S*, there exist a state S' where $S' \subset S$ considered feasible for both the attacker and defender while the remaining states in set *S* are infeasible.

8. Encoding Game States

The choice of encoding scheme is a factor of the problem and the complexity of the network under modeling. For complex networks (such as the Internet), the components and interconnections are modeled as nodes/vertices and link/edges of a giant graph network. For small-scale networks (e.g. intranets), we propose a linear binary representation scheme. However the choice of encoding is also influenced by the solution method chosen for the game. The binary representation scheme encodes a state as a binary string of zeroes (0's) and ones (1's) of length equal to the number of network components. Each component is represented with a **1 (ON)** if in operation and **0 (OFF)** if not. That is a possible state of 3-component network could be **101**. Therefore for a network For a binary scheme, the total number of states could be easily computed using

$$N = 2^k + 1 \tag{4}$$

With respect to equation (4), the total number of states for a network with *K=5* is computed as

$$N = 2^K + 1 = 2^5 + 1 = 32 + 1 = 33$$

We generate the sequence of the bits in each state string according to a priority indicate the security index of an asset and the position of such asset in the network. This ordering also describes how packets transverse the network and also influences the order of transitions of states in the game.

9. Generating Game Matrices

In this model, all state games are in their strategic form represented as a two-dimensional matrix. We designate the *defender* the row player, while the *attacker* is the column player. The elements of the matrices are payoffs to be either gained or lost when each player play the corresponding action in their strategy profile for that state. The base matrix (start game) is purely deterministic while subsequent state matrices are mostly probabilistic because of the influence of transition probabilities. Associated with a state S_K is a matrix $G^{(k)}$ described by equation (5)

$$G^{(K)} = (a_{i,j}^k + \sum_i^N P_i^{(l)} G^{(L)}) \ for\ k = 1 \ldots N \tag{5}$$

At each state *k*, players simultaneously choose a row *i* and a column *j* of the state matrix causing the attacker to win the amount $a_{i,j}^k$ from the defender who apparently looses same amount and with a probability that depends on *i*, *j* and the state, the game either stops or moves to another state or itself. The probability that the game ends at state *k* is denoted as S^k and the probability that the next state is *l* is denoted by $P_i^{(l)}(l)$ [7]. Therefore, it suffice to say that

$$s^k + \sum_i^N P_i^{(l)}(l) = 1 \tag{6}$$

Also, $p^{(l)}$ is the total probability that the game can go to state from any state i.e. $p^{(l)} = \sum_i^N P_i^{(l)}$

To generate the state matrices, we look at defining the payoffs from the perspective of the defender since our interest lies in analyzing the defender's game. We value each asset as the amount of time (perceived or measured) it takes to it back to a working state after an attack. This value could also be referred to as the mean time to repair of the asset. It is believed that when an attacker successfully compromise an asset she's gains an amount of time equal to the mean time to repair such asset and can take that time-advantage to propagate another attack.

We use the following methodology to determine elements of the base matrix. Let *A* be the asset that *attacker's* action a_i affects, so *C* can be defined as the MTTR of asset A. Also, let *B* be the asset that *defender's* action d_j affects, then *K* can be defined as the MTTR of asset B. Therefore, suffix to say

$$U = \begin{cases} C+K & i \neq j \\ C & otherwise \end{cases} \tag{7}$$

The resulting bi-matrix therefore contains the game matrix for both the *attacker* and *defender*. For this model the intent is to analyse defender's moves against the attacker's, so the defender's component of the bi-matrix is extracted. The base (starting game) matrix is captured as a bi-matrix in (8);

$$G = (a_{i,j}, -a_{i,j}) \tag{8}$$

where for 0< *i* < *m*, 0 < *j* < *n*, $m = \left|p_1^a\right|, and\ n = \left|p_2^a\right|$

10. Computing Game Values and Optimal Strategies

According to Shapley (1952) associated with each state s_k is a matrix game $G^{(K)}$ and each game $G^{(K)}$has a value *V(K)* [7]. For all games matrices, the game values are the unique solutions of (5) with game values given as (10)

$$V(k) = Val(a_{i,j}^k + \sum_i^N p_i^{(l)} V(l)) \tag{10}$$

Stochastic games are characterized by games that may themselves have other games as components where the outcome of a particular choice of pure strategies of the players may be that the players have to play another game depending on some probability. We use this knowledge as a way of modeling transitions between states. To get the solution of such games, our algorithms has to recursively iterate over each game to obtain its value. [7] notes that if the matrix of a game *G* has other games as component, the solution of *G* is the solution of the game whose matrix is obtained by replacing each game in the matrix of *G* by its value.

Every finite 2-person zero-sum game has a value, called the value of the game. The value of the game can be defined in terms of the *min-max* theorem

"There is a mixed strategy for player I such that I's average gain is at least V no matter what II does and there is a mixed strategy for Player II such that II's average loss is at most V no matter what I does. Also, If V=0, the game is fair. If V>0 the game is said to favour Player I, otherwise if V<0 the game favours player II" [7].

The first step to solving each state game is to determine if there exists a saddle point, if it does the value of the game is the saddle point. If not, we convert the matrix game into a linear programming problem that could be solved using any linear programming (LP) solution method. Next, each game matrix in the defender's game is converted to a *min* linear programming (LP) problem that is then solved using a variant of the Simplex Algorithm called the Pivot Method. The linear programs are constructed in a way that minimizes the payoff of the defender i.e. the average loss of the defender as well as minimizes the average gain of the attacker. According to [7], the following LP ensures that his average gain is *v*;

Choose v and $p_1,\ldots, p_m$ to maximize v.
Subject to the constraints

$$v \le \sum_{i=1}^{m} p_i a_{i1} \ldots\ldots\ldots v \le \sum_{i=1}^{m} p_i a_{in} \qquad (11)$$

$$p_1 + \ldots + p_m = 1,\ p_i \ge 0\ \ for\ i = 1,\ldots,m$$

Similarly, the dual of the above program gives the LP problem for the defender, ensuring that his average loss is *v*;

Choose w and $p_1,\ldots, p_m$ to maximize v.
Subject to the constraints

$$w \le \sum_{j=1}^{n} p_j a_{1j} \ldots\ldots\ldots w \le \sum_{j=1}^{n} p_j a_{mj} \qquad (12)$$

$$p_1 + \ldots + p_n = 1,\ p_j \ge 0\ \ for\ \ j = 1,\ldots,n$$

The expected output are two vectors representing the optimal mixed strategies for both the attacker and the defender at each state of the game, and a vector of real game values containing the values of games played in all states.

The optimal mixed strategies produced by this algorithm can be represented as;

$$\begin{aligned} X^* &= \{\mathrm{p}=(\mathrm{p}_1,\ldots,\mathrm{p}_m){:}0 \le \mathrm{p}_i, \\ &\mathrm{p}_i \le 1 \forall i = 1,\ldots,m \ and \sum_{i=1}^{m} p_i = 1\} \end{aligned} \qquad (13)$$

$$\begin{aligned} Y^* &= \{\mathrm{q}=(\mathrm{q}_1,\ldots,\mathrm{q}_n){:}\ 0 \le \mathrm{q}_i, \\ &\mathrm{q}_i \le 1 \forall i = 1,\ldots,n \ and \sum_{i=1}^{n} q_i = 1\} \end{aligned} \qquad (14)$$

Also, the expected vector of game values is represented as follows;

$$V = (v(0), v(1), \ldots\ldots, v(N))$$

where *N* is the number of states.

11. A Network Scenario

In this section we introduce a simple game scenario and how model components maps to this scenario. We however, leave out the simulation of the model using this scenario. We intend to address this and the development of the custom stochastic game algorithm in subsequent works.

Figure 2 presents a simple network scenario showing network assets and players. On one side is the network defender(s) usually the network administrators designated to manage and protect the network. On the other side are the users, assessing the network over the internet (and possibly over a LAN) to access network resources. Within this set are the malicious users, the attackers, whose intent is to thwart the operation of the network by exploiting its vulnerability. In the secured perimeter are the network infrastructure (assets) which the database, file, application servers and the firewall. Other possible assets are routers, cryptographic devices etc.

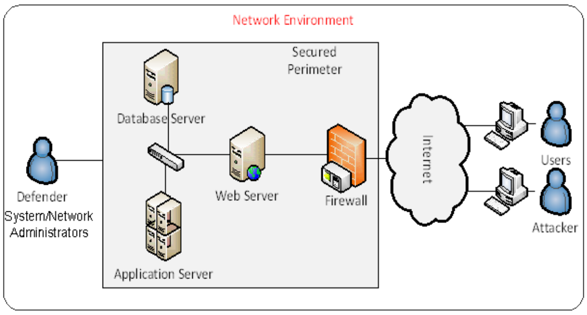

Figure 2. Typical Network Scenario

In this scenario, we identify game players as the defender (administrators) and attacker (malicious user). Next we prioritize the network components (according to their perceived value) as follows *firewall, web server, application server*, and *database server*. We then assign to each component a practical MTTR (obtained from real life systems) and initialized the set of actions available to each player as follows;

$$\begin{aligned} p_1^a = \{&restart \,|\, patch_{firewall,} restart \,|\, path_{webserver,} \\ &restart \,|\, patch_{appserver,} restart \,|\, path_{dbserver,} end_{game}\} \end{aligned}$$

$$\begin{aligned} p_2^a = \{&attack_{firewall,} attack_{webserver,} attack_{appserver,} \\ &attack_{dbserver,} end_{game}\} \end{aligned}$$

Where p_1^a and p_2^a belongs to the defender and attacker respectively. According to equation (4) the number of states for this network game will be 17! In future simulation we shall attempt to generate (a) transition diagrams showing the probabilistic game movements; (b) game matrices and how these are transformed to linear programs using basis introduced in section 9; (c) custom stochastic algorithm that will be used for simulating the network to compute optimal strategies especially for the defender.

12. Result and Discussion

At every state, there exists an optimal pair of vectors X^*, Y^* generated by the algorithm and there exist an element in both X^* and Y^* with the highest probability value. These high probabilities indicate that corresponding actions in the action sets for both players are optimal. The reason for that is in the rationality of the players, since defenders make their moves in response to that of the attackers, and so they tend to make moves that will minimize their average loss regardless of the actions taken by the attackers. However, the attacker too may change the dynamics of the game by conspicuously ignoring the assets that defenders may possibly fortify (assets directly affected by the action having the maximum optimal strategy) and instead attack those assets with next highest optimal strategy. Nevertheless, the defender at the same time may, while defending the most vulnerable asset, also fortify asset with next highest optimal strategy.

The vector of game values *V*, helps analysts to determine the nature of the game at each state. It helps to identify if the game favours the defender or the attacker. For the defender's game vector elements indicate the average loss of the defender for the corresponding state while for an attacker's game it depicts average attacker's gain. When these dynamics is observed and analysed over all game states, the defender can easily determine the most vulnerable network assets, the possible attacker's behaviour and the corresponding counter-measures.

13. Conclusions

Stochastic modeling of computer networks allows researchers to be able to model and analyse the both defender's and attacker's behaviour with respect to underlining system environment. This work presents a quantitative method for analysing network security using stochastic modeling technique. The method has demonstrated how the real-time behaviour of the system in response to player actions can be assessed. It has also been shown how the complexity of network components, the dynamic nature of underlying network environment, and probabilistic nature of player strategies can be captured in one model to predict the behaviours of players. By computing and analysing the optimal mixed strategies of the games, it has been shown the possibility of predicting adversary's attacks, determine the set of assets that are most likely to be attacked, and possibly suggest defense strategies for the defender.

In future works, we intend to carry out a full scale simulation use our custom stochastic game algorithms to achieve the deliverables identified in section 11. Also, in order to properly model threats/vulnerabilities we shall employ the use of attack graphs to analyse how vulnerabilities are exploited by attackers while stochastic security games shall be used for formal analysis and prediction of adversaries' behaviour. This will serve as a basis for recommending appropriate optimal counter-measures for defenders to better manage the network infrastructures.

References

[1] Adetunmbi A.O. Falaki S.O., Adewale, O.S. and Alese, B.K. (2008) "Intrusion Detection based on rough Set and k- Nearest Neighbour", International Journal of Computing and ICT Research, vol. 2 No. 1. pp. 60-66.

[2] Adetunmbi A.O., Alese B.K., Ogundele O.S. and Falaki S.O. (2007) "A Data Mining Approach to Network Intrusion Detection", Journal of Computer Science & its Applications, vol. 14 No. 2.pp 24-37.

[3] Alpcan T. and Baser T (2010), "Network Security: A Decision and Game-Theoretic Approach", 1st ed. Cambridge University Press.

[4] Arome G. (2010) "Modelling of Internet Protocol Security Policies in a Networking Environment". M.Tech. Thesis, Department of Computer Science, Federal University of Technology, Akure. Nigeria.

[5] Assane Gueye "A Game Theoretical Approach to Communication Security" (2011), Electrical Engineering and Computer Sciences University of California at Berkeley. Technical Report No. UCB/EECS-2011-19.

[6] Cavusoglu H., Raghunathan S., and Yue W.(2008), "Decision-Theoretic and Game-Theoretic Approaches to IT Security Investment" Journal of Management Information Systems, vol. 25, pp. 281 September.

[7] Ferguson S. T. (2007) "Game Theory II – Two-Person Zero-Sum Games".

[8] Karin Sallhammar (2007) "Stochastic Models for Combined Security and Dependability Evaluation". Ph.D. Thesis, Department of Telematics, FITME, Norwegian of Science and Technology. Trondheim, Norway.

[9] Karin Sallhammar, Knapskog S. J. (2004) "Using Game Theory in Stochastic Models for Quantifying Security" In Proceedings of the 9th Nordic Workshop on Secure IT-systems (Nordsec 2004). Espoo, Finland.

[10] Karin Sallhammar, Knapskog S. J. and Helvik B. E. (2005) "Using Stochastic Game Theory to Compute the Expected Behavior of Attackers", In Proceedings of the 2005 International Symposium on Applications and the Internet (Saint 2005). Trento, Italy.

[11] Khanna R. and Liu H.(2007), "Distributed and Control Theoretic Approach to Intrusion Detection" Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing, ser. IWCMC '07. New York, NY, USA: ACM.

[12] Lye Kong-wei, Jeanette Wing (2002) "Game Strategies In Network Security", Extended Abstract for FCS.

[13] Roy S., Ellis C., Shiva S., Dasgupta D., Shandilya V. and Wu Q (2010). "A Survey of Game Theory as Applied to Network Security". Proc. of the 43rd HICSS, Hawaii.

[14] Ryan Trost (2009) "Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century" Addison-Wesley Professional.

[15] Schneier B. (1999), "Attack trees: Modeling security threats," Dr. Dobb's Journal, December.

[16] Shapley L. S. (1953) "Stochastic Games". Proceedings of the National Academy of Science USA, vol 39, pp. 1095-1100.

[17] Steffan J. & Schumacher M. (2002) "Collaborative Attack Modeling" In proceeding of the Symposium on Applied Computing, Madrid, Spain.