Ibidunmoye et al., J Comput Eng Inf Technol 2013, 2:1
http://dx.doi.org/10.4172/2324-9307.1000103

Journal of Computer Engineering & Information Technology

Research Article

A SCITECHNOL JOURNAL

A Game-theoretic Scenario for Modelling the Attacker-Defender Interaction

Ibidunmoye EO[1], Alese BK[1] and Ogundele OS[1*]

Abstract

Existing computer security techniques lack the quantitative decision framework required to defend against highly organized attacks. Game theory provides a set of quantitative and analytical tools for describing and analyzing interactive decision situations in computer security. Recently, game-theoretic approaches such as stochastic security games have been used to study security problems as an optimization game comprising multiple players notably the attackers and the defenders (system administrators). Stochastic security games are a probabilistic approach appropriate for studying particularly complex networks where attacks often go from a state and proceeds to another according to a probability distribution. A stochastic game-model that views the interaction between malicious users and network administrators as a two-player zero-sum game was developed. A binary coding scheme was employed for identifying game states and game transition diagrams were generated to describe possible movements of players. A stochastic algorithm was developed to solve the game and compute the optimal strategies for the players. A simulation of the algorithm was carried out the output analyzed to show the techniques that network administrators can employ to predict adversary's actions, determine vulnerable network assets and suggest optimal defense strategies for the defender.

Keywords

Security games; Strategies; Attackers; Defenders; Stochastic games; Game theory

Introduction

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 [1]. 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. As a consequence, network security has gained significant attention in research and industrial communities as a result of the global connectivity provided by the Internet [2]. This has led to a variety of traditional defense mechanisms ranging from cryptography, firewalls, antivirus software, to intrusion detection systems.

In practice, the act of securing network infrastructures involves decision making activities. Whether it is about examining networks for security vulnerabilities and attacks propagation, or choosing the right security policy and mechanism for the network; systems administrators are often required to make decisions that often involve the allocation of scarce security resources to guard the network infrastructure. Such security decision-making procedures and the process of securing systems 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 [4], control theory [5], and data mining [6] are major mathematical models that have been utilized to model security problems. However, these attempts fail to capture the rationality and dynamic nature of players involved in security provisioning large scale networks. In a typical security interaction, there is the possibility of attackers intelligently choosing their targets and alter their attack strategies based on the defensive schemes that are put in place by system administrators guarding the network [7]. As a result, such techniques are not suitable for modeling the interaction with dynamic, pro-active, and cognitive adversaries [8].

Effective provisioning of security services requires that system administrators understand the network, the level of vulnerability, and how attackers exploit and eventually propagate their attacks. In order to determine threat levels and the mechanism of attack propagation, a number of closely related attack modeling techniques have been developed. Schneier B [9] presents a formal way of describing the security of a system via attack trees/graphs which are. Attack graphs are used to study how an attacker can combine vulnerabilities to stage an attack [10]. 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 [11]. 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 [12].

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 [13]. Stochastic games aims both to capture the unknown and uncontrollable parameters in security

***Corresponding author:** Department of Computer Science, Federal University of Technology, Akure, Nigeria, Tel: +2348037752527; E-mail: oloruntoba78@yahoo.com

Received: December 07, 2012 **Accepted:** March 01, 2013 **Published:** March 08, 2013

problems and analyze the behavior of rational attackers which is usually represented as a probability distribution over the possible attacks [3].

The nature of game-theoretic models in security is often very dependent on the problem domain; therefore there exists no generic game-theoretic approach for analyzing security problems. This is more profound when the variability in composition and configuration of computer networks or security systems are considered. Many existing work in this area have considered mostly general-sum games and since general-sum games often result in non-linear often intractable solutions for even moderately sized networks, we introduce a zero sum simplification of the problem. The zero-sum nature of our game makes for easy computation and approximation of players' optimal strategies.

The contributions of this paper are to propose a simpler yet powerful stochastic security game model that results in tractable linear programs; to describe its practical applicability using simple attack-defense scenario. Most importantly, our model is able to consider the rationality of players, dynamic nature of network environments, the importance of heuristics in computing the attack probabilities and most of all how to interpret computed strategies for use in making defense decisions.

The paper is organized in a mixed format. We first introduce the theoretical basis for each section and then describe how this basis is used in our case study. 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 an attack-defense scenario and a sample network environment. 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 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. Using our scenario we identify possible game states, describe how we encode each state and describe the probabilities guiding the movement from one state to another. In sections 8 and 9 we describe how game matrices from each state are generated, and how the resultant linear programs are solved to compute game values and optimal strategies for the players. Sections 10, 11, and 12 describe the system flowchart for the simulation program and how the results of simulation can inform the defender on how best to protect the network. Section 13 concludes the paper with relevant recommendations.

Related Works

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 [3,12,14,15]. Recently there has been increased interest in probabilistic method for quantifying the operational security of networked computer systems [16]. In fact the security of network infrastructure and data in not only the cyber space but also the physical space are now being game-theoretically analyzed [17]. 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 [15]. 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 Lye Kong-wei and Jeanette MW [14] viewed the interactions between an attacker and the administrator as a two-player stochastic game and constructed a general-sum game model for the problem. The resultant nonlinear programs were solved by finding all Nash equilibria of the games. However, their game model seems to be more suitable in host-based environments and specific scenarios such as virus and operating system interactions. Our work addresses similar situations in network environments only.

Sallhammar et al. [1,12,16] used a dynamic approach to study systems, the types of vulnerabilities they are opened to, the way systems are operated and how they behave in a certain environment. Their work describe also describe how stochastic assumptions are used to study systems yet to be built or existing systems whose vulnerabilities are unknown. They demonstrated how the stochastic modeling approach can be used for real-time risk assessment of a system, and suggested how the system's security and dependability behavior can be predicted in real-time. However, their work appears largely bordering on the vulnerability assessments and dependability evaluation of network systems.

The recent trend in the use of game-theory in security is in addressing the issue of efficient allocation of security resources in large scale network. Rong Yang et al. [18] proposed game-theoretic human behavioral models for computing defender optimal allocation strategies against rationally bounded human adversaries. Also Vanek et al. [19] studied the resource allocation problem of selecting and inspecting potentially vulnerable packets in large networks. They further provided novel algorithms and models for addressing these issues.

The work of Lye Kong-wei and Jeanette, Alpcan and Baser, Sallhammar et al. [3,12,14] and many others view stochastic security games as general-sum, non-linear programming problems that could be solved using dynamic programming algorithms, the value iteration algorithm or any other similar approaches. In this paper we investigate how attack scenarios can be analyzed 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.

The Stochastic Game Model

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

$$G = (S, P, (A_i, \propto_i, U_i)_{1 \le i \le |P|}, Q) \qquad (1)$$

Table 1: Definition of parameters.

Parameter	Expression
States S	$S = \{s_1, s_1, s_1, .., s_t\}_{1 \le t \le \lvert S \rvert}$
Players P	$P = \{p_k\}_{k=1,2}$
Actions A_i	$\forall (p_k \in P) \exists A_i = \{a_1, a_2 \ldots .a_n\}$
State Actions α_i	$\alpha_i : S \to A_i \, , \; i = 1,2+$
Player Profile $S\alpha$	$S\alpha = \{(s,a) : s \in S, a = (a_i), a_i \in \alpha_i(s); 1 \le i \le \lvert P \rvert\}$
Payoff U_i	$U_i : S\alpha \to R \, , \; i = 1,2$
Probability Distribution Q	$Q : SA \to P(S)$

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.

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 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.

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 is valued according to the amount of time required to restore such to an operative state by the defender as captured in Table 2. The MTTR as shown in the table were gotten based advice of system administrators.

For our two-person finite zero-sum game, we considered the magnitude of the attacker's reward to equal to the loss of the defender. That is, if it takes asset Z and average X minutes to be restored after a breakdown, then the cost of asset C is X minutes. So it suffices that the defender's loss and attacker's gain are –X and +X minutes respectively. Thus satisfying the zero-sum property: $\mathrm{X} + (-\mathrm{X}) = 0$

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 Eq. (2) and Eq. (3) respectively

$$A_1 = p_1^a = \{a_1, a_2 n........a_n\} \tag{2}$$

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

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.

Table 2: Value and Priority Index of Network Components.

Asset	Symbol	Priority	Mean Time To Repair (mins)
Firewall	F	1	5
Web Server	W	2	35
Application Server	A	3	30
Database/File Server	D	4	20

For the two players in this game defender and attacker, their action set are captured below respectively;

$$\mathrm{p}_1^{\mathrm{a}} = \{\mathrm{restart}|\mathrm{patch}_{\mathrm{firewall}}, \mathrm{restart}|\mathrm{patch}_{\mathrm{webserver}},$$

$$\mathrm{restart}|\mathrm{patch}_{\mathrm{appserver}}, \mathrm{restart}|\mathrm{patch}_{\mathrm{dbserver}}, \mathrm{end}_{\mathrm{game}}\}$$

$$\mathrm{p}_2^{\mathrm{a}} = \{\mathrm{attack}_{\mathrm{firewall}}, \mathrm{attack}_{\mathrm{webserver}}, \mathrm{attack}_{\mathrm{appserver}},$$

$$\mathrm{attack}_{\mathrm{dbserver}}, \mathrm{end}_{\mathrm{game}}\}$$

Modeling and Encoding 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 operational, or completely out of operation. Lye Kong-wei and Jeanette MW [14] 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, the current state and the past attack records of the attacker. 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 $\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_3, \ldots\ldots, \mathrm{s}_\mathrm{t}\}_{1 \le \mathrm{t} \le |\mathrm{S}|}$ 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_1 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, $0 \le P^k \le 1$, $\sum_{k=1}^{|\mathrm{S}|} \mathrm{P}^\mathrm{k} \forall \mathrm{k}$. These probabilities are computed based on the experience of the system administrators considering the rationality of attackers and the nature of past attack propagation in the network.

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 $\mathrm{S}' \subset \mathrm{S}$ considered feasible for both the attacker and defender while the remaining states in set S are infeasible.

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 using a binary naming scheme, the total number of states could be easily computed using Eq. (4)

$$N = 2^K + 1 \qquad (4)$$

With respect to equation (4) and the hypothetical network shown in Figure 1 where we have a total of $K = 4$ network assets, the total number of states is computed as 17.

We generate the sequence of the bits in each state string according to a priority as shown in table 2 indicating 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.

Using Table 2, each state is encoded as a 4-digit length binary string of the format $S = FWAD$ where the character symbols (FWAD) are permutatively replaced by binary digits indicating ON (1) and OFF (0) e.g. 0010,1001 etc.

Since K=4 as shown in Figure 1, and using Eq. (4) the number of state is 17 computed as

$$N = 2^K + 1 = 2^4 + 1 = 16 + 1 = 17.$$

The set of all possible states (in no particular order) is given below;

$$S = \{1111, 0111, 0011, 0001, 1110, 1100, 1000, 1010, 0101, 1101, 1011, 0100, 0110, 1001, 0010, 0000, \varnothing\}$$

However, the set of States

$$S' = \{1111, 0111, 0011, 0001, 0101, 0100, 0110, 0010, 0000, \varnothing\}$$

where $S' \subset S$ are considered feasible.

State Transition Diagrams

Game transition diagrams are drawn to show the probabilistic movement of players from one state to another within an attack-defend scenario. The transitions are according to a probability distribution that takes into consideration knowledge of the network, the security index of its components and experience of past attack propagations. Figure 2 shows the possible game played by an attacker while Figure 3 would be a likely defense response by the defender. The circles depict game states labeled with a binary string while the arcs or edges show movement from state to state. Each transition arc is labeled by the action taken to move from a current state to the next state and the probability of making such a choice.

Generating Game Matrices

All state games are represented in strategic form as a two-dimensional matrix. The defender is designated the row player, while the attacker is designated as 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 Eq. (5).

$$G^{(k)} = (a^k_{i,j} + \sum_i^N P_i^{(l)} G^{(l)}) \quad \text{for } k = 1..N \qquad (5)$$

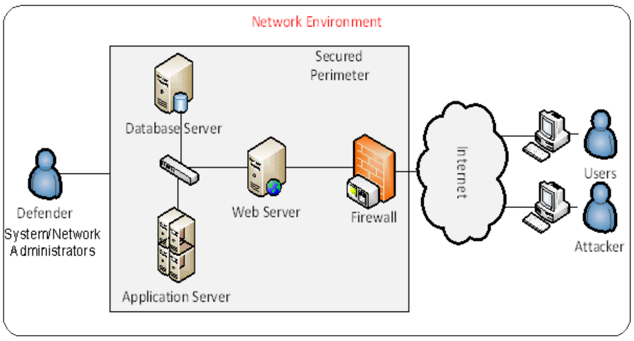

Figure 1: Typical Network Environment.

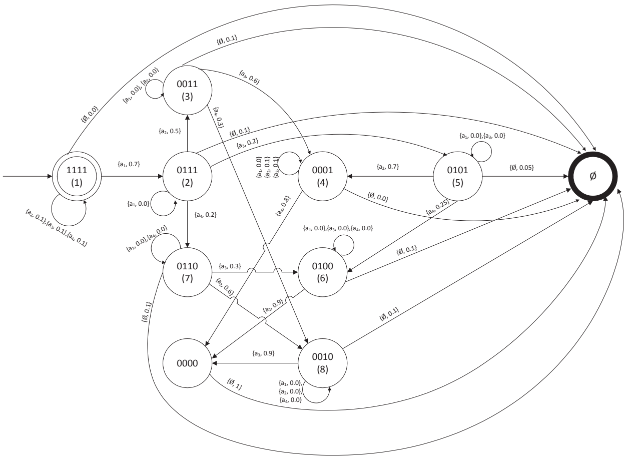

Figure 2: Attacker's Game.

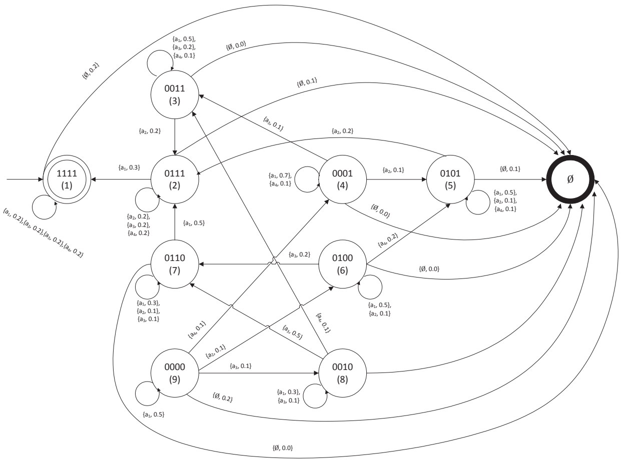

Figure 3: Defender's Game.

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 $P_i^{(l)}(l)$ [20]. 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 *l* from any state i.e. $P^{(l)} = \sum_i^N P_i^{(l)}$. A history of attack propagation H over previous games is recorded with information regarding the strategies taken by players at each game state. H could be useful for computing probability p^k given the number of states in H where taking action a_k transits the game to state s_k and the total number of states in H. This probability is then normalized by a factor of heuristic f which is a function of the analyst perception and knowledge of the network.

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 (Table 2). Also, let B be the asset that defender's action d_i 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 analyze 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 Eq.(8);

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

That is for the network in Figure 1 the base matrix is given as G and the defender's game is given as G′

$$G = \begin{pmatrix} 5,-5 & 40,-40 & 35,-35 & 25,-25 \\ 40,-40 & 35,-35 & 65,-65 & 55,-55 \\ 35,-35 & 65,-65 & 30,-30 & 50,-50 \\ 25,-25 & 55,-55 & 50,-50 & 20,-20 \end{pmatrix}$$

$$G = \begin{pmatrix} -5 & -40 & -35 & -25 \\ -40 & -35 & -65 & -55 \\ -35 & -65 & -30 & -50 \\ -25 & -55 & -50 & -20 \end{pmatrix}$$

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

Using the technique and equations described in this section, the state matrices for major game states are derived as shown in Table 3.

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 $V^{(k)}$ [20]. For all games matrices, the game values are the unique solutions of (5) with game values given as Eq. (9)

$$V(k) = Val(a_{i,j}^k + \sum_i^N P_i^{(l)} V(l)) \tag{9}$$

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. Ferguson ST [20] 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 I I does and there is a mixed strategy for

Table 3: Game Matrices.

STATE: 1111 $P^1 = 0.2 + 0.2 + 0.2 + 0.2 = 0.8$ $s^1 = 0.2$ $G = \begin{pmatrix} -5 + P^1G^1 & -40 & -35 & -25 \\ -40 & -35 + P^1G^1 & -65 & -55 \\ -35 & -65 & -30 + P^1G^1 & -50 \\ -25 & -55 & -50 & -20 + P^1G^1 \end{pmatrix}$
STATE: 0111 $P^1 = 0.3$ $P^2 = 0.2 + 0.2 + 0.2 = 0.6$ $s^2 = 0.1$ $G = \begin{pmatrix} -5 + P^1G^1 & -40 & -35 & -25 \\ -40 & -35 + P^2G^2 & -65 & -55 \\ -35 & -65 & -30 + P^2G^2 & -50 \\ -25 & -55 & -50 & -20 + P^2G^2 \end{pmatrix}$
….
STATE: 0000 $P^4 = 0.1P^6 = 0.1$ $P^8 = 0.1$ $P^9 = 0.5$ $s^8 = 0.2$ $G = \begin{pmatrix} -5 + P^9G^9 & -40 & -35 & -25 \\ -40 & -35 + P^6G^6 & -65 & -55 \\ -35 & -65 & -30 + P^8G^8 & -50 \\ -25 & -55 & -50 & -20 + P^4G^4 \end{pmatrix}$

doi:http://dx.doi.org/10.4172/2324-9307.1000103

Player I I such that I I'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 I I" [20].

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 Ferguson ST [20], the following LP ensures that the attacker's average gain is v;

Choose v and $p_1, ..., p_m$ to maximize v

Subject to the constraints

$$v \le \sum_{i=1}^{m} p_i a_{i1} \ldots\ldots\ldots v \le \sum_{i=1}^{m} p_i a_{in} \qquad (10)$$

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

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

Choose w and $p_1, ..., p_m$ to minimize v

Subject to the constraints

$$w \ge \sum_{j=1}^{n} p_j a_{1j} \ldots\ldots\ldots w \ge \sum_{j=1}^{n} p_j a_{mj} \qquad (11)$$

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

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;

$$X^* = \{p = (p_1, \ldots, p_m) : 0 \le p_i, p_i \le 1 \forall i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} p_i = 1\} \quad (12)$$

$$Y^* = \{q = (q_1, \ldots, q_n) : 0 \le q_i, q_i \le 1 \forall i = 1, \ldots, n \text{ and } \sum_{i=1}^{n} q_i = 1\} \quad (13)$$

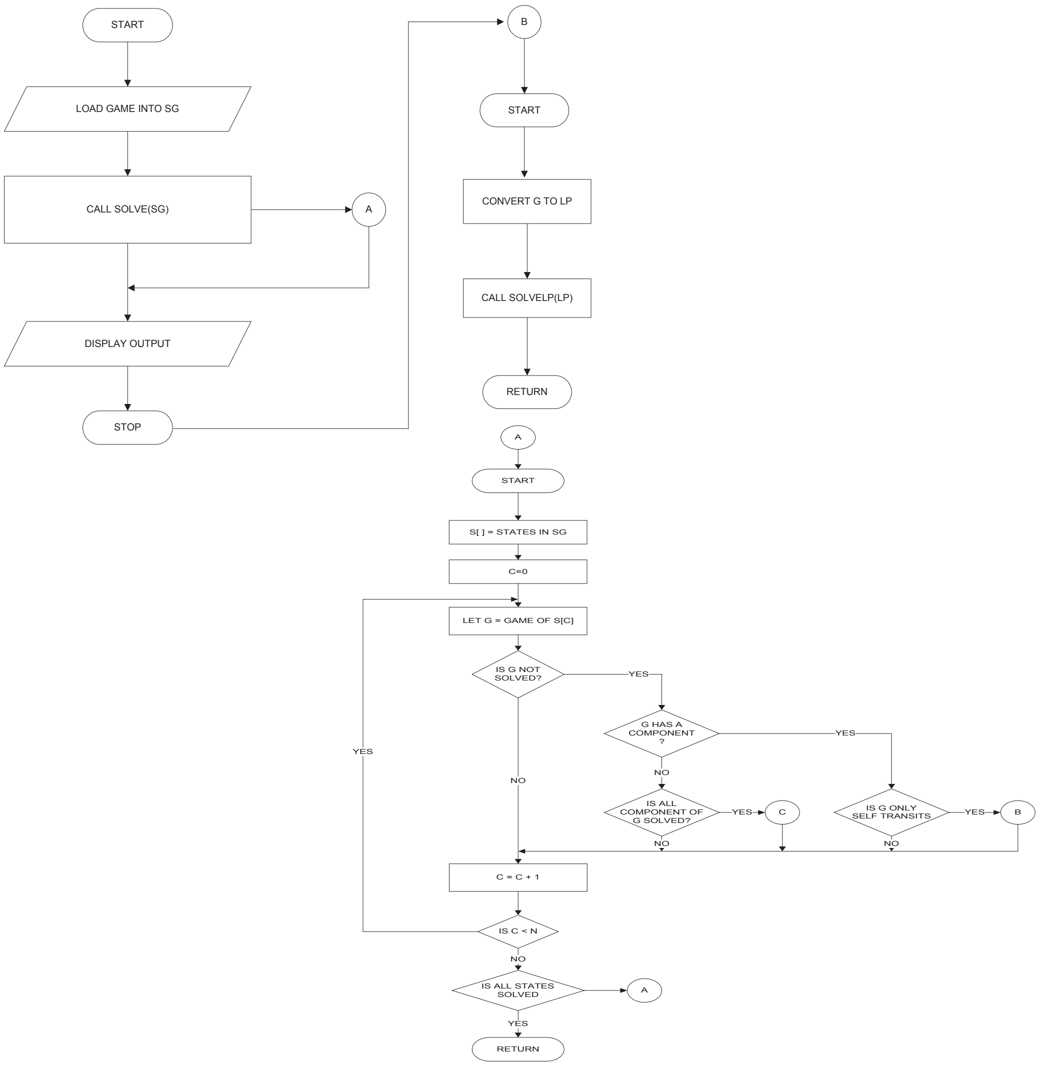

Figure 4: System Flowcharts.

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

V=(v(0),v(1),……,v(N)) where N is the number of states.

System Flowchart

Figure 4 shows the procedure for loading and solving games using the stochastic game model developed.

Simulation

The simulation environment is made of up a standalone PC with configuration of 320GB HDD, 4GB RAM, Duo Core running Windows 7 Operating System. The simulation software was developed using the C# programming language running on the Microsoft .NET Framework 4.0. It accepts game inputs and generates game outputs in Extensible Markup Language (XML) format. XML is chosen over CSV file format for its support for structured and domain-specific data. Using plain old text format requires extra level of complex programming logic to transverse the huge data required to load and initialize the game.

Result and Discussion

Table 4 and Table 5 show computed optimal strategies for both players and corresponding bar charts showing the suggested actions for each game state. 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.

Table 4: Computed Optimal Strategies for Defender.

ACTION				
STATE	a_1	a_2	a_3	a_4
1111	0.8586	0.144	0	0
0111	1	0	0	0
0011	0.8392	0.1608	0	0
0001	0.859	0.141	0	0
0101	0.859	0.141	0	0
0100	0.859	0.141	0	0
0110	0.859	0.141	0	0
0010	1	0	0	0
0000	0.8392	0.1608	0	0

Table 5: Computed Optimal Strategies for Attacker.

ACTION				
STATE	a_1	a_2	a_3	a_4
1111	0	0.8564	0.1436	0
0111	0	0.8586	0.144	0
0011	0	1	0	0
0001	0	0.9673	0.0352	0
0101	0	0.8564	0.1436	0
0100	0	0.8564	0.1436	0
0110	0	0.8564	0.1436	0
0010	0	1	0	0
0000	0	0.9673	0.0352	0

Table 6: Computed Game Values.

S/N	State	State Game Value
1	1111	-39.359
2	0111	-39.2931
3	0011	-40.0623
4	0001	-39.8744
5	0101	-39.359
6	0100	-39.359
7	0110	-39.359
8	0010	-40.0623
9	0000	-39.8744

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 the next highest optimal strategy. The vector of game values V shown in Table 6, 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. The pattern shows an all negative game values meaning that the game favours the attacker than the defender. That is the defender looses more to the attacker and as such must make his moves in order to minimize the expected average payoff. 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.

Conclusions

Stochastic modeling of computer networks allows researchers to be able to model and analyse the both defender's and attacker's behaviour with the 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.

The reactive nature of existing security schemes is not suitable for the changing landscape of network security, a proactive technique that cannot only determine the set of vulnerable network assets and

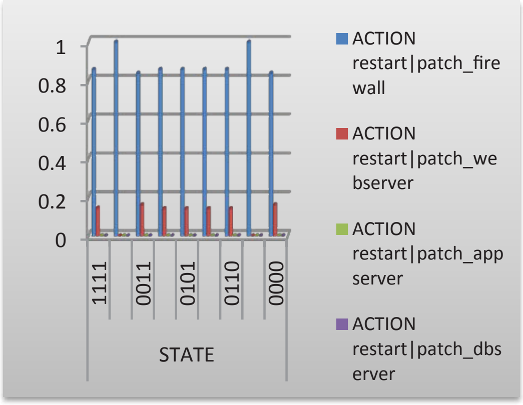

Figure 5: Bar Chart showing suggested actions per state (Defender).

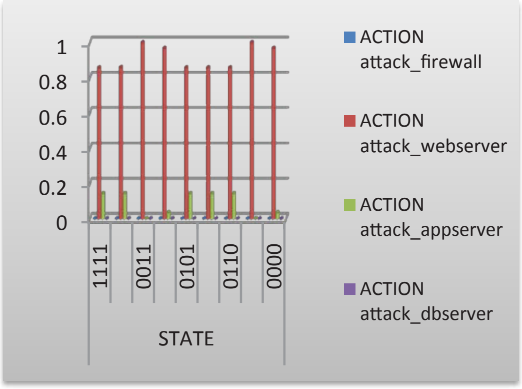

Figure 6: Bar Chart showing suggested actions per state (Attacker).

their vulnerabilities, but also predict the pattern of behaviour of the players. Also, it is very difficult to obtain comprehensive and robust models solely on the use of state-based stochastic techniques. So in future works we shall look into combining attack graph, stochastic petri nets as described in Wang Y et al. [21] and stochastic games, where attack graph is used to study how attacks are propagated and serve as a basis for risk computations while stochastic game net is used for analysis of attacks, predicting attackers behaviour and recommending appropriate counter-measures.

References

1. Sallhammarb K (2007) Stochastic Models for Combined Security and Dependability Evaluation. Ph.D. Thesis, Department of Telematics, FITME, Norwegian of Science and Technology. Trondheim, Norway.
2. 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.
3. Alpcan T, Baser T (2010) Network Security: A Decision and Game-Theoretic Approach. (1st edn). Cambridge University Press, UK.
4. Adetunmbi AO, Alese BK, Ogundele OS, Falaki SO (2007) A Data Mining Approach to Network Intrusion Detection. Journal of Computer Science & its Applications 14: 24-37.
5. Khanna R, Liu H (2007) Distributed and Control Theoretic Approach to Intrusion Detection. International Conference on Wireless Communications and Mobile Computing, ser. IWCMC '07. New York, NY, USA.
6. Adetunmbi AO, Falaki SO, Adewale OS, Alese BK (2008) Network Intrusion Detection based on rough Set and k- Nearest Neighbour. International Journal of Computing and ICT Research 2: 60-66.
7. Cavusoglu H, Raghunathan S, Yue W (2008) Decision-Theoretic and Game-Theoretic Approaches to IT Security Investment. J Manage Inform Syst 25: 281.
8. Assane Gueye (2011) A Game Theoretical Approach to Communication Security. Electrical Engineering and Computer Sciences Department, University of California at Berkeley. Technical Report No. UCB/EECS-2011-19.
9. Schneier B (1999) Attack trees: Modeling security threats. Dr. Dobb's Journal, December.
10. Ryan T (2009) Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century. Addison-Wesley Professional, USA.
11. Steffan J, Schumacher M (2002) Collaborative Attack Modeling. Symposium on Applied Computing, Madrid, Spain.
12. Sallhammar K, Knapskog SJ, Helvik BE (2005) Using Stochastic Game Theory to Compute the Expected Behavior of Attackers. International Symposium on Applications and the Internet (Saint 2005). Trento, Italy.
13. Shapley LS (1953) Stochastic Games. Proceedings of the National Academy of Science USA 39: 1095-1100.
14. Lye Kong-wei, Jeanette MW (2005) Game Strategies In Network Security: Extended Abstract for FCS. Int J Infor Secur 4: 71- 86
15. Roy S, Ellis C, Shiva S, Dasgupta D, Shandilya V, et al. (2010) A Survey of Game Theory as Applied to Network Security. System Sciences (HICSS)-43rd Hawaii International Conference, Hawaii, USA.
16. Sallhammar K, Knapskog SJ (2004) Using Game Theory in Stochastic Models for Quantifying Security. 9th Nordic Workshop on Secure IT-systems, Espoo, Finland.
17. Zhuang Jun, Rao Nageswara S V, He Fei (2012) Game-Theoretic Analysis of Attack and Defense in Cyber-Physical Network Infrastructures. Industrial and Systems Engineering Research Conference, Orlando, USA.
18. Rong Yang, Fei Fang, Albert Xin Jiang, Karthik Rajagopal, Milind Tambe, et al. (2012) Designing Better Strategies against Human Adversaries in Network Security Games. 11th International Conference on Autonomous Agents and Multiagent Systems- Innovative Applications Track, Valencia, Spain.
19. Vanek O, Yin Z, Jain M, Bošanský B, Tambe M, et al. (2012) Game-theoretic Resource Allocation for Malicious Packet Detection in Computer Networks. 11th International Conference on Autonomous Agents and Multiagent Systems. Valencia, Spain.
20. Ferguson ST (2007) Game Theory II – Two-Person Zero-Sum Games.
21. Wang Y, Li J, Meng K, Lin C, Cheng X (2013) Modeling and Security Analysis of Enterprise Network using Attack–Defense Stochastic Game Petri Nets. Security Comm Networks 2013, Wiley Online Library 6:89-99.

Author Affiliation

Top

[1]*Department of Computer Science, Federal University of Technology, Akure, Nigeria*