DSGM-Based Analysis of Computer Networks Security

Boniface Kayode Alese, Iwasokun Gabriel Babatunde, Haruna Danjuma Israel
Department of Computer Science, Federal University of Technology
Akure, Nigeria

Abstract - The computer game framework is an extension of attack graphs based on game theory. It is used for the anticipation and analysis of intruder and administrator concurrent interactions within the network. Like attack-graph-based model checking, the goal of an anticipation game is to prove that a safety property holds. However, using this kind of game is tedious and error prone on large networks since it is based on the assumption that the analyst has prior and complete knowledge of critical network services. This work presents a non-zero, Deterministic Stochastic Game-theoretic Modeling (DSGM) method for analyzing the security of computer networks. The interaction between an attacker and a defender is viewed as two-player non-zero deterministic game whose model is constructed using a saddle point solution (non-linear program) to compute the value of the game. The probability of possible attack on a network given available attacker strategy or best-response strategies for the attacker and the defender was demonstrated. The simulation of the model was achieved using Java, which is a high level language with object oriented programming capabilities and features. The backend is implemented using MySQL, which is a widely used Relational Database Management System (RDBMS). Selected results of the games played were presented with analysis showing some likely attacker activities and the counter measures from the defender.

Keywords:- Game theory; modeling; network attack; defence strategy; saddle point

I. INTRODUCTION

The continuous evolution of computer networks and mobile applications has drastically changed the nature of their security and privacy. Computer networks play an increasingly important role in modern society and also resulted in the emergence of new types of security and privacy problems that involve direct participation of network agents. Network agents are individuals, as well as devices or software with self motivation [1]. Security on the Internet and Local Area Networks (LAN) is now at the forefront of computer related issues. Throughout the evolution of networking and the internet, the threats to information and networks have increased tremendously with cleverly exercised attacks that cause damages or committal of theft [2]. Network security is the most vital component in information security because it is responsible for securing all information passed through networked computers. It refers to all hardware and software functions, characteristics, features, operational procedures, accountability measures, access controls, and administrative and management policy required to provide an acceptable level of protection for hardware, software, and information in a network [3]. For a successful prevention of information loss, network security must follow three fundamental precepts [4]. First, a secure network must have integrity such that all of the information stored therein is always correct and protected against fortuitous data corruption as well as willful alterations. Next, there must be confidentiality, or the ability to share information on the network with only those people for whom the viewing is intended. Finally, network security requires availability of information to its necessary recipients at the predetermined times without exception.

Network security has become more important to personal computer users, organizations, and the military. With the advent of the internet, security became a major concern. The internet structure itself allowed the occurrence of several security threats. Knowing the attack methods allows for the appropriate counter (security) measure. Many businesses secure themselves against Internet threats using firewalls and encryption mechanisms. Some create Intranet to establish secured connection to the Internet [5]. Internet ubiquity has also made computer security more important than ever to several organizations including governments, banks, businesses and universities. It is now important for security specialists to know what an intruder can do to a computer network as well as counter measures [6].

Network security has social dimension [7] and is not to be seen as a problem with one time or final solution via better protocols and algorithms or any other engineering solution.

When viewed from a game theoretic perspective, it can be seen as a game comprising multiple players notably the attackers and the defenders. The interactive behavior between an attacker and a defender is similar to information warfare with both the attacker and defender having several available strategies.

Network security attacks target the following key elements of the network security architecture [8]:
A. Authentication Attacks (Unauthorized access)

These types of attacks occur when a user manipulates system resources or gains access to system information without authorization by either sharing logins or passwords or by using an unattended terminal with an open session. They are performed using a program that runs across a network and attempts to log into a shared resource, such as a server [9].

B. Confidentiality Attacks (Network Snooping/Sniffing)

With the use of dozens of freeware and shareware packet sniffers available, which do not require the user to understand anything about the underlying protocols, the attackers would capture all network packets and thereby the users login names, passwords, and even accounts. The intruders usually take advantage of human tendency such as using a single, same password for multiple accounts.

C. Integrity Attacks (Message Alteration, Delay, and Denial)

In this type of attack, data or information is added, removed, or modified in transit across the network. This requires root access to the system or a router. If a program does not check buffer limits when reading or receiving data, an attacker can add into a program or system, an arbitrary data that gives the intruder root access. Integrity attacks can create a delay, causing data to be held or otherwise made unavailable for a period of time.

D. Access Control Attacks (Address Masquerading)

An attacker “listens” to the network traffic, finds the Internet Protocol (IP) address of a trusted host or system, configures his/her own network interface, and transmits the message as if from the trusted host. This is called IP address masquerading or IP spoofing.

Common internet attacks methods are broken down into categories. Some attacks gain system knowledge or personal information, such as eavesdropping and phishing. Attacks can also interfere with the system’s intended function, such as viruses, worms and trojans. The other form of attack is when the system’s resources are consumes uselessly, these can be caused by denial of service (DoS) attack. Other forms of network intrusions are land, smurf and teardrop attacks. They are not as well known as DoS attacks, but are used in various forms [5].The common attack methods and the security technologies are shown in Table I[10].

The following are the different detection and defense mechanisms for dealing with these attacks [5, 10]:

- **Cryptographic systems**, which is a useful and widely used tool in security engineering and involve the use of codes and ciphers to transform information into unintelligible data.
- **Firewall**, which is a typical border control mechanism with the aim of blocking unauthorized internal and external traffics. It serves as a front line defense mechanism against intruders through unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both.
- **Intrusion Detection Systems (IDS)**, which is an additional software or hardware based protection measure that helps ward off computer intrusions. Its products are used to monitor connection in determining whether attacks are been launched. Some IDS systems just monitor and give alert of an attack, whereas others try to block the attack.
- **Anti-Malware Software and scanners**, used to detect and cure system infected with virus and malwares.
- **Secure Socket Layer (SSL)**, which is a suite of standard way protocols for achieving a good level security between a web browser and a website through a secured channel or tunnel. It provides authentication of clients to server and certificates for proving identity.

II. GAME THEORY

Game theory, initially developed by Von Neumann [11], is an abstract mathematical theory for analyzing interactions among multiple intelligent actors which may be people, corporations, nations, intelligent software agents, or robots [12]. Game theory provides mathematical framework for the analysis and modeling of network security problems and supports finding out the optimal strategy for all party. It also describes multi-person decision scenarios as games where each player chooses actions, which result in the best possible rewards, while anticipating the rational actions from other players [13].

<table>
<thead>
<tr>
<th>Computer Security Attributes</th>
<th>Attack Methods</th>
<th>Technology for Internet Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality</td>
<td>Eavesdropping, Hacking, Phishing, DoS and IP Spoofing</td>
<td>IDS, Firewall, Cryptographic System, IPSec and SSL</td>
</tr>
<tr>
<td>Integrity</td>
<td>Viruses, Worms, Trojans, Eavesdropping, DoS, and IP Spoofing</td>
<td>IDS, Firewall, Anti-Malware Software, IPSec and SSL</td>
</tr>
<tr>
<td>Privacy</td>
<td>Email bombing, Spamming, Hacking, DoS and Cookies</td>
<td>IDS, Firewall, Anti-Malware Software, IPSec and SSL</td>
</tr>
<tr>
<td>Availability</td>
<td>DoS, Email bombing, Spaming Systems Boot Records Infectors</td>
<td>IDS, Anti-Malware, Software and Firewall</td>
</tr>
</tbody>
</table>
In providing mathematical basis for understanding intelligent actors’ interactions with each other, game-theoretic approaches assume that these intelligent actors will anticipate each other’s moves, and act appropriately [7, 14]. Game theory is a branch of mathematics where strategic situations are studied and players choose different actions in an attempt to maximize their returns [13]. Each player has a number of strategies (feasible actions), which determine the outcome of the game and the pay-off to each player. The conflict arises when particular players value their outcomes differently. A pure conflict in a ‘zero-sum game’ situation arises when two players aim to achieve opposite outcomes. An equilibrium outcome is achieved when each player has chosen a strategy, either pure or mixed, and neither has any incentive to move to a different strategy. This happens only when a maximin strategy of one player gives the same outcome as a minimax strategy of another player [15].

Game Theory has attracted the attention of security researchers aiming to develop quantitative analysis techniques for Network and Information Security. For them, it provides a natural vehicle for a rigorous formulation of attacks, threat analysis, and reactive decision-making [16-17]. The success of a security scheme depends not only on the actual defense strategies that have been implemented, but also on the strategic actions taken by the attackers to launch their attacks. It also depends on the actions of the users sharing the systems, and on the actions of their peers situated in other networks. All these agents act rationally according to their various incentives. Game Theory provides means to represent these complex, competitive, and multi-agent interactions into mathematical models that allow a rigorous analysis of the problem [13, 17]. It also helps the agents predict each other's behavior and suggests a course of action to be taken in any given situation. Game theory is in two major forms; namely extensive and normal forms. The extensive form formalizes games with a time sequencing of moves. Games are played on trees as shown in Figure 1 with each vertex (or node) representing a point of choice for a player. A number listed by the vertex specifies a player, lines from the vertex represent possible actions for the player and payoffs are specified at the bottom of the tree [18]. The extensive form is also called a game tree with a complete description of how game is played over time. The description includes the order in which players take actions, the information available to players at the time they must take actions and the times at which any uncertainty in the situation is resolved. A game in extensive form can be analyzed directly or converted into equivalent strategy [19]. In Figure 1, Player 1 takes the first move and chooses either F or U. Player 2 sees Player 1’s move and then chooses A or R.

If Player 1 chooses U and then Player 2 chooses A, then Player 1 gets 8 and Player 2 gets 2.

The extensive form also captures simultaneous-move and imperfect information games which are represented either by a dotted line connecting different vertices as being part of the same information set (that is, players do not know at which point they are), or a closed line [19]. The normal (or strategic) form is represented by a matrix showing the players, strategies, and payoffs as shown in Table 2. A Player is designated as the “row” player while the other is the “column” player. Each rows or column represents a strategy and each box represents the payoffs to each player for every combination of strategies. Games on the concept of this form are solved using the Linear Programming and Nash equilibrium techniques [19-20]. As shown in Table 2, there are two players; one chooses the row and the other chooses the column. Each player has two strategies specified by the number of rows and columns. Payoffs are provided in the interior and the first number is the payoff received by the row player (Player 1) while the second is the payoff for the column player (Player 2). Given that Player 1 plays Up and that Player 2 plays Left, then Player 1 gets a payoff of 4, and Player 2 gets 3 [18].

![Figure 1: Extensive Game Representation](image)

<table>
<thead>
<tr>
<th>Player 1, Chooses</th>
<th>Player 2, Chooses Left</th>
<th>Player 2, Chooses Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>4, 3</td>
<td>-1, -1</td>
</tr>
<tr>
<td>Down</td>
<td>0, 0</td>
<td>3, 4</td>
</tr>
</tbody>
</table>

When a game is presented in normal form, it is presumed that each player acts simultaneously or, at least, without knowing the actions of the other. If players have some information about the choices of other players, the game is usually presented in extensive form. Every extensive-form game has an equivalent normal-form game, however the transformation to normal form may result in an exponential blowup in the size of the representation, making it computationally impractical [21].

III. RELATED WORKS

The use of game theory in the modeling of the interaction between users within a system has appeared in several areas of information security research [22]. Many mathematical models have been used to model and analyze the decision-making problems in security. Machine Learning [23], Control Theory [24], and Pattern Recognition [25], are other mathematical models that have been utilized to solve the security decision problem. More recently, Game Theory has been considered to study network security problems. Among all these methods,
Game Theory seems very appealing because, in addition to providing a principled way to understand security problems, game theoretic models capture the adversarial nature of the security problem. Instead of designing a defense against a specific attack, Game Theory attempts to design a defense against a sophisticated attacker who plans in anticipation of a complex defense. As of such, both the defender and attacker’s actions can be in principle computed and analyzed. Furthermore, Game Theory can model issues of trust, incentives, and externalities that arise in security systems.

Game theoretic models had been applied to network security issues [1, 13], and some of these approaches look promising. Liu et al. in [26] employed the concept of incentives to express attackers’ intentions, while the concept of utilities was used to integrate incentives and costs in such a way that the system as well as attackers’ objectives can be practically modeled. They introduce a conceptual model for determining attacker intent, objectives and strategies rather than using a specific type of game for modeling attacks. Furthermore, they introduce conditions under which a specific type of game model will be feasible and desirable. Alpcan and Basar in [7] investigated a security game as a two player, non-cooperative, non-zero-sum game. The game is assumed to be a complete information game and the player’s optimal strategy depends only on the payoff function of the opponent. Lye et al. in [6] presented how a network security problem can be modeled as a general-sum stochastic game between attacker and the administrator and showed the computation of Nash equilibria. Their approach is specific to network security applications and assume the benefit of attackers arises from harming the network, hence only dealing with malicious users. Liu et al. in [27] introduced some of the problems in performing tradeoff analysis in network security. Static and dynamic Bayesian games where formulated to demonstrate the suitability of game theory for the development of various control algorithms in intrusion detection. However, the work only dealt with potentially malicious users, who may only have a positive payoff through attacking the system. Browne in [28] gave a description of the use of static games to analyze attacks scenarios involving complicated and heterogeneous military networks. A defense team has to defend a network of three hosts against an attacking team’s worms. Member chooses whether or not to run a worm detector and depending on the combined attack and defense actions, each outcome has different costs. The interactions between the two teams, however, are dynamic and can be better represented using a stochastic model instead of deterministic.

Bell in [29] considered a zero-sum game in which the router has to find a least cost path and a network tester seeks to maximize this cost by failing a link. Two players are in some form of control over the network and they have opposite objectives. Finding the least-cost path in their problem is analogous to finding a best defense strategy. Hespanha & Bohacek in [30] discussed routing games in which an adversary tries to intersect data packets in a computer network. The designer of the network has to find routing policies that avoid links that are under the attacker’s surveillance. Finding optimal routing policy is similar to finding the least-cost path in [29] and the best defense strategy is that at every state, each player has to make a decision on what action to take. A simple one-player game for FRIARS cyber-defense decision system capable of reacting autonomously to automated system attacks is presented in [31]. The system predicts the opponent’s next move one at a time. The formulated model is closer to Markov decision problem because it is a single-player game.

IV. PROPOSED DETERMINISTIC SECURITY GAME MODEL

The proposed deterministic security game system’s model aimed at constructing a typical network shown in Figure 2 and its risk assessment framework. An attack-defense game model, which quantifies the probability of threats, is formulated for the risk assessment. Since the proposed game model is centred on an attacker and a defender interaction in a computer network environment and the intention of each player is to maximise his or her wins or minimise his or her loss as much as possible, a strategic tuple $G = (n, A_i, R_i)$ is therefore defined as:

$$G = (n, A_i, R_i)$$

where $n$ is the number of players, $i = 1, 2, \ldots, j A_i$ is the set of actions or strategy space available to a player and $R_i$ is the player’s payoff function $A_i \rightarrow R_i$.

![Figure 2: The system network environment](image)

Each player selects actions from available action space with the goal of maximizing the payoff, which depends on the other player’s actions. These are often called matrix game because the $R_i$ can be written as n-dimensional matrices.

In the proposed model, the game played is a deterministic, non-zero sum payoff matrix game with a win for one not necessarily implying a loss for the opponent. The matrix game played for the attacker and the defender is made up of rows and columns.
The number of rows in the matrix represents the available strategy space for player 1 and the number of corresponding column represents the strategy space for player 2 as shown in Table 3.

**TABLE 3: PAYOFF MATRIX**

<table>
<thead>
<tr>
<th>Player 2</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Player 1</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

The elements of the matrix represent gain or loss on computing resources or assets which correspond to a strategy in the strategy space played by a player. The model creates a network environment for the interaction between the two players to play the matrix game \( G \), defined as:

\[
G_{k,j} = \begin{cases} 
\text{true}, & f(x_{a,b}) > 0 \\
\text{false}, & f(x_{a,b}) < 0 
\end{cases} \quad (2)
\]

\( G_{k,j} \) is the matrix with row and column of size \( k \) and \( j \) respectively. \( k \) is the number of strategy for player 1 and \( j \) represents the number of strategy for player 2. \( a \in k \) and \( b \in j \), \( f(x_{a,b}) \) is a function that returns an integer number for an element in a position in \( G \) given that:

\[
-9 \leq f(x_{a,b}) \leq 9 \quad (3)
\]

For each deterministic game \( G \) played by the players, the attacker selects an optimal strategy from row \( k \) of the matrix to maximise his gain. This will compel the defender to select another optimal strategy from column \( j \) to minimise his loss. A resultant positive value from the game is win to the defender and a loss to the attacker. Consequently, a negative value represents a loss to the defender and a win to the attacker.

A mixed strategy procedure is used to play the game with each player choosing the strategy using a discrete probability distribution. For player 1, the probabilities are \( (x_1, x_2, \ldots, x_m) \), \( x_i \) is the probability that strategy \( i \) will be chosen. For player 2, the probabilities are \( (y_1, y_2, \ldots, y_n) \), \( y_j \) is the probability that strategy \( j \) will be chosen. Both sets of probabilities must sum to 1 and the expected value of the payoff of the game when the players use a mixed strategy is given by [32]:

\[
\text{Expected payoff} = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_i y_j \quad (4)
\]

\[
\sum_{i=1}^{m} x_i = 1, \sum_{j=1}^{n} y_j = 1 \quad (5)
\]

If both players follow their minimax strategies, the value of the game, \( v \), is equal to the expected payoff. To describe the logic of the mixed strategy, consider the problem of player 1 when selecting the probabilities \( (x_1, x_2, \ldots, x_m) \). The player knows that the expected payoff of the game to player 1 when player 2 use pure strategy \( j \) is:

\[
\sum_{i=1}^{m} p_{ij} x_i \text{ for } j = 1, \ldots, n \quad (6)
\]

Player 2 will from wisdom, selects a strategy that will ensure that the value of the game for player 1, \( v_l \), is less than or equal to the expected value for any pure strategy \( j \). Then for each \( j \), the constraint is written as:

\[
\sum_{i=1}^{m} p_{ij} x_i \geq v_l \text{ for } j = 1, \ldots, n \quad (7)
\]

Player 1 will then select values for the probabilities that maximizes return leading to the linear programming solution for player I's strategy as:

\[
\text{Maximize } v_l \\
\text{subject to } \sum_{i=1}^{m} p_{ij} x_i \geq v_l \text{ for } j = 1, \ldots, n \quad (8)
\]

\[
\sum_{i=1}^{m} x_i = 1 \\
x_i \geq 0, i = 1, \ldots, m
\]

The last two constraint sets assure that the decision variables \( x_i \) define an acceptable probability distribution. It is not hard to show that the dual of this model solves the selection problem for player 2. Assigning the dual variables \( y_j \) to the constraints in the set, we obtain the dual problem [32]:

\[
\text{Minimize } v_l \\
\Sigma_{j=1}^{n} y_i = 1 \\
\text{subject to } \sum_{j=1}^{n} p_{ij} y_j \leq v_l \text{ for } i = 1, \ldots, n \\
y_j \geq 0, j = 1, \ldots, n \quad (9)
\]

\( v_l \) is the cumulative game value or gain of attacker, \( x_i \) is the probability that the attacker takes \( (i^\text{th}) \) action in his action set. \( P_{ij} \) is the element at location \( (i,j) \) in the game matrix and \( m \) is the number of actions in the attacker’s action set. \( v_l \) is the cumulative game value of the defender, \( y_j \) is the probability that the attacker takes \( (j^\text{th}) \) action in his action set. \( P_{ij} \) is the element at location \( (i,j) \) in the game matrix and \( n \) is the number of actions in the defender’s action set.

Solving either one of these linear programs yields the optimal solution for one player while dual values provide the solution for the other player. The expected output of the game is the probability of the optimal of the mixed strategies. For each game \( G \) with game size (dimension) \( i \), played by the players for \( z \)
number of times, the probability of the attacker $P_i^a$, achieving a gain (and the defender having a loss), is defined as:

$$P_i^a = \frac{\sum_{n=1}^{\infty} i = i + 1}{Z} \quad \forall f(x,a) < 0 \quad (10)$$

$$P_i^d = \frac{\sum_{n=1}^{\infty} i = i + 1}{Z} \quad \forall f(x,a) \geq 0 \quad (11)$$

and $P_i^a + P_i^d = 1 \quad (12)$

$P_i^a$ the probability of defender achieving a gain (and the attacker having a loss). Higher value of $P_i^a$ implies a win for the attacker meaning that the tendency for the attacker to wreak havoc to the network resources is high. On the contrary, a higher value of $P_i^d$, implies a win to the defender which means the network is secure. Tables 4 and 5 show the possible scenarios for attacker and defender respectively based on the probability values.

### TABLE 4: ATTACKER PROBABILITY DESCRIPTION

<table>
<thead>
<tr>
<th>LABEL</th>
<th>RANGE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.8 &lt; $P_i^a$ ≤ 1.0</td>
<td>Complete destruction of network resources and computer assets(server)</td>
</tr>
<tr>
<td>A2</td>
<td>0.6 &lt; $P_i^a$ ≤ 0.8</td>
<td>Denial of service(worm and virus attack)</td>
</tr>
<tr>
<td>A3</td>
<td>0.4 &lt; $P_i^a$ ≤ 0.6</td>
<td>Scanning, acquiring and controlling more network resources and assets</td>
</tr>
<tr>
<td>A4</td>
<td>0.2 &lt; $P_i^a$ ≤ 0.4</td>
<td>Unauthorised access and use of resources(security bridge)</td>
</tr>
<tr>
<td>A5</td>
<td>0.01 &lt; $P_i^a$ ≤ 0.2</td>
<td>Probing, attack attempt, scanning</td>
</tr>
</tbody>
</table>

### TABLE 5: DEFENDER PROBABILITY DESCRIPTION

<table>
<thead>
<tr>
<th>LABEL</th>
<th>RANGE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>0.8 &lt; $P_i^d$ ≤ 1.0</td>
<td>Monitor resources, probing client activities and network traffic</td>
</tr>
<tr>
<td>D2</td>
<td>0.6 &lt; $P_i^d$ ≤ 0.8</td>
<td>Disconnect all users, check system and object privileges and access level</td>
</tr>
<tr>
<td>D3</td>
<td>0.4 &lt; $P_i^d$ ≤ 0.6</td>
<td>Shutdown firewall, rebuild block and irrelevant ports</td>
</tr>
<tr>
<td>D4</td>
<td>0.2 &lt; $P_i^d$ ≤ 0.4</td>
<td>Shutdown network, repair network, include honey port, anti-worm, antivirus and antispyware</td>
</tr>
<tr>
<td>D5</td>
<td>0.01 &lt; $P_i^d$ ≤ 0.2</td>
<td>Computer resources lost, rebuild network and resources(servers). restore backup</td>
</tr>
</tbody>
</table>

### V. SYSTEM SIMULATION

The model, theoretical framework and user interface were implemented using Java, which is an object oriented high-level programming language. Java platform differs from most other platforms in that it runs on top of other hardware-based platforms and has two components; namely Virtual Machine (VM) and Application Programming Interface (API). VM compiles and translates Java scripts into codes, which are the machine instructions while API provides a large collection of ready-made software components including graphical user interface (GUI) widgets. The database for the system was implemented using MySQL Relational Database Management System (RDBMS), which is the world's leading open RDBMS [33] that runs as a server providing multi-user access to a number of databases. The simulation environment comprises of a wireless ad hoc network hosted for the Department of Computer Science, Federal University of Technology, Akure, Nigeria. The network is divided into two parts; namely the server and the client applications. The clients were in two parts with the first configured on Window 7 platform on a 2GB RAM, 250GB HDD, AMD Athlon X2 and 1.8GHz processor speed while the second was configured on Window Vista platform on 2GB RAM size, 350 HDD, Intel dual core T4200 and 2.0GHz processor speed. The Server System was on a Window Vista on a 2GB RAM size, 350 HDD, Intel dual core T4200 and 2.0GHz processor. The server application coordinates the client’s application that resides on the client machines and used by the players to interact and play the game. Figure 3 shows the client server configuration page that sets up the clients connected to the server.

All the connected clients were considered as normal users displayed under the “Connected clients” platform. The “Get Game” button is used to select a unique number to represent the game to be played for different matrices and the consequent activation of the “Set Up” button. Setting up a player as an attacker or defender is via the ‘Connected clients’ and ‘Client Description’ menus. The payoff matrix of the game played by the players is displayed on the page presented in Figure 4 alongside the optimal strategy, the condition of the game (either
stable or unstable), the selected strategy by both players and the outcome of the game.

Table 6 shows the players and the number of games played. The total game played is the sum of all possible dimensions (2x2, 2x3, 3x2, 3x3, 3x4, 4x2, 4x3, 4x4). Table 7 also presents the payoff game matrix showing the strategy played by players. The solution value of each game played is dependent on the status of the game. A stable game has a single value that is the saddle point. For a non-stable game, the result value is computed by converting the game into a linear programming problem and solving it, using pivot algorithm. Table 8 and 9 present the summary for the two players and the probabilities of their outcomes in a particular game strategy space (dimension), and conditional probability $C_B|C_A$ for games 4 and 10 respectively. The chart of Figure 5 and 6 also show the probability of an attacker wreaking havoc given a number of strategic spaces for games 4 and 10 respectively.

### Table 6: Game Players and Total Games Played

<table>
<thead>
<tr>
<th>ID</th>
<th>Player</th>
<th>Category</th>
<th>Game</th>
<th>Total Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>DEFENDER</td>
<td>1</td>
<td>106</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>ATTACKER</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>DEFENDER</td>
<td>4</td>
<td>99</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>ATTACKER</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>DEFENDER</td>
<td>7</td>
<td>101</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>ATTACKER</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>DEFENDER</td>
<td>10</td>
<td>104</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>ATTACKER</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>I</td>
<td>DEFENDER</td>
<td>14</td>
<td>104</td>
</tr>
<tr>
<td>10</td>
<td>J</td>
<td>ATTACKER</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>K</td>
<td>ATTACKER</td>
<td>17</td>
<td>96</td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>DEFENDER</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>M</td>
<td>ATTACKER</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>14</td>
<td>N</td>
<td>DEFENDER</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>O</td>
<td>ATTACKER</td>
<td>23</td>
<td>103</td>
</tr>
<tr>
<td>16</td>
<td>P</td>
<td>DEFENDER</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Q</td>
<td>ATTACKER</td>
<td>26</td>
<td>99</td>
</tr>
<tr>
<td>18</td>
<td>R</td>
<td>DEFENDER</td>
<td>26</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7: Payoff Game Matrix Showing the Strategy Played by Players (POM=PAY OFF MATRIX, OP=OPTIONAL, PL=PLAY, OC=OUTCOME)

<table>
<thead>
<tr>
<th>Game</th>
<th>Size</th>
<th>POM</th>
<th>Status</th>
<th>Attacker Strategy</th>
<th>Defender Strategy</th>
<th>OC</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>2x2</td>
<td>A</td>
<td>Stable</td>
<td>1 1 1 1</td>
<td>1 1</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>2x3</td>
<td>B</td>
<td>Stable</td>
<td>1 1 3 3</td>
<td>3 3</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2x4</td>
<td>C</td>
<td>Unstable</td>
<td>2 2 1 3</td>
<td>3 3</td>
<td>2.91</td>
</tr>
<tr>
<td>23</td>
<td>3x2</td>
<td>D</td>
<td>Stable</td>
<td>3 3 1 1</td>
<td>3 3</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>3x3</td>
<td>E</td>
<td>Unstable</td>
<td>3 3 3 3</td>
<td>3 3</td>
<td>2.80</td>
</tr>
<tr>
<td>23</td>
<td>3x4</td>
<td>F</td>
<td>Stable</td>
<td>3 3 3 0</td>
<td>3 3</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>4x2</td>
<td>G</td>
<td>Stable</td>
<td>4 4 1 1</td>
<td>4 4</td>
<td>4</td>
</tr>
<tr>
<td>26</td>
<td>4x3</td>
<td>H</td>
<td>Unstable</td>
<td>4 4 3 4</td>
<td>4 4</td>
<td>1.51</td>
</tr>
<tr>
<td>26</td>
<td>4x4</td>
<td>I</td>
<td>Stable</td>
<td>3 3 4 3</td>
<td>3 3</td>
<td>3</td>
</tr>
</tbody>
</table>

$$A = \begin{pmatrix} 8 \\ 7 \\ -6 \end{pmatrix}, B = \begin{pmatrix} 6 & 4 & 3 \\ 9 & -4 & -4 \end{pmatrix}, C = \begin{pmatrix} 4 & 0 \\ 6 & 1 \\ -2 \end{pmatrix},$$

$$D = \begin{pmatrix} -5 & -7 \\ -4 & 5 \\ 1 & -8 \end{pmatrix}, E = \begin{pmatrix} 2 & -7 \\ 9 & 8 \\ 2 & -2 \end{pmatrix}, F = \begin{pmatrix} 4 & -9 & -9 \\ 1 & 9 & -9 \end{pmatrix},$$

$$G = \begin{pmatrix} 0 & -4 \\ -3 & 6 \\ 4 & 5 \end{pmatrix}, H = \begin{pmatrix} -1 & 3 & -9 \\ -8 & 6 & -6 \end{pmatrix}, I = \begin{pmatrix} 5 & -3 & -2 \\ -3 & 0 & 9 \end{pmatrix},$$

$$J = \begin{pmatrix} -3 & 6 \\ 0 & 2 \\ -3 \end{pmatrix}.$$
studying particularly complex networks where attack often goes from a state and proceed to another according to a predetermined probability. DSGM uses a binary coding scheme for identifying game states and interaction diagrams are used to describe possible movements of players. Both models observed the probability of the attacker and defender at each state and for each game. Table 10 presents the probability of the attacker and defender in SSGM for each state leading to terminal state as a distinct game where the states represent game dimension size in DSGM. Figure 7 present a comparison of SSGM (PA) and SSGM (PD) probabilities. SSGM(PA) and SSGM(PD) are the probabilities of the attacker and the defender in SSGM respectively while DSGM(PA) and (DSGM(PD) are the probabilities of the attacker and the defender in DSGM respectively.

VI. CONCLUSION

The paper focused on adopting a deterministic stochastic game modeling strategies for both attacker and defender in a network environment. The deterministic nature of the model implies that when players provide a specific set of inputs at different time, exact set of outputs will be obtained for the same game. This is particularly important in situations where operational consistency is on the demand. Network Security Engineers and System administrators can use the resulting games and their solutions to formulate measures for solving interactive network security problems.

Since the intention of network attackers’ is to achieve a common goal (destruction), therefore, network and information security engineers as well as security researchers can use the proposed DSGM game and its solutions as basis for formal decision making, analytical tool and algorithm development as well as to predict attacker behavior. It can also be used as a rich mathematical tool to analyze and model new security problems. Moreover, through the security game, the defender can gain a deeper understanding of the attacker’s strategies and potentials.

The Stochastic Game Model (SSGM) proposed in [34] views the interaction between malicious users (attackers) and network administrators (defenders) as a two-player zero-sum game. In contrast, the proposed model is a Deterministic Stochastic Game Model that uses probabilistic approach for
REFERENCES


