Information Security Analysis Using Game Theory and Simulation

Bob G. Schlicher and Robert K. Abercrombie
Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA

Abstract - *Information security analysis can be performed using game theory implemented in dynamic simulations of Agent-Based Models (ABMs). Such simulations can be verified with the results from game theory analysis and further used to explore larger scale, real world scenarios involving multiple attackers, defenders, and information assets. Our approach addresses imperfect information and scalability that allows us to also address previous limitations of current stochastic game models. Such models only consider perfect information assuming that the defender is always able to detect attacks; assuming that the state transition probabilities are fixed before the game assuming that the players' actions are always synchronous; and that most models are not scalable with the size and complexity of systems under consideration. Our use of ABMs yields results of selected experiments that demonstrate our proposed approach and provides a quantitative measure for realistic information systems and their related security scenarios.*

Keywords: Information Security Analysis, Game Theory, Simulation, Confidentiality, Integrity, and Availability

1 Introduction

Title 44 of the U.S. Code [1] defines Information security as a means of protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide:

- confidentiality, which means preserving authorized restrictions on access and disclosure, including means for protecting personal privacy and proprietary information;
- integrity, which means guarding against improper in-formation modification or destruction, and includes ensuring information nonrepudiation and authenticity; and
- availability, which means ensuring timely and reliable access to and use of information

Today's security, economic, and industrial systems depend irrevocably on the security of myriad devices and the networks that connect them and that operate in ever-changing threat environments. Adversaries are applying increasingly sophisticated methods to exploit flaws in software, telecommunication protocols, and operating systems; to infiltrate, exploit, and sabotage weapon systems, command, control, and communications capabilities, economic infrastructure, and vulnerable control systems; or exfiltrate sensitive data, and to obtain control of networked systems in order to prepare for and execute attacks. Information security continues to evolve in response to disruptive changes with a persistent focus on information-centric controls. A healthy debate is needed to address balancing endpoint and network protection, with a goal of improved enterprise / business risk management.

Traditional network security solutions, typically employing firewalls and intrusion detection devices do not have a quantitative decision framework [2]. To this end, a few groups of researchers have started advocating the utilization of game theoretic approaches [2]. Game Theory provides mathematical tools and models for investigating multi-player strategic decision making. Another technique that is promising is the application of simulations [3].

1.1 The Problem

The motivation for this work, is highlighted by existing and emerging technologies that complement The Roadmap for Cybersecurity Research in context of survivability of time-critical systems [4] and the President's Comprehensive National Cybersecurity Initiative [5] with respect to extending cyber security into critical infrastructure domains.

The research and practicing community have been paying close attention to cyber security problems for more than two decades. However, Shiva et al. [6] state and it is generally agreed that the problem is far from being solved. In fact, some would argue that it is getting worse. As our dependence on the cyber infrastructure grows more complex and more distributed, the systems that compose it become more prone to failures and exploitation [7]. Failures in complex, tightly coupled systems can only be mitigated by collective decision making and organizational learning [8]. This is one way to view this game-theoretical approach.

The submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-00OR22725. Accordingly, the U.S Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

1.2 Paper Organization

In this paper, we first define information security and briefly review the weakness of traditional security solutions as they do not have a quantitative decision framework. We address the motivation for this work and in Section 2 introduce the game theory in the context of information security. In that section, we will also identify the distinguishing features of our approach to the subject domain. We then document four scenarios that will be the basis for the development of the Agent-Based Model (ABM) and its testing. An overview of the alignment of computational models and the challenges with comparing models is presented. This concept is important since when comparing models, one would want to know which features or capabilities are superior to other models. To address this subject, we pattern our ABM after two works which utilize a similar base, but approach game theory from different perspectives. We set our model up according to their assumptions and produce some interesting results. In the experimental section, we address the probability of successful attacks, and the tenants of information security dealing with confidentiality, integrity and availability. We conclude with a discussion on ideas for future work.

2 Related Work: Game Theory In Information Security

Roy et al. provide an excellent review of the taxonomy and different approaches to game theory as it can be applied to network security [2]. Recent work analyzes information security in the subject domain of e-commerce based on game theory with the introduction of the penalty parameter of the defender and the attacker [9]. This approach encourages the defender to invest in information security. Another recent paper defines and uses an analytical framework to analyze strategic choices and identify the best strategies and corresponding defenses used in virtual coordinate systems [10].

2.1 Limitations of Present Research

Many of the current game-theoretic security approaches are based on either static game models (e.g. Bayesian Formulation [2]) or games with perfect information or games with complete information. However, in reality a network administrator often faces a dynamic game with incomplete and imperfect information against the attacker. Some of the current models involving dynamic game with incomplete and imperfect information are specific to mobile ad hoc networks [11] while others do not consider a realistic attack scenario [2].

In particular, Roy et al. [2] point out that some of the limitations of the present research are: (a) Current stochastic game models [12] only consider perfect information and assume that the defender is always able to detect attacks; (b) Current stochastic game models [13] assume that the state transition probabilities are fixed before the game starts and these probabilities can be computed from the domain knowledge and past statistics; (c) Current game models assume that the players' actions are synchronous, which is not always realistic; (d) Most models are not scalable with the size and complexity of the system under consideration [2].

2.2 Distinguishing Features of Our Approach

2.2.1 Hypothesis and Explanation

Information security analysis can be performed using evolutionary game theory implemented in dynamic simulations of ABMs. Such simulations can be verified with game theory analysis results and further used to explore large-scale, real world scenarios involving multiple attackers, defenders, and information assets.

2.2.2 Simulation Approach

The simulation is based on ABM where the active components of the model, referred to as agents, engage in interactions on scenario-by-scenario basis.

2.2.3 Agent-Based Model (ABM) Overview

ABMs have been used to simulate evolutionary game theory involving multiple players in both cooperative and competitive or adversarial postures [14, 15].

ABMs bring significant benefits when: (1) interactions be-tween the agents are complex, nonlinear, discontinuous or discrete; (2) space is crucial and the agents' positions are not fixed; (3) the population is heterogeneous; (4) the topology of the interactions are heterogeneous and complex; or (5) the agents exhibit complex behavior, including learning and adaptation [14, 15].

The agents in the simulation include the attacker and the defender or administrator. The agents perform actions that can change the system state of the enterprise. For each state, agents are limited in the actions they can perform. Depending on the scenario, the attacker executes one of many actions with an associated probability of deciding to do the action and a probability that the action will be successful once the decision has been committed. Within each time unit, the simulator thread visits each agent giving them the opportunity to perform an action or not.

The administrator performs actions that begin with the probability of detecting something wrong with their enterprise. Since the enterprise state is known, the simulation limits the administrator's actions, which for the most part is a possible counter action to the most current action performed by the attacker. This is a reasonable assumption in that a competent administrator is assumed to be able to recognize a problem with their system. Before the administrator performs any counter action, a detection action is required to confirm the type of attack. In the simulation, our time unit represents one minute. We executed 1,000 simulations with each simulation spanning 250 simulated minutes. Experimental results were aggregated into bins and averaged to arrive at the probabilities of attack success.

Several scenarios are considered with a description of one of many sequence s that can be realized in the simulation as depicted in Table 1 for the scenario involving httpd being hacked by an attacker and recovered by the administrator. This scenario is used to gain an understanding for the agent interactions and the probabilities associated with decision points. There are many branches that the attacker can decide. Our ABM is flexible to accommodate arbitrary topologies and enterprise states. For familiarity, we have chosen the network topology for our analysis to those similar to Lye and Wing [12] and expanded upon in [13] and Wang et al. [16] as shown in Fig. 1.

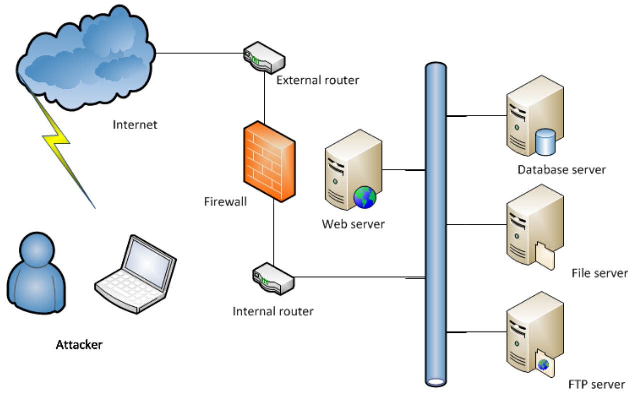

Figure 1. Enterprise network topology showing attach paths and enterprise servers.

In Table 1, we use P (a) to indicate the probability of taking the action and P(s) to indicate the probability of success of the action. P(s) is also the trigger for the state changes within the system. For example, at each time unit and for an attacker with the opportunity to continue_attacking, the attacker in Table 1 has a 0.5 uniform probability of deciding to perform that action and if so, has a 0.5 probability of succeeding. When the administrator performs a successful detection, in this case the detection of httpd being hacked, the payoff of -1 indicates that the recovery will not be considered until the next time unit. Hence the payoff of a negative value is interpreted as a delay in the number of time units. At that next time frame, the httpd is corrected with a 1.0 probability of action and a 1.0 probability of success. The payoff of -20 indicates a 20-minute duration to perform this action.

2.2.4 Overview of Challenges with Comparing Models

Computational models differ widely in their assumptions and implementations. These models must be normalized so their respective results can be analyzed to determine which model is more general and advances the field in question. This alignment is needed to determine whether two models can produce the same results, which in turn is the basis for critical experiments and for tests of whether one model can subsume another. This "alignment of computational models" has been referred to as “replicating” [17] or “docking” [18].

Table 1. Scenario 001 with Simulation Parameters

Scenario 001. httpd is hacked and recovered	**Simulation parameters and notes**
1. The attacker attacks an httpd process.	Attack_http, P(a)=0.5, P(s)=1.0
2. The attacker continues the attack to compromise the httpd.	continue_attacking, P(a)=0.5, P(s)=0.5
3. The attacker compromises the httpd system, httpd has been hacked.	State change to Httpd_hacked.
4. The admin detects the hacked httpd.	detect_httpd_hacked, P(a)=0.5, P(s)=0.5, payoff = -1.
5. The admin removes the compromised account and restarts httpd.	remove_compromised_account_restart_httpd, P(a)=1.0, P(s)=1.0, payoff= -20.

Table 2. Scenario 002 Deface Website with Admin Correction

Scenario 002. Deface Web Site
1. The httpd is hacked, but not recovered (see Scenario 001).
2. The attacker defaces the web site.
3. The admin detects the defaced web site.
4. The admin restores the website and removes the compromised account.

Table 3. Scenario 003 Denial Of Service (DOS)

Scenario 003. Denial of Service (DOS)
1. The httpd is hacked, but not recovered (see Scenario 001).
2. The attacker installs a sniffer and a backdoor program.
3. The attacker runs a DOS virus on the web server.
4. The network traffic load increases and degrades the system.
5. The admin detects the traffic volume and identifies a DOS attack.
6. The admin removes the DOS virus and the compromised account.

Table 4. Scenario 004 File Server Data Stolen

Scenario 004. File Server Data Stolen
1. The httpd is hacked, but not recovered (see Scenario 001).
2. The attacker installs a sniffer and a backdoor program.
3. The attacker attempts to crack the file server root password.
4. The attacker cracks the password; the file server is hacked.
5. The attacker downloads data from file server.
6. The admin detects the file server hack.
7. The admin removes the file server from the network.

Model equivalence testing is of central importance when comparing two or more computational models. The “equivalence” of models with stochastic elements must be defined in a precise statistical context. In many cases this is not trivial. There are at least two categories of equivalence beyond the initial criterion of numerical identity, (1) distributional and (2) relational equivalence. Distributional equivalence describes models that produce distributions that are statistically equivalent. Relational equivalence describes two models that produce the same internal relationship among their results [18, 19].

2.3 Attack Models

For certain defense techniques, some of the best attack strategies involve an inflation attack with varying percentages of malicious nodes [10].

2.4 Defense Models

From a defense posture, spatiotemporal and spatial outlier detections produce the best results against attacks. Temporal outlier detection in isolation is ineffective [10].

2.5 Experimental Plan to Test Hypothesis

We have focused our model on previous works that have documented several attack scenarios. The chosen case study was developed by interviewing network managers [12, 13]. Our enterprise network topology illustrated in Fig. 1 is quite similar to the previous papers [13, 16] and serves as our exploratory basis.

2.5.1 States

Our enterprise network is typical of many current configurations. Our model utilizes the following states from Lye and Wing [13] as follows:

1. normal_operation
2. httpd_attacked
3. httpd_hacked
    a. detect. hacked_detected
4. ftpd_attacked
5. ftpd_hacked
6. website_defaced
    a. detect. website_defaced_detected
7. webserver_sniffer
8. webserver_sniffer_detector
9. webserver_dos_1
    a. detect. webserver_dos_1_detected
10. webserver_dos_2
11. fileserver_hacked
    a. detect. fileserver_hacked_detected
12. fileserver_data_stolen_1
13. workstation_hacked
    a. detect. workstation_hacked_detected
14. workstation_data_stolen_1
15. network_shut_down

2.5.2 Actions

An action pair (one from the attacker and one from the administrator) causes the system to move from one state to another in a probabilistic manner. A single action of the attacker can be any part of his attack strategies, such as flooding a server with SYN packets or downloading a password file.

When a player takes no action, we denote the inaction as ø. Attacker consists of all the actions he can take in all the states. The actions can be described as:

- *Attack_httpd*
- *Attack_ftpd*
- *Continue_attacking*
- *Deface_website_leave*
- *Install_sniffer*
- *Run_DOS_virus*
- *Crack_file_server_root_password*
- *Crack_workstation_root_password*
- *Capture_data*
- *Shutdown_Network*

The action candidates in each state are taken from this list. For example, in the state Normal operation, the attacker has actions Attack_httpd, Attack_ftpd and ø.

In [13, 16] for similar actions taken by the administrator are mainly preventive or restorative measures. Our model uses the nomenclature provided in [13]. The actions of the administrator can be described in the following:

- *Remove_compromised_account_restart_httpd*
- *Restore_Website_remove_compromised_account*
- *Remove_virus_and_compromised_account*
- *Install_sniffer_detector*
- *Remove_sniffer_detector*
- *Remove_compromised_account_restart_ftpd*
- *Remove_compromised_account_sniffer*

The explanations of the above actions are similar to that of Lye and Wing [13], and Wang et al. [16]. Both papers assume that the administrator does not know whether there is an attacker or not, as we do. We also assume, as in [13, 16], that the attacker may have several objectives and strategies that the administrator does not know. Another realistic aspect of this model is assignment of probabilities of attack and success. Furthermore, not all of the attacker’s actions can be observed.

2.5.3 Parameter Modeling Set

Following the logic of our of our model of our typical enterprise network, Table 5 identifies the parameter modeling set that guided the data collection and analysis section for the attacker. Table 6 identifies the parameter modeling set that guided the data collection and analysis section for the defender administrator.

Table 5. Attacker Parameter Modeling Set

Action Name	Prob. Action	Prob. Success	Payoff	State From	State To
Attack_httpd	0.5	0.5	10	1	2
Continue_attacking	0.5	0.5	0	2	3
Deface_website_leave	0.5	0.5	99	3	6
Install_sniffer	0.5	0.5	10	3	7
Run_dos_virus	0.5	0.5	30	7	9
Crack_file_server-root-pw	0.5	0.5	50	7	11
Capture_data_file_-server	0.5	0.5	999	11	12
Shutdown_network	0.5	0.5	999	9	15

Table 6. Defender Administrator Parameter Set

Action Name	Prob. Action	Prob. Success	Payoff	State From	State To
Detect_httpd_hacked	0.5	0.5	1	3	3a
Detect_defaced_website	0.5	0.5	-1	6	6a
Detect_webserver_sniffer	0.5	0.5	-1	7	8
Remove_sniffer	1.0	1.0	0	8	1
Remove_compromised_-account_restart_httpd	1.0	1.0	10	3a	1
Restore_website_remove_-compromised_account	1.0	1.0	-10	6a	1
Detect_dos-virus	0.5	0.5	-1	9	9a
Remove_virus-and_-compromised_account	1.0	1.0	-3.0	9a	1
Detect_fileserver_hacked	0.5	0.5	-1	11	11a
Detect_fileserver_hacked	0.5	1.0	-1	11	11a
Remove_compromised_-account_restore_fileserver	1.0	1.0	-20	11a	1

3 Experimental Results

In this section we simulate the security of the enterprise network via the above model. We initially address what constitutes a successful attack and then address the confidentiality, integrity and availability of the enterprise network.

3.1 Security Analysis – Probability of a Successful Attack

The probability of a successful attack is determined by the parameter modeling set defined in Table 6. Fig. 2 illustrates the successful attacks in the enterprise network at each time interval (minutes), which is not cumulative.

Fig. 3 shows the same data as a cumulative distribution indicating when the probability of successful attacks reaches 1 for the arrival rates (0.13, 0.37, 0.65 and 0.94) respectively. This particular model illustrates that the attacker has a distinct advantage as the arrival rates of the attack increases.

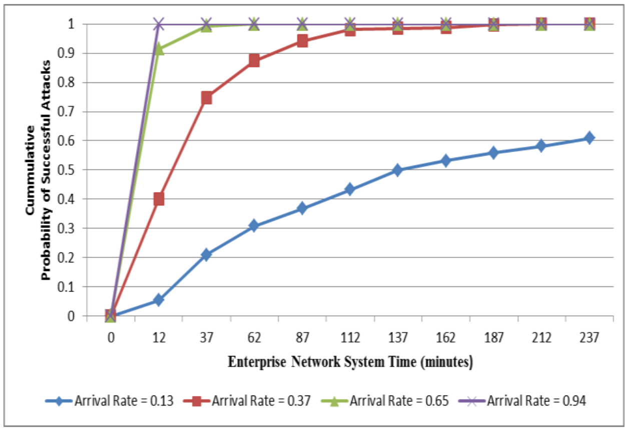

Figure 2. The probability of successful attacks cumulatively in the enterprise network per system time

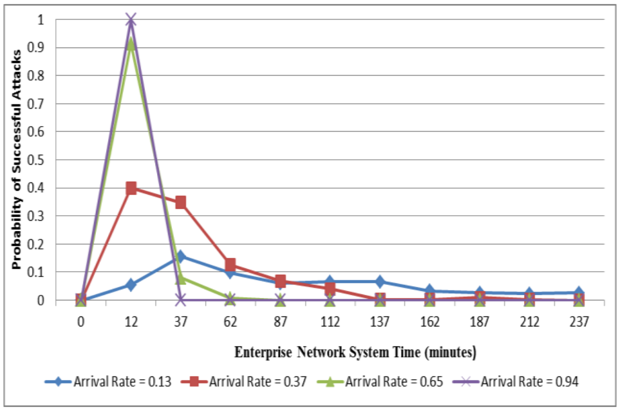

Figure 3. The probability of successful attacks in the enterprise network per system time.

3.2 Confidentiality

Wang et al. [16] define confidentiality as the absence of unauthorized disclosure of information. A measure of confidentiality is the probability that important data and information are not stolen or tampered. We adapt this logic to our model and the confidentiality can be shown as:

$$C = 1 - (P_{Fileserver\_data\_stolen} \times P_{Worstation\_data\_stolen}) \qquad (1)$$

where $P_{Fileserver\_data\_stolen}$ and $P_{Workstation\_data\_stolen}$ are the probability that the attacker succeeded in the "data stolen" category. Fig. 4 illustrates the confidentiality variation over time of the $P_{Worstation\_data\_stolen}$ to illustrate similar data trending. When compared to data in [16], it is clearly evident that our approach lends itself to the alignment of disparate models quite well.

3.3 Integrity

Wang et al. [16] define integrity as the absence of improper system alterations, preventing improper or unauthorized change. It is further described as the probability that the normal network services are affected or destroyed.

Our models follows Lye and Wing [13]. Integrity can be shown as:

$$I = 1\text{-}(P_{Website\_defaced} \times P_{Webserver\_DOS}) \qquad (2)$$

where P$_{Website\_defaced}$ and P$_{Webserver\_DOS}$ denote the probability in our model that the attacker succeeded in defacing the website, or inserting a virus and/or shutting down the network via the actions *Website_defaced* and *Webserver_DOS*. Fig. 5 illustrates the integrity dynamics of P$_{Website\_defaced}$ over time. Again the arrival rate of attacks has a profound effect on the dynamics of the probability of the particular website being defaced.

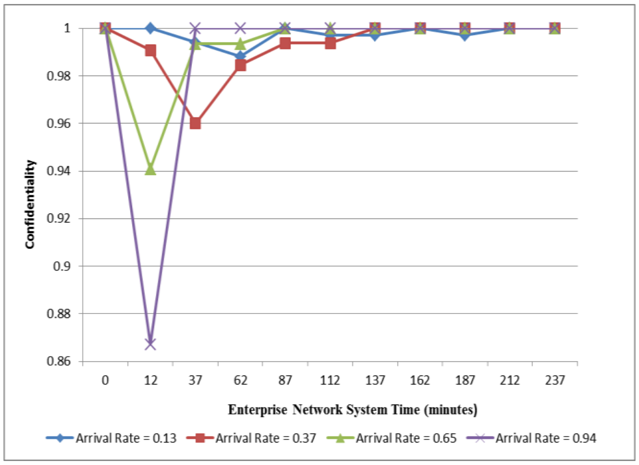

Figure 4. Confidentiality dynamics of $P_{Worstation\_data\_stolen}$ in the enterprise network.

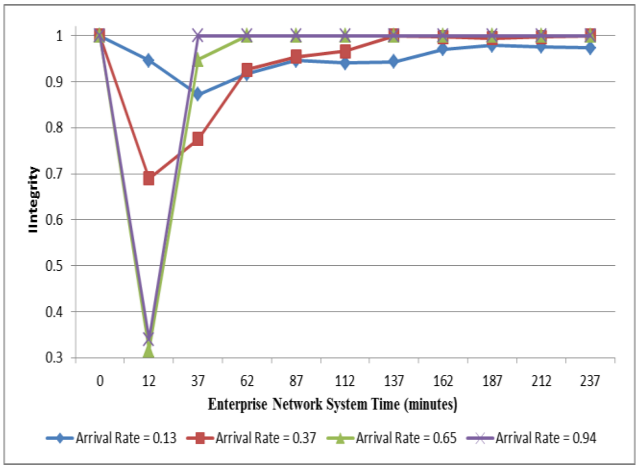

Figure 5. Integrity dynamics of $P_{Website\_defaced}$ in the enterprise network.

3.4 Availability

Wang et al. [16] define availability as systems being available when needed, and computing resources can be accessed by authorized users at any time. It is further described as whether the authorized users can access the information when necessary, when considering the probability that the normal network services are affected or destroyed. Our model differs from Wang et al. [16] with availability expressed as:

$$A = 1\text{-}(P_{Webserver\_DOS} \times P_{Network\_shut\_down}). \qquad (3)$$

Here P$_{Webserver\_DOS}$ denotes the probability the attacker succeeded in the defacing the website, or inserting a virus and/or P$_{Network\_shut\_down}$ denotes shutting down the network via the actions *Webserver_DOS* and *Network_shut_down*. Fig. 6 illustrates the availability variation.

Comparing and contrasting Figs. 4-6, we find confidentiality, integrity, and availability decrease at the beginning of the attack and then increase over time, as the administrator recovers from the attack. Therefore, it is crucial to the safety of the system that the administrator can discover the attack as early as possible.

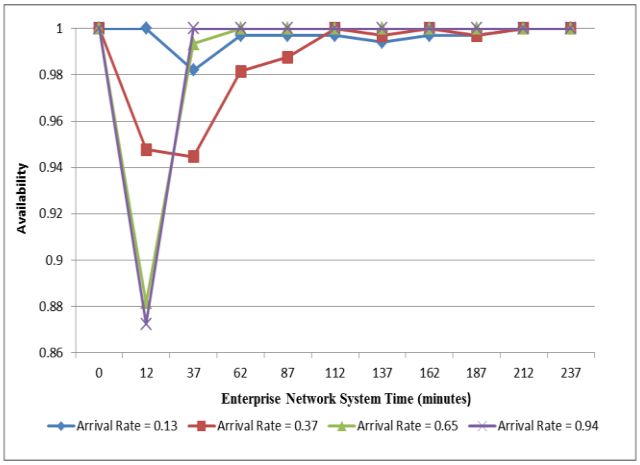

Figure 6. Availability dynamics of $P_{Webserver\_DOS}$ in the enterprise network.

4 Conclusion and Future Work

The main motivation for this work was that many of the current game-theoretic security approaches are based on either static game models (e.g. Bayesian Formulation [2]) or games with perfect information or games with complete information. In reality a network defender (the administrator) often faces a dynamic game with incomplete and imperfect information against the attacker. Some of the current models involving dynamic game with incomplete and imperfect information and others do not consider a realistic attack scenario.

In particular, Roy et al. [2] point out that some of the limitations of the present research are: (a) Current stochastic game models [12] only consider perfect information and assume that the defender is always able to detect attacks; (b) Current stochastic game models [12, 13] assume that the state transition probabilities are fixed before the game starts and these probabilities can be computed from the domain knowledge and past statistics. This second premise is the basis for our selection of the ABM approach. This allows us to expand our data collection beyond fixed probability values. Additionally, they state that current game models assume that the players' actions are synchronous, which is not always realistic and most models are not scalable with the size and complexity of the system under consideration [2]. As we embarked on the development of our ABM approach, great care was taken to duplicate exacting the underlining assumptions (transition probabilities) in the works by Lye and Wing [12, 13] and Wang et al. [16].

Our model is a simulation based on Agent Based Model (ABM) where the active components of the model, the agents, engage in interactions on scenario-by-scenario basis. The agents in the simulation include the attacker and the defender (or administrator). This is in contrast to the previous techniques that utilized nonlinear program in *MATLAB* [13] and Petri nets [16]. The agents perform actions that can change the system state of the enterprise. For each state, agents are limited in the actions they can perform. Depending on the scenario, the attacker executes one of many actions with an associated probability of deciding to do the action and a probability that the action will be successful once the decision has been committed. Within each time unit, the simulator thread visits each agent giving them the opportunity to perform an action or not. In our particular simulation, each time unit represented one minute. We executed 1,000 simulations with each simulation spanning 250 simulated minutes. Experimental results were aggregated into bins and averaged to arrive at the probabilities of attack success.

We believe that model equivalence testing (normalization) for comparing models is of central importance when comparing two or more computational models. The “equivalence” of models with stochastic elements must be defined in a precise statistical context. In some cases, this is not trivial. There are at least two categories of equivalence beyond the initial criterion of numerical identity, (1) distributional and (2) relational equivalence as described earlier. From a comparative perspective, our results matched modeling techniques that utilized nonlinear program in *MATLAB* [13] and Petri nets [16], even though our technique was quite dissimilar.

One interesting finding we discovered during the analysis of the results is in reality, damage can occur in other states, while the initial attack is being repaired. Future work will address this subject. We also plan to broaden the field of play, allowing multiple attacks to occur over the enterprise. An interesting theme will be to address unknown or zero-day attacks. The ABM approach will provide security analysts with a useful decision-making tool for information security. This tool will also provide security analysts and financial analysts a useful decision-making tool to augment analysis and investments decision making in the enterprise.

5 References

[1] "Public Printing and Documents," in *44 USC 3502*, ed. USA, 2009, p. 3542.

[2] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A Survey of Game Theory as Applied to Network Security," in *43rd Hawaii International Conference on System Sciences*, Koloa, Kauai, Hawaii, 2010, pp. 1-10.

[3] H. Gintis, *Thee Bounds of Reason: Game Theory and the Unification of the Behavioral Sciences*: Princeton University Press, 2009.

[4] "A Roadmap for Cybersecurity Research," Department of Homeland Security, Washington, DC November 2009.

[5] "The Comprehensive National Cybersecurity Initiative," The White House, Washington, DC 2010.

[6] S. Shiva, S. Roy, and D. Dasgupta, "Game Theory for Cyber Security," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, Oak Ridge, Tennessee, 2010, pp. 1-4.

[7] F. T. Sheldon, S. Prowell, R. K. Abercrombie, and A. Krings, "Cyber Security and Information Intelligence Challenges and Strategies Theme," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research.*, Oak Ridge, TN, 2010, pp. 1-8.

[8] C. Perrow, *Normal Accidents: Living with High-Risk Technologies*. Princeton: Princeton University Press, 1999.

[9] W. Sun, X. Kong, D. He, and X. You, "Information Security Problem Research Based on Game Theory," in *2008 International Symposium on Electronic Commerce and Security*, Guangzhou City, 2008, pp. 554-557.

[10] S. Becker, J. Seibert, D. Zage, C. Nita-Rotaru, and R. State, "Applying Game Theory to Analyze Attacks and Defenses in Virtual Coordinate Systems," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks*, Hong Kong, 2011, pp. 133-144.

[11] A. Patcha and J.-M. Park, "A Game Theoretic Formulation for Intrusion Detection in Mobile Ad Hoc Networks," *International Journal of Network Security*, vol. 2 (no. 2), pp. 131-137, 2006.

[12] K.-w. Lye and J. M. Wing, "Game Strategies in Network Security," in *Proceedings of the Workshop on Foundations of Computer Security*, 2002.

[13] K.-w. Lye and J. M. Wing, "Game Strategies in Network Security," *International Journal of Information Security*, vol. 4, pp. 71-86, 2005.

[14] E. Bonabeau, "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems," *Proceedings of National Academy of Sciences*, vol. 99 Suppl 3, pp. 7280-7287, 2002.

[15] M. A. Nowak and R. M. May, "The Spatial Dilemmas of Evolution," *International Journal of Bifurcation and Chaos*, vol. 3, pp. 35-78, 1993.

[16] Y. Wang, M. Yu, J. Li, K. Meng, C. Lin, and X. Cheng, "Stochastic game net and applications in security analysis for enterprise network," *International Journal of Information Security (online First)*, pp. 1-12, October 22, 2011.

[17] U. Weldnsky and W. Rand, "Making Models Match: Replicating an Agent-Based Model," *Journal of Artifical Societies and Social Simulation*, vol. 10(4)2, 2007.

[18] R. Axtell, R. Axelrod, J. M. Epstein, and M. D. Cohen, "Aligning Simulation Models: A Case Study and Results," *Computational & Mathematical Organization Theory*, vol. 1, pp. 123-141, 1996.

[19] J. Xu, Y. Gao, and G. Madey, "A Docking Experiment: Swarm and Repast for Social Network Modeling," in *Seventh Annual Swarm Researchers Conference (Swarm2003)*, Notre Dame, 2003, pp. 1-9.