Game Theory for Cyber Security

Sajjan Shiva
Computer Science Department
University of Memphis
Memphis, TN 38152
1-901-678-5667
sshiva@memphis.edu

Sankardas Roy
Computer Science Department
University of Memphis
Memphis, TN 38152
1-901-678-5465
sroy5@memphis.edu

Dipankar Dasgupta
Computer Science Department
University of Memphis
Memphis, TN 38152
1-901-678-4147
ddasgupt@memphis.edu

ABSTRACT[1]

While there are significant advances in information technology and infrastructure which offer new opportunities, cyberspace is still far from completely secured. In many cases, the employed security solutions are ad hoc and lack a quantitative decision framework. While they are effective in solving the particular problems they are designed for, they generally fail to respond well in a dynamically changing scenario. To this end, we propose a holistic security approach in this paper. We find that game theory provides huge potential to place such an approach on a solid analytical setting. We consider the interaction between the attacks and the defense mechanisms as a game played between the attacker and the defender (system administrator). In particular, we propose a *game theory inspired defense architecture* in which a game model acts as the brain. We focus on one of our recently proposed game models, namely *imperfect information stochastic game*. Although this game model seems to be promising, it also faces new challenges which warrant future attention. We discuss our current ideas on extending this model to address such challenges.

Categories and Subject Descriptors

H.0 [**Information Systems**]: General

General Terms

Cyber Security, Game Theory, Attacks, Countermeasures

Keywords

Stochastic Games, Imperfect Information, Learning

1. INTRODUCTION

The research and practicing community have been paying attention to the cyber security problem for more than two decades. However, the problem is far from being completely solved. The main limitation of the current cyber security practice is that the security approach is largely heuristic, increasingly cumbersome, and it is struggling to keep pace with rapidly evolving threats. The core security breaches occur in terms of confidentiality, integrity, and availability.

CSIIRW'10, April 21-23, Oak Ridge, Tennessee, USA.
Copyright © 2010 ACM *978-1-4503-0017-9*…$5.00.

To overcome these problems, four types of security endeavors have been employed in the past.

(a) **Secure Communication Infrastructure**: We have seen cryptographic algorithms being designed and used to build secure networking protocols such as Internet Protocol Security (IPSEC), Transport Layer Security (TLS) or its predecessor Secure Sockets Layer (SSL), and Virtual Private Network (VPN). Although crypto is a crucial mechanism for security, it is not a panacea. If one end point of the secure communication infrastructure is compromised, then crypto will not help us.

(b) **Monitoring or Response System**: The research community has spent huge amount of effort to build a monitoring or response system. Firewalls, network-based IDSs, host-based IDSs and anti-virus programs have been widely deployed. Firewalls are generally incapable of protecting against attacks on the application-layer. In some cases, it is imperative that *deep packet inspection* be performed. Furthermore, with the advent of the virtualization technology, researchers are advocating to host applications on a virtual machine X, so that all activities in X could be observed by a monitor application residing outside X. Nevertheless, a perfectly safe monitor is yet to be designed.

(c) **Built-In vs. Bolt-On Approaches in System Development**: In the Built-In approach, security features are designed up front and form part of the system development. The Bolt-On approaches compensate for the mistakes made earlier in the development cycle. Although considered as a weaker option, to secure a legacy system we resort to the Bolt-On approach.

(d) **Code Instrumentation Tools and Self Checking Modules**: Researchers have designed techniques to enforce data and control flow integrity of a software or hardware component (Castro 2006). These techniques compute a flow graph using static or dynamic analysis, and instrument the program to check if the execution at runtime confirms to the flow graph. A few researchers (Chang 2001) also proposed to develop self-checking software which can detect its errors by verifying the data/control flow integrity within itself. However, these techniques are not generally effective against polymorphic exploits.

A Holistic Security Approach: Despite the past considerable effort to protect the cyber space as summarized above, hacking endeavors still grow in numbers and sophistication, which strongly indicates that we need a game-changing strategy. We have to accept the fact that there is no panacea to overcome the ever-growing plethora of cyber security problems. It is literally an ongoing war between the system administrators and the hackers, which is simultaneously open in several frontiers.

We propose a holistic security approach which suggests the system administrator to take the full picture in mind and make a

[1] This work is supported by ONR under grant N00014-09-1-0752.

thorough analysis of the security threat to the whole system, instead of securing the system part by part. We envision a 4-layer approach illustrated in Figure 1. The layers are defined as follows: (i) At the innermost layer are the core hardware and software components. We envision each of these components having a provision of being wrapped with a self-checking module (with inspiration from the traditional BIST architecture). We call them Self-checking HW/SW components. (ii) At the 2nd layer lies the traditional network security infrastructure which is built using techniques such as cryptographic algorithms. (iii) At the 3rd layer reside the secure applications which are designed with Built-In or Bolt-On security approaches utilizing self-checking concepts and components. (iv) Most importantly, we envision a game theoretic decision module at the top layer which has the responsibility of choosing the best security strategy for all the inner 3 layers. We observe that in the past the research community has put majority of their effort only for the 2nd and the 3rd layers. Traditional intrusion detection system can be considered as residing in the top layer, which can be made more effective by the use of game theory. Note that our layered view is an operational one and does not have any direct relationship with the traditional ring-oriented privilege separation principle or the OSI network stack.

The crux of our holistic security approach is utilization of game theory. Game theory can choose the security measures which make the best tradeoff between the incurred cost and level of security achieved. The cost includes the investment expenditure as well as the performance degradation of the system due to the extra load of the security action. There are two kinds of security decisions—one that is statically chosen which has an investment cost or performance penalty and one which is dynamically selected which causes some performance degradation.

Some examples of the statically selected decisions (where each one has a tradeoff) are: (a) choosing the access control policy being a discretionary access control (DAC), or a mandatory access control (MAC), or a roll-based access control (RBAC); (b) whether to deploy x number of firewalls in certain locations in an enterprise network topology; (c) whether to use http or https protocols to implement the remote login process; (d) whether only VPN connections should be allowed and if so, then whether to use IPSEC or SSL as the underlying technology; (e) whether to enable the self-checking software functionality of certain software and hardware components.

Few of the dynamically selected decisions are: (a) when a web-server observes a malformed http request which matches an attack signature, the http connection should be dropped or it should be forwarded to a honeypot; (b) when a compressed http request is observed, should we turn on the deep packet inspection module for the related whole session; (c) under SYN-flood attack of rate z whether the server should apply mitigation techniques such as SYN Cookies or the table of SYN entries in the kernel to be dynamically resized; (d) after detecting email spams pouring in with rate s, whether the proxy mail server should blacklist the source address (e) after a rootkit is detected in a web-server W, whether W should be disconnected from the network right then.

Our observation is that to achieve reliable security we have to consider the interaction of the security decision for one component in the system with the policy taken for others, and as a result, the decision space becomes large. Game theory is a potential tool to model and analyze such an enormous search space involving numerous what-if scenarios. It can also model the inherently selfish and competitive behaviors of the attacker and system administrator and analyze the possible strategies. In addition, game theory has the capability of examining hundreds of thousands of possible scenarios before taking the best action; hence, it can sophisticate the decision process of the network administrator to a large extent.

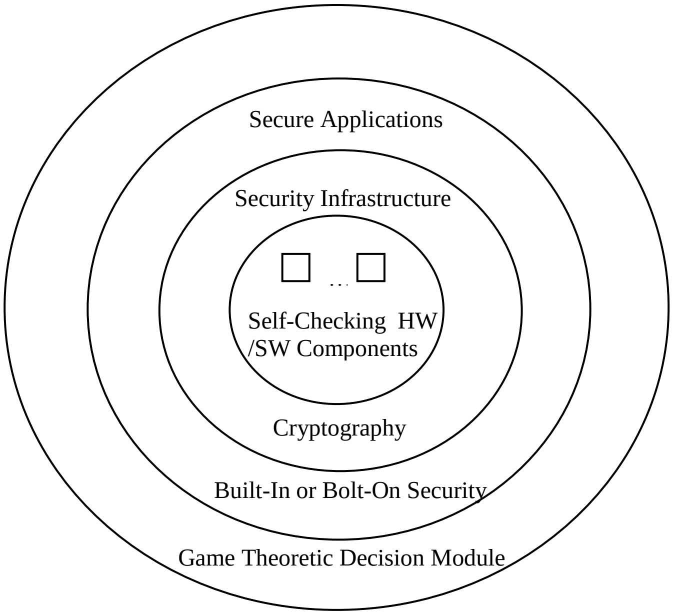
Secure Applications
Security Infrastructure
...
Self-Checking HW /SW Components
Cryptography
Built-In or Bolt-On Security
Game Theoretic Decision Module

Figure 1: A Holistic Security Approach

A survey on the existing game theoretic security solutions which have addressed some of the challenges is available in (Roy 2010). To date, many researchers have designed security mechanisms using game theory—several directions have been explored such as using a stochastic game to model network security scenarios (Lye 2005), modeling DDoS attacks (Wu 2010), security of physical and MAC layers (Sagduyu 2009), intrusion detection systems (Alpcan 2006), and so on.

The major contributions of this paper are summarized below: We propose a holistic security approach and present a game theory inspired defense architecture (GIDA) which utilizes a game model as its brain. We briefly review our imperfect information stochastic game model and present our current ideas on extending this model.

In Section 2, we discuss GIDA. In Section 3 we present our imperfect information game model, and in Section 4, we discuss possible extension of this model to address existing challenges.

2. A Game Theory Inspired Defense Architecture (GIDA)

We envision a semi-autonomous defense architecture which leverages a game theoretic model to counter cyber attacks. We suggest the system administrator to take a "carrot and stick" approach to guard against the adversary. Carrot and stick approach refers to a policy of offering a combination of rewards and punishment to induce the adversary behavior. The brain of GIDA is a game model which decides the best countermeasure after a thorough analysis of the cost and reward as discussed in Section 1. The game model is not specific to any particular attack and countermeasure. As an example, we can envision wrapping a self-testing software module over individual components of the system with a tradeoff among the cost, security, and performance.

We envision GIDA as being a distributed architecture and consisting of three key components: A set of game agents along with the central game coordinator, an administrative console, and a dynamic honeynet. These three components interact in a semi-autonomous fashion in order to provide a means to identify, evaluate, and act upon network flows as illustrated in Figure 2. The honeynet in particular, provides a means to redirect malicious flows into dynamically instantiated honeypots for observation of malicious activity and the forensic data pertaining to it. Finally, the administrative console will provide a user interface that will allow the correlation of the network state data, provide a control channel for messaging, perform forensic analysis of honeypot data, and configure the various components.

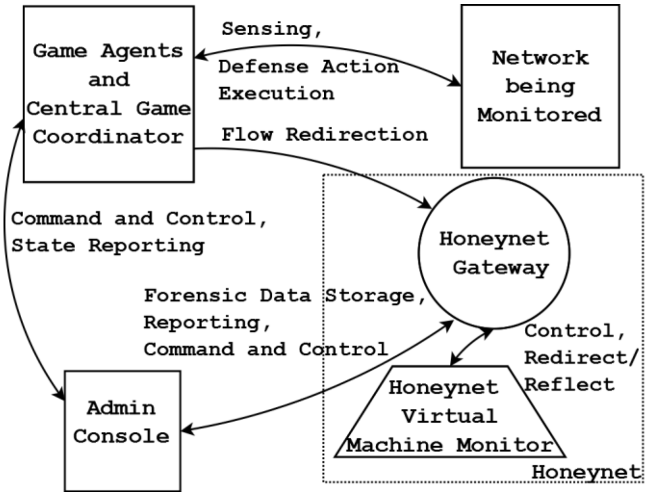
Game Agents and Central Game Coordinator
Sensing,
Defense Action Execution
Network being Monitored
Flow Redirection
Command and Control, State Reporting
Honeynet Gateway
Forensic Data Storage, Reporting, Command and Control
Control, Redirect/ Reflect
Honeynet Virtual Machine Monitor
Admin Console
Honeynet

Figure 2: Various components of GIDA and their relationship

3. Imperfect Information Stochastic Game

We recently proposed two game theory models to address some of the challenging cyber security issues. The first model uses the static and dynamic games and is specific to the class of DDoS attacks and their potential countermeasures (Wu 2010), while the second model is based on an imperfect information stochastic game which fits to the generic cyber security scenario. Below we briefly review our stochastic game model whose details are available in (Shiva 2010).

To model cyber attacks and defense mechanisms, a stochastic game model was previously proposed in the literature (Lye 2005). The state of the game probabilistically changes depending on actions taken by the players (i.e., type of attacks and defender's countermeasure) and the system configurations. During each state transition, each player gets a payoff or incurs some cost (negative payoff). Techniques exist by which a player can determine the best strategy to get the highest overall payoff considering all of the possible strategies of the adversary. Game theoreticians formulate the solution concept of a stochastic game by the notion of Nash equilibrium, and already provided the analysis indicating the existence of the equilibrium (Filar 1997).

As stated, the prior stochastic game models for network security (Lye 2005) assume that the players have perfect information about the current state of the game, which implies that the defender is always able to detect an attack and the attacker is always aware of the employed defense mechanism. In real systems, a player uses a sensor (e.g., the defender's sensor can be a part of the Intrusion Detection System (IDS)) to observe the current status of the system to decide the strategy. It is widely believed that no real sensor can perfectly read the environment, i.e., usually there is a non-zero error probability. So, in most cases, the above assumption about perfect information does not hold in real life. We relaxed this assumption and designed a stochastic game model which is able to capture more realistic scenarios. It considers that a player knows the system's true state at a particular moment with some error probability, i.e., at any given point of time the true state and a player's perception can be potentially different. With this additional constraint of imperfect information, we computed the best strategy for a player considering other players' choice of possible strategies.

In particular, we presented a theoretical analysis by which the defender can compute his/her best strategy to reach the Nash equilibrium of a stochastic game assuming the defender's sensor is imperfect. It is implicit that the defender knows the error probability of his sensor and the players' objectives are directly opposite, i.e., it is a zero-sum game. Moreover, we showed that if the defender follows the strategy prescribed by the perfect information model, then the Nash equilibrium is not achieved and the attacker's payoff can be more. Our algorithm for computing the best strategy runs offline well before the game is being played, i.e., our game analysis is static. Furthermore, our theoretical results are validated via simulation experiments in MATLAB.

4. Extension to Imperfect Information Stochastic Game Model

In (Shiva 2010), we assumed that the defender knows his sensor's error probabilities. In particular, for the working example in that paper, we assumed that the defender is aware of the false positive (FP) ratio γ_1 and false negative (FN) ratio γ_2 and then investigated how the defender can utilize this information to decide the optimal strategy. It was an offline (or static) analysis from each player's point of view and no learning was involved. We now relax the assumption and focus on how the defender can figure out his best strategy without knowing his sensor's error probabilities.

Our first approach involves learning algorithms—we allow the defender to guess a potential strategy and observe the outcome. Then, depending on the previous actions, observations and payoffs, he modifies his strategy in the current stage and again observes the outcome. After a few iterations as the above, he could eventually learn the best strategy. We are now utilizing a learning technique, known as *min-max Q* (Littman 94), to aid in the gradual improvement of the defender's decision quality. Note that Alpcan et al. have used the min-max Q-learning approach (Alpcan 2006). However, as stated in that paper, their model can only handle reactive defense actions, whereas in our model, pro-active defense measures are allowed.

Further note that the traditional min-max Q-learning algorithm (Littman 94) assumes that the players know the current state of the game with certainty—there is no error in sensing the current game state. The imperfectness in information of a player is in terms of knowing the transition probabilities of the stochastic game. We have modified the original algorithm and adapted it for players' sensors' imperfect information about the current state. The research issue is how many game iterations are needed for the defender to learn the best strategy. The challenge is how to reduce the number of iterations spent in this learning phase. We also aim to ensure that during this learning phase the defender always choose a safe strategy so that he does not incur big loss in the learning phase itself, which might nullify the benefit of learning. We stress that Q-learning is just one technique in the class of the reinforcement learning algorithms (Shoham 2007) in which an agent can learn a strategy of his own that does well against the opponents, without explicitly learning the opponent's strategy. We are also studying other body of work on multi-agent learning in a zero-sum game (Kearns 2002). Furthermore, we are exploring the most recent efforts in this line of research as extended to general-sum games such as Nash-Q by (Hu 2003).

Table 1. Summary of the Imperfect Information Stochastic Game Models

Models	Zero or General-Sum?	Involves Agent learning?	Limitation
Min-max Q (Littman 94)	Zero-Sum	Yes	Assumes no state sensing error
Alpcan et al. (2006)	Zero-Sum	Yes	No proactive defense
Kearns et al. (2002)	Zero-Sum	Yes	Assumes no state sensing error
Nash-Q (Hu 2003)	General-Sum	Yes	Restrictive convergence
Our 1st approach	Zero and General-Sum	Yes	Work in progress
POSG (Hansen 2004)	Equal-Payoff	No	Finite horizon
ε-equilibrium (Ganzfried 2009)	Zero-Sum	No	Restrictive convergence
Our 2nd approach	Zero and General-Sum	No	Work in progress

In addition to design a learning algorithm, we are currently exploring what happens if the players' sensor's error ratio varies with scenarios or time, i.e., what the outcome will be if the error ratio follows a Gaussian distribution such as $FN \sim \mathcal{N}(\gamma_1, \sigma_1)$ and $FP \sim \mathcal{N}(\gamma_2, \sigma_2)$. We are studying the dependency of the equilibrium payoff on σ_1 and σ_2. Our current study indicates that the benefit of learning the optimum strategy vanishes if σ_1 and σ_2 cross a threshold. We will present our theoretical and experimental results related to this threshold in the full paper.

Alongside the reinforcement learning discussed above, in our second approach, we are currently investigating another research domain which is known as the *partially observable stochastic games* (POSGs). Hansen et al. proposed a dynamic programming algorithm for solving a POSG, which uses iterated elimination of dominated strategies in normal form games with hidden states (Hansen 2004). Basically, they redefined the transition probability function of a stochastic game involving an *observation set*. However, this algorithm is prohibitively computation-extensive because the search space of the algorithm grows very rapidly with the length of the game horizon. We are investigating if we can make the efficiency better for the special case of zero-sum games exploiting some unique properties of this game. Furthermore, we are investigating how to extend their algorithm to be applied for infinite-horizon POSGs. We are also investigating the applicability of (Ganzfried 2009)'s *ε-equilibrium* computation algorithms in our game scenario, which were especially designed to model a famous cards game, Poker. Table 1 compares the prior imperfect information models with our current approaches.

In addition, we will study how to leverage the existing works of attack graphs and scenario graphs to generate game states and action sets of the attacker and the defender at each state. We will investigate the performance of our imperfect information stochastic game model with large state space whereas each player's action set is moderately wide. We will also explore how to efficiently solve a general-sum POSG designed in the context of real cyber scenarios. Finally, we will design the yardsticks to compare the performance of game-theoretic security solutions.

5. REFERENCES

[1] Alpcan T. and Baser T. 2006. An intrusion detection game with limited observations." 12th Int. Symp. on Dynamic Games and Applications. Sophia Antipolis, France.

[2] Castro, M, Costa, M. and Harris, T. 2006. Securing Software by Enforcing Data-flow Integrity. Proc. of the 7th OSDI.

[3] Chang, H. and Atallah, M.J. 2001. Protecting Software Code by Guards. Proc. of 1st ACM Workshop on Security and Privacy in Digital Rights Management.

[4] Filar, J. and Vrieze, K. 1997. Competitive Markov decision processes. New York: Springer.

[5] Ganzfried, S. and Sandholm, T. 2009. Computing Equilibria in Multiplayer Stochastic Games of Imperfect Information", Proc. of the Intnl. Joint Conference on Artificial Intelligence.

[6] Hansen, E.A. and Bernstein, D.S. and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games".Proc. of the National Conf. on Artificial Intelligence.

[7] Hu, J. and Wellman, M. 2003. Nash Q-learning for general-sum stochastic games. J. of Machine Learning Research (4).

[8] Kearns, M. and Singh, S. 2002. Near-optimal reinforcement learning in polynomial time". Machine Learning 49(2).

[9] Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning". Proceedings of the 11th International Conference on Machine Learning.

[10] Lye, K. and Wing J. 2005. Game strategies in network security." Intnl J. of Information Security 4 (02): 71--86.

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

[12] Sagduyu, Y.E. and Berry, R. and Ephremides, A. 2009. MAC Games for Distributed Network Security with Incomp. Inf. of Selfish and Malicious User Types. GameNets.

[13] Shiva, S., Roy, S., Bedi, H., Dasgupta, D., and Wu, Q. 2010. An Imperfect Information Stochastic Game Model for Cyber Security. The 5th Intnl Conference on i-Warfare and Security.

[14] Shoham, Y., Powers, R., and Grenager, T. 2007. If multi-agent learning is the answer, what is the question? Artificial Intelligence, Elsevier.

[15] Wu, Q., Shiva, S., Roy, S., Ellis, C., and Datla, V. 2010. On Modeling and Simulation of Game Theory-based Defense Mechanisms against DoS and DDoS Attacks, SpringSim.