The Third International Conference on Availability, Reliability and Security

Boosting Markov Reward Models for Probabilistic Security Evaluation by Characterizing Behaviors of Attacker and Defender

Zonghua Zhang, Farid Naït-Abdesselam
IRCICA/LIFL - CNRS UMR 8022 - INRIA Futurs
University of Sciences and Technologies of Lille, France

Pin-Han Ho
Department of Electrical and Computer Engineering
University of Waterloo, Ontario, Canada

***Abstract*— While Markov Reward Models (MRMs) have been widely used for system dependability evaluation, their application for evaluating security still poses as a challenge. It is observed that attacker behavior plays a key role in causing models of security evaluation to be complicated. Another observation is that representing attacker behavior in terms of attack effects instead of attack itself enables the system security to be indirectly evaluated by identifying families of attacks rather than individual instantiations. Furthermore, an attacker behavior tends to be affected by defense mechanisms (we say defender) due to their close interactions. These observations motivate us to boost MRMs to the security context by extracting the behaviors of attacker and defender. To do that, we present a general yet simple state-based approach to characterizing and inferring the behaviors of attackers and defenders in typical network attacks. It specifically contributes in two folds: 1) two objective-oriented models are developed to measure the attacker's and defender's behaviors, respectively; 2) the objectives, actions, and the resultant effects by the attacker and defender, along with the underlying system states, are then integrated and formulated as Partially Observable Markov Decision Processes. The developed models and analysis allow the behaviors of attacker and defender to be characterized in a fine-grained way, and specific attack-defense strategies to be inferred approximately via existing model-based algorithms. The system security hereby can be indirectly validated on the basis of the aggregated effects resulted from the interactive behaviors of attacker and defender. A real trace study is conducted to show feasibility and effectiveness of our proposed approach.**

I. Introduction

As a desired capability of an information system is to perform its intended function and preserve essential security properties in the presence of any attack, the quantification of the security of a given design is a critical task. A straightforward approach is to develop comprehensive and systematic methodologies for examining and validating system's security properties. The increasingly complexity of today's computer systems, however, causes them to be system-specific and a meticulous process. Alternatively, it can be done via modeling behaviors of attackers in terms of attack effects.

The abundant research literature on system dependability evaluation [16] have shown that excluding the user-driven activities and their effects, most evaluation models in the dependability context can be introduced to security domain with proper modification [2], [16], such as model-checking [3], [17], fault trees [5], and stochastic analysis [18], [20]. It is the user, either malicious or legitimate or both, which causes security evaluation to be more complicated. Compared with the specific analysis of attack characteristics, we believe the measuring of attacker behaviors in terms of such elements as intent, objective and attack effect, has more potential to achieve a higher level of representation covering a large class of attacks (including novel ones), thereby promoting security evaluation significantly.

In practice, however, the development of such approaches should not only consider attacker but also the system itself and associated defender due to their close interactions. This is an important observation for describing and analyzing the system security and attacker/defender's behavior. Therefore, we envision that the states of an information system under intrusion can be quantified using a direct, high-level measure of security attributes like availability and integrity, leading to the following advantages: 1) the system security can be modeled through either known or unknown vulnerabilities, 2) the attacker/defender's behavior can be modeled and measured through their interactions, and 3) the operational measures of attacker and defender can be used to quantify the system states regarding security attributes under the aggregated effects of attack and defense.

This paper presents a state-based approach to characterize and infer the behavior of defender/attacker in multi-stage cyber attacks. First, two goal-directed models are developed to measure the attacker's behavior in terms of intents and attack cost, and to quantify the defender's behavior in term of some cost factors associated with particular countermeasures, respectively. Second, the objectives of attacker/defender, along with the underlying system states, are then integrated and formulated as Partially Observable Markov Decision Processes (POMDP). Based on the developed models, we will show that the behaviors of attacker/defender can be characterized in a fine-grained way, and the specific attack/defense strategies can be inferred via appropriate model-based algorithms. Although the model construction involves a set of parameters, we find there is no rigorous requirement on their accuracy, their relative values rather than absolute ones determine the results. In general, our proposed approach can solve the following two issues: (1) on the basis of the threat/vulnerability assessment and the assumed attackers' objective, we can figure out the most significant elements (vulnerability, a specific host/service, etc.) of the target system, enabling us to predict the attackers' actions, such as what type of attacks or which strategies are more likely to be launched; (2) the modeling and analysis of attacker behavior allow a defender to detect pieces of an ongoing attack and to predict the potential actions, a sophisticated attack thus can be decomposed into several atomic ones with particular subgoals. Also, the results may enable the probabilistic validation of security with respect to high-level system properties.

The remainder of the paper is organized as follows. In Section II, we investigate the related work. Section III proposes two goal-directed models for attackers and defenders respectively. In Section IV, we present a model-based approach for modeling and characterizing the behaviors of attacker and defender. Section V addresses behavior inference for security evaluation. Section VI reports a simulation-based real trace study to demonstrate its application to security validation in practical scenarios. Section VII concludes our work and points out open problems.

0-7695-3102-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ARES.2008.75

IEEE computer society

II. Related Work

Markov reward models (MRM), a kind of state-based technique, have been widely used for quantitative, model-based evaluation of computer system dependability [3], [4], [16]. A challenging issue for MRM models is the huge amount of states called *largeness* problem, especially for complex systems encompassing large numbers of components. Another challenge impeding their applications to security domain lies in the fundamental difference between dependability and security issues: dependability evaluation commonly assumes the faults are caused by system themselves, while in security evaluation, malicious attacks are generally intentional and human-driven.

A novel work on the application of Markovian models to attacker behavior characterization is [10], in which "working" and "security failed state" were specified as two states to support the security measurement. A Markov model was built in [6] to quantify the attacker behavior using a two-year-long experimental data, this work presented us some empirical supports to the modeling of attacker behavior from a high-level perspective, even though its focus is at the fine-grained state transitions. More recent work are [13], [19], which used stochastic activity networks and semi-Markov model respectively to validate and evaluate the intrusion-tolerant systems. The emphasis of the former work is the effects of intrusions and the system behavior responding to such effects, and the latter one focuses on the quantification of the measure of the system security in terms of availability, integrity, and confidentiality.

Compared with the above state-based attacker modeling works, our work has some distinctive features. First, the key elements of the models are different. Our model treats system states implicitly via observations emitted by the key discrete states rather than the continuous ones, avoiding state space growing prohibitively large. The rewards of our model are defined to capture attacker and defender's objectives rather than underlying system states or their transitions. Second, the abovementioned models are tailored to the specific systems for example, intrusion-tolerance systems, while our work focuses on the general systems with assumed vulnerabilities, exploits and security measures. Third, not only concerns the effects caused by attacks, our model also systematically integrates the defender's action effects, as well as the temporal interactions between attacker and defender. In general, our model has several components: it contains representations of attackers and defenders, as well as the system assets and resources; it represents attacker and defender behavior as decision-making process, and the process evolution is directed by the objectives, which are extracted and represented using the system security metrics in terms of high-level security attributes.

Some other useful techniques for the modelling of attacker behavior include [11], [12], in which game-theoretic models were extensively discussed to deal with attack strategies and defense mechanisms by viewing attacker and defender as two game players. The best defense posture was inferred via *Nash equilibrium* based on the specification of system states and their transitions. More importantly, the latter work presents a general concept, *utilities*, to integrate the attacker's various intents, objectives and costs, and a game-theoretic model was developed to capture the inherent interdependency between attacker's objectives and strategies and those of defenders. However, the game theoretical models treat the temporal correlation between attacker actions loosely and may take much time to infer the strategies of a multi-stage attack. Moreover, the objective of the attacker in such game-theoretic models is to achieve the greatest payoff (as well as defender) regarding the system including defense component as a whole, and the strategies can only be inferred at the balanced state between attacker and defender. Henceforth, the model can hardly help us to insight the attack schemes (some atomic ones) individually, and provides no way to figure out the key elements of a particular attack scheme. Also, the high computational cost usually limits their application in practice. Our proposed model tackles these problems well.

III. Goal-directed Behavioral Modeling

This section develops two objective models to characterize attacker and defender behavior respectively by capturing the key elements in particular attack-defense scenarios.

A. Characterizing Rational Attacker

As social criminals, no attacker in cyberspace launches attack without any incentive. In another word, an attacker must enforce attacks for achieving some malicious goals. During an attack, the attacker must take a set of actions based on his/her observations, meanwhile observes the subgoals in terms of effects that have been achieved. Moreover, for a rational attacker, he/she has to consider the cost associated with actions when launching an attack scheme, such as assisting tools (hardware, programs, trust relationships with insider, etc.), the risk of being detected, and so on. Therefore, a well-planned attack scheme can be viewed as a strategic decision process acted by an attacker, which incorporates attacker's intent, action, and action effects. Suppose an attack as a *venture investment*, where the *objective* of an attacker is to earn $reward$ as much as possible by paying some *cost*. The attacker intent associated with the malicious goals can be regarded as *gross profit*. Then their relationship is measured as $Reward = GrossProfit - Cost$. A particular *investment* must fall into two cases (1) cost is constant, profit to be maximized, (2) profit is constant, cost to be minimized.

A DDoS attack is a typical example for the first case, where the cost is almost constant (several computers, some connections with zombies, and some handlers controlling them), and the profit, from attacker's viewpoint, is the extent of the degradation of QoS of the target system. A multi-stage attack scenario may help us to understand the second case, where the intended profit is almost ascertained while attacking cost is expected to be saved by attacker. In practice, it is hard to quantify intent and attacking cost finely due to numerous unspecific factors, so we have to gauge them indirectly via tangible measurements. For instance, in order to measure attacker's intent, we may turn to assess attack effects, e.g., the degradation of services provided by the system, deterioration of system performance with respect to high-level security properties, etc., and attacking cost can be measured in terms of the risk of being detected, resource used to launch attack, and so on. It is also worth noting that for the systems equipped with defense tools like IDS, the measures of attacker's intent and attacking cost is closely related to the defense mechanism, since the defense tools can prevent attacker from achieving his/her goals associated with system state transitions. In a well-planned, multiple-stage, carefully hidden (sometimes closely coordinated) attack, what the attackers concern is "how to achieve the goal with as few effort/cost as possible meanwhile avoiding to trigger the defender's awareness?" so attacker behavior is essentially a decision process with following properties,

Property 3.1: (**Goal-directed**) An attacker launches attacks with some particular incentives, or is directed to particular malicious goals.

Property 3.2: (**Action-dependence**) System state transitions completely depend on the actions set if the observations and attacker's incentive is predetermined.

Property 3.3: (**Reward-awareness**) During a multi-stage attack, an attacker is usually aware of the goal he/she has achieved and the cost must pay, determining his/her behavior and the consequence of an attack scheme.

B. Cost-sensitive Defender

While numerous defense mechanisms have come into use, none of them is a *silver bullet* ensuring perfect security without any negative consequence. For instance, an IDS has to tune the tradeoff between the detection accuracy and false alerts. We generally envision practical defense systems that can dynamically preserve the most desirable properties of a system in the presence of attacks. To do that, the tradeoffs between the failure cost due to attacks and the maintenance cost[1] due to defense/response must be tuned in fine-grained manners. This is an issue regarding system *survivability* [2], which can be formulated as an optimization problem with a objective function subject to particular cost constraints. Unfortunately, the precise calibration of relevant factors associated with cost analysis in security/risk assessment is always impossible, which is largely due to the intrinsic complexity of today's computer systems, also because of the system-specific configurations & security policies or human-biased specification. A more realistic way, from the point of view of defender, is to figure out those elements that are essential to the security rules, models and policies, and then leverage them in a generic cost-sensitive manner with the tradeoff concerns discussed previously. The prerequisite to achieve that is to specify the cost-sensitive defender behavior in formal ways.

A cost model was presented in [8] to describe cost-sensitive IDSs by considering the cost factors that are possibly involved in an intrusion detection process. It is assumed that most automated security polices built upon real-time IDSs are ad-hoc, typically relying on individual reports at a certain time, and a multiple phase attack is usually broken into pieces so that each detectable atomic attack can be calibrated with cost models. In practice, however, defenders are not only limited to automated responses to thwart attacks, but also include manual investigation, management and adjustment of security policies, as well as actual measures related to security posture or recovery. For multi-stage coordinated attacks that we are particularly concerned, it is desirable that cost-sensitive preemptive actions are taken prudently to mitigate ongoing attacks based on the estimates of system state and the anticipated consequence of response. Similar to the attacker behavior, defender behavior can also be regarded as a dynamic and strategic decision-making process: "what action should be taken under what system state with respect to observations".

Formally, we define $Cost_f$ and $Cost_m$ as system maintenance cost and failure cost, measuring the behavioral effects of defender and attacker respectively. It is obvious that their quantification requires a good knowledge of the target system, and a deep understanding on attack taxonomies and appropriate countermeasures. For instance, with the assistance of the tools for risk/threat assessment, some evaluation scores can be assigned to the system components/services to measure their significance of role in the whole system, or *criticality*, which is denoted as D_s; similarly, attack variants and security measures can also be assigned values to specify their threat degree and effectiveness, denoted as D_a and D_d respectively. Thus, the maintenance cost and failure cost can be calculated as, $Cost_m$=$D_s \times D_d$, $Cost_f$=$D_s \times D_a$. Although the metrics are site- and human-specific, they may serve as an evaluation basis for quantifying the relevant costs and balancing their importance.

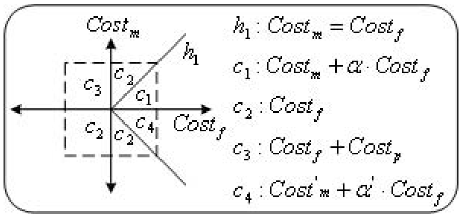

Fig. 1. Cost model of defender

As we mainly consider multi-stage attacks that includes a sequence of stages associated with different system states, an appropriate cost model needs to be built to depict its special characteristics. Following but abusing the result in [8], we assume the response of a defender at a particular stage falls into five categories: $\mathcal{C}_1$, "Alert" for a true attack; $\mathcal{C}_2$, "Alert" for a normal behavior; $\mathcal{C}_3$, "Silence" for an attack; $\mathcal{C}_4$, "Silence" for a normal operation and possibly; $\mathcal{C}_5$, an attack is misdiagnosed as another attack variant. A case-specific cost definition is shown in Fig. 1 ($\mathcal{C}_4$ is replaced by $\mathcal{C}_5$ since it is always 0) in an order of quadrants, where $Cost_p$ is penalty cost due to false alert, and $Cost_m^{'}$ is maintenance cost due to wrong attack variant detection. More specifically, as shown in the first quadrant, the defender takes no action (the cost is c_2) if $Cost_f < Cost_m$; otherwise, the defender takes an appropriate action against the attack with the cost c_1, i.e., $Cost_m$+α·$Cost_f$, where $\alpha \in [0, 1]$ is a weight denoting the threat degree of ongoing attack. Another weighting parameter $\beta \in [0, 1]$ can be introduced here to balance the tradeoff between the failure cost and maintenance cost, i.e., $\beta Cost_m + \alpha(1 - \beta) Cost_f$. As a matter of fact, a multi-stage attack always keeps advancing until it is detected and thwarted by the defender. Since the cost of each stage during attack must fall into the cases in Fig. 1, the cumulative cost of an attack scenario can be simply calculated as a sum of costs of independent system states under attack,

$$\begin{cases} Cost_s = \sum_{i=0}^{n} \{[\beta_i \cdot Cost_m(s_i) + \alpha_i(1 - \beta_i) Cost_f(s_n)]\gamma_0 \\ +[Cost_p(s_i) + Cost_m(s_i')]\gamma_1 + [Cost_m(s_i') + \alpha_i' \cdot Cost_f(s_n)]\gamma_2\} \\ \sum_{k=0}^{2} \gamma_k = 1, \alpha_{i-1} \leq \alpha_i, 0 < i \leq n \end{cases} \tag{1}$$

where $\alpha_i, \alpha_i', \beta_i$ have the same definitions as before. γ_i=1 if the corresponding case occurs, else $\gamma_i = 0$; $s_0 \cdots s_n$ is a set of system states that may occur in an attack, and s' is a misdiagnosed state. In addition, $\alpha_i Cost_f(s_n)$ can be replaced by $Cost_f(s_i)$ if the weight α_i of a particular state s_i cannot be determined. We also observe that in the cumulative cost function penalty cost $Cost_p(s_i)$ of false alert is actually triggered in the face of normal seeming operation (normal s_i is regarded as attack state for stealthy attacks), so the latter two terms in Eq. (1) can be combined together,

$$\begin{aligned} Cost_s = \sum_{s_i, s_i' \in \grave{S}} & [\beta_i \cdot Cost_m(s_i) + \alpha_i(1 - \beta_i) Cost_f(s_n)]\gamma_0 \\ & +[Cost_m(s_i') + Cost_p(s_i')]\gamma_1 \end{aligned} \tag{2}$$

where $Cost_p$ is penalty cost due to false alert and misdiagnose.

IV. Behavior-Based Model Formulation

This section presents a two-sided analytical model to characterize behaviors of attacker and defender by incorporating the elements that we discussed. It is assumed that both attacker and defender make their decisions at each system state with their own concerns, and they always intend to seek optimal strategies to maximize their objectives as the system state evolves over an extended period of time. However, in practice, system uncertainty and non-stationarity cause the system evolution and emitted observations to be complicated and intractable, which deems that they are not well-posed optimization problems. Therefore, stochastic decision formulations, which seek probabilistic rather deterministic strategies, is a more suitable optimization method to meet our realistic objective [7], [21].

A challenge is that the system states can not be directly observed, and they can only be estimated through relevant observations. For instance, an attacker may observe victims with a particular vulnerability in a network by *probing*, knowing his/her status of access privileges and so on. But the transitions towards compromised system states are only partially determined by the combination of the observations; a defender can estimate the system states on the basis of reports of security mechanisms, and maintains the states at certain security scales. Furthermore, the decision process of attacker and defender is obviously a Markov decision process, since the next system state is dependent solely upon the current state and action. A Partially Observable Markov Decision Process (POMDP) [1] is

[1] We use failure cost here to integrate a large variety of failure or degradation of system security associated with availability, integrity, confidentiality, etc.; while maintenance cost can be viewed as a concept unifying costs during the execution of response mechanisms, either automated or manual, in terms of time, computational overhead, and even negative effects due to over stringent or wrong response.

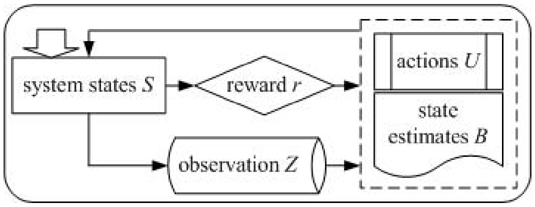

Fig. 2. A generic behavioral model

therefore formulated to characterize both attacker and defender's behavior. Moreover, if a group of heterogenous attackers/attack parties participate in an attack simultaneously, their cooperative behaviors can be further formulated as multi-agent POMDP, or MPO-MDP.

A generic behavior model of attacker and defender is shown in Fig. 2, where (a) both defender and attacker are taken as "peer" of the system, that is, the whole system contains three parts: *service part* includes all and only the components providing services to normal users, *defense part* includes the defense mechanisms, and *attack part* includes the set of attack strategies; (b) both attacker and defender achieve their respective objectives via appropriate actions based on the system states estimated by partial observations; (c) it is attacker, defender, and normal user who collectively drive the transitions of system states, but our model only considers explicitly the states driven by attacker and defender (we generally call them operator) regardless of those normal states driven by legitimate users. From a high-level perspective, this model can serve as a basis for probabilistic evaluation of system-level security, which not only encompasses representations of attackers, defenders in terms of their objectives and aggregated action effects, but also contains the abstractions of system services associated with workloads, resources/assets, privileges, and applications. In addition, on the basis of the model, the behaviors of attackers and defenders in a form of action sequences are also represented as decision-making processes with temporal considerations. More formally, the behavior-based model is structurally characterized by four key elements, or $\mathcal{M}$=$\{S, U, Z, R\}$,

- $S = \{s_1, s_2, ..., s_n\}$ is a finite state space of n distinct system states, containing subset of normal states $\bar{S}$ and attack sates $\grave{S}$,
- $U = \{u_1, u_2, ..., u_m\}$ a control space of m distinct actions (or responses) that are available to the operator,
- $Z = \{z_1, z_2, ..., z_q\}$ is an observation space of q distinct observations observed by operators and,
- a (possibly stochastic) reward $r(i) \in \mathbb{R}$ for state $s_i \in S$; or cost $c_{i,j}(u)$ for state transition from s_i to s_j with action u.

$\mathcal{M}$ describes the interaction between an operator g (either attacker or defender) and its operating environment, which includes a sequence of decision stages as follows,

1) In stage t, the system is in a particular state $s_t \in S$, along with observation $z_t \in Z$ emitted according to a probability distribution $\nu(z_t|s_t)$ over observation vectors, or $Pr(z_t|s_t, u_{t-1})$,
2) Operator takes action $u_t \in U$ in accordance with a randomized policy based on a probability distribution $\mu(b_t)$ over actions, with prior observed information with respect to *belief state*,
3) Actions u_t determines a stochastic matrix $Pr(u_t) = [p_{ij}(u_t)]$, where $p_{ij}(u_t)$ is the probability of making a transition from state s_i to state s_j under action u_t, or $Pr(s_j|s_i, u_t)$,
4) Operator receives a reward signal r_{t+1} after the state transition, which guides the operator to choose a policy so as to maximize the long-term average reward in terms of specific objectives.

As such, the parameters governing model construction can be organized into a family of action-dependent matrices: $m \cdot n \times n$ state transition probability $Pr\{s_j|s_i, u_i\}$ of matrices $\mathbf{H}$, $m \cdot n \times q$ observation probability $Pr\{z_i|s_i, u_{i-1}\}$ (or $\nu(s_i)$) of matrices $\mathbf{Q}$, $m \cdot n \times n$ transition cost (or action reward) matrices $\mathbf{R}$.

The decision process shows that at each stage an operator sees only the observations z_t and the reward r_t, while it has no knowledge about the underlying system states, how the actions affect the evolution of states, how the reward signal depends on the states, or even how the observations depend on the states. While the model is used for characterizing the general behaviors of defender and attacker, their specific constructions are different in essence and must depend on practical scenarios, which will be specified in Section V-B. A generic model [1] $\mathcal{M}$ always assumes an operator has access to a set of internal states which evolve as a function of the current observation and previous internal state, and then implements a parameterized policy mapping observation and internal state into probability distributions over the action set, thereby allowing both the action and internal state transitions to be stochastic. We assume $\mathcal{B}_t$ is internal system state at stage t, which is a $l-$length vector (i.e., $\mathcal{B}_t \in \{v \in [0,1]^l | \sum_{i=0}^{l-1} [v]_i = 1.0\}$, where l is the dimensionality of the system state space, $l = |\grave{S}|$ for attack state space, and $l = |\bar{S}|$ for normal state space or states related to defense), and the ith element $[v]_i = b_t(i) = Pr(s_i|\tau_t)$ (where τ_t denotes the previous observed information) representing the probability that the system is in state s_i at stage t given the observation up to t, which can be calculated by *Bayes theorem* as follows,

$$b_t(i) = \frac{\nu(z_{t-1}|s_i) \sum_{j \in l} b_{t-1}(j) Pr(s_i|s_j, u_{t-1})}{\sum_{z' \in Z} \nu(z'|s_i) \sum_{j \in l} b_{t-1}(j) Pr(s_i|s_j, u_{t-1})} \tag{3}$$

Intuitively, $b_t(i)$ represents the relative confidence that the true system state in stage t is s_i, that is, the greater the value of $b_t(i)$, the more likely that s_i is the true system state. The worst case is that $\mathcal{B}_t$ is a uniform distribution over l sates, which provides no knowledge to determine the true state. On the basis of state estimates, an operator may take an action $u_t^* = \mu(\mathcal{B}_t)$ to maximize the reward signal η. The long-term average reward signal of the overall control and the anticipated action can be represented using the following equations,

$$\begin{cases} \eta = \lim_{T \to \infty} \{\mathbb{E}_\mu [\frac{1}{T} \sum_{t=0}^{T} r_t | \mathbf{H},\mathbf{Q}, \mathcal{B}_t]\} \\ u^*(\mathcal{B}_t) = \arg\max_{u_t \in U} \{r_t + \sum_{s_j \in S} [\mathbf{H}_{ij}(u_t)] * \eta^*(s_j)\} \end{cases} \tag{4}$$

where T is the total number of stages, $\mathbb{E}_\mu[\cdot]$ denotes the expectation over all action traces with parameterized probability matrices in terms of observations, actions, and system states. While r_t can be specified for attacker and defender respectively as previous definitions, e.g., the reward signal of attacker is *reward=profit−cost* accumulating different goals at each attack stage, and that of defender can be defined as Eq. (2). So an optimal action u^* should be taken as the estimated states, where $[\mathbf{H}_{ij}(u_t)] * \eta^*(s_j)$ corresponds to the long-term average reward signal if s_i is in fact true, action u_t is taken, and optimal action continues for all future steps.

V. Behavior Inference for Security Evaluation

This section addresses how to guide construction of and provide parameter values for models $\mathcal{M}$, and how to practically infer and measure the behaviors. A prerequisite is that the appropriate level of detail/abstraction of the model should be determined based on the target system, the type and accuracy of parameter values thus can be obtained from the service, data, operator's prior knowledge, etc.

A. An Evaluation Framework

In general, three steps with the purpose of model construction and inference need to follow, as shown in Fig. 3.

Investigation on the target system. The operator collects and extracts the observations of interest by examining and understanding system properties in both hardware and services. For instance, an attacker may find exploits or vulnerable victims by port-probing and capture the traffic pattern and communication sessions via sniffer; a defender may use security scanners combined with security checkers and auditing tools to estimate model parameter values.

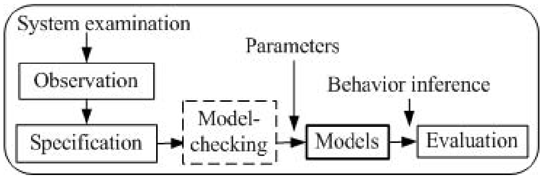

Fig. 3. Evaluation procedure

Model specification & algorithm selection. Specifying a set of model-based elements by definitions, engineering assumptions, and even experimental training, including state space S, observation space Z, action set U, reward signal r, and probability matrices $\mathbf{H}, \mathbf{Q}, \mathbf{R}$. Also, policy-learning algorithms such as dynamic programming, value iteration, policy iteration can be used to infer the practical behaviors, which is usually system- and scenario-specific.

Evaluation & analysis. As appropriate attacker models are critical to quantitative security evaluation, additional analysis are necessary to validate the consistency or accuracy between the inferred behavior and the actual actions that are practically taken. For an attacker, a set of actions with minimal cost can be approximately derived for a particular intent, meanwhile those observations exploited by different strategies can be also discovered for attack *reliability* analysis. On the other hand, with the constrained cost, the inferred behaviors allow us to derive the attacker's intents or the scale of his/her intended profit. The evaluation on the defender's behavior is usually based on several traditional security metrics, such as the trade-off between the detection accuracy and false alarm rate (for IDS), the trade-off between maintenance cost and failure cost, response latency etc.

While these three steps are closely related and all are essential to construct an integrated framework or to develop methodologies for probabilistic security validation, we focus on the second step here.

B. Elements Specification

System state is characterized by numerous features and properties drawn from various hardware components (e.g, hosts, communication links and topology in a network) and software services (e.g., applications, services, database, user privileges), any change of any element may cause state transition.

Assumption 1: The inherent continuous system states can be viewed as a sequence of discrete events, where $|S| \ll |\bar{S} \bigcup \grave{S}|$, and $|\grave{S}| \ll |\bar{S}|$, and in most cases, the states in $\bar{S}$ and $\grave{S}$ are interchangeable.

Observation is the variable that can be observed to characterize system states, such as audit events, privilege processes, CPU consumption, network traffic patterns. Observations to attackers can be vulnerabilities, access privileges, inter-host trust relationships, link connectivity, etc., and they can be also generated by attacker as action effects; defender may obtain observations via assisting tools, such as snort, port/protocol examiner, etc. The mapping of system states to observations is probabilistic (denoted as $\nu(z_i|s_i)$ in $\mathcal{M}$), and the more observations the more deterministic of a state estimate.

Assumption 2: At a certain time, the system state can be estimated via emitted observations, and generally $\nu_d(z_i|s_i) < \nu_a(z_i|s_i)$ (where $\nu_d(\cdot)$ is defender's estimate, $\nu_a(\cdot)$ is attacker's estimate). A POMDP turns into a MDP when $\nu_a(z_i|s_i) \approx 1$.

Actions are taken by defender and attacker to move the system to their desired states, and a sequence of actions is combined as a strategy or policy, dan the strategy space is usually bounded in practice. Further, attacker and defender rarely take actions simultaneously, and a long-run system states depend on the aggregated effects of their actions instead of individual ones.

Assumption 3: At a certain time t, if the underlying state is $s_t \in \grave{S}$, its associated attacker action should be u^a_{t-1} and defender action should be $u^d_{t+\Delta t}$ or null, and a number of attack states and normal states may occur during Δt.

Reward essentially measuring the actions costs and their effects, and guides the evolution of attacker and defender behaviors. It is the most important element in $\mathcal{M}$ and has been discussed in Section III.

Probability Matrices is about the probability distribution in a 3D space. The elements of those matrix families can be populated by *a prior* knowledge or experimental training. For attacker, the values depends on his/her knowledge, experience and even intuition, so the more sophisticated an attacker the more accurate of the values. Likewise, an experienced security analyst may set parameter values with a higher accuracy, and an extensive experimental training would be helpful to improve value accuracy.

Assumption 4: With appropriate initial parameter settings of $\mathcal{M}$, the model-based algorithms may eventually infer practical policies approximately for a particular attack-defense scenario, and the relative parameter values rather than absolute ones contribute more to the model construction.

VI. Simulation-based Case Study

The purpose of this section is to examine the feasibility and effectiveness of our proposed model using real trace data and to demonstrate how the behaviors of attacker and defender are inferred for probabilistic evaluation of system security in a practical scenario subject to DDoS attack.

A. Scenario Description and Model Construction

Dataset The experimental data was drawn from one of the MIT 2000 DARPA intrusion detection scenario specific data sets, LLDOS 1.0 [14], and slightly modified as demand. The original simulation network is divided into three segments as outside network, inside network, and DMZ of the organization AFB. IDS sensors are deployed for monitoring the entire network sessions and several particular hosts. The scenario contains a series of DDoS attacks launched by a novice attacker, including five phases in Table I. Also, since the main objective of this data set is to evaluate IDS, the defender was assumed to be naive, we can not infer the actual defender behavior coping with attacker neither their aggregated effects, but we can still observe the attacker behavior and anticipated defender behavior.

We examine system security in terms of availability, confidentiality, and integrity under DDoS attack. So from defender's perspective, we assume system states as $S = \{W, P, C, B\}$, where W represents the *normal* state, P denotes the state under probing and subject to the loss of *availability*, C denotes the state under exploitation and breach of both *integrity* and *availability*, and B denotes the state that has been compromised and no security attribute preserved. For the defender, the observations used for estimating system states are collected from IDS sensors, i.e., Z=$\{z^t_1, z^t_2, ..., z^t_q\}$ (t=$0, 1, ...$), where z^t_i denotes report of a sensor at time t. The observation set is updated at each time t and only q sensor reports are selected as observations, and the selection is based on the sensors' priority that is ranked in terms of *criticality* D_s of the host they are deployed. Two NIDS sensors and two host-based sensors were deployed in this experimental scenario. Also, since the defender is naive, the action set is small, i.e., $U = \{Silence, Alert\}$, plus the candidate countermeasures expected to be taken (as shown in Table I). **Network assets** We need assume network assets for quantifying the rewards of defender and attacker. While the network is consisted of three subnets, we only examine the hosts in DMZ and inside networks, and use D_s to measure the significance of those hosts. As shown in Table II, "loud" is a router in DMZ with value 20, "solomon" is DMZ sniffer with $D_s = 10$, and other hosts are assigned 1. In addition, "mill" and "locke" are DNS server and sniffer in inside network, their D_s are thus assigned to 50 and 10 respectively. Another vulnerable host "pascal" is set to 5, and other insides host are set to 2. Note these assumed values only help us to define the rewards, and they do not necessarily reflect the true network assets. Since the defender is native in this scenario, action cost D_d is then set to 0 for *silence*, 1 for *alert*, and the costs of those assumed countermeasures

TABLE I
FIVE PHASES IN THE ATTACK SCENARIO

Step	Attacks	D_a	Goals	Defender Countermeasures (Assumed)	D_d
1	IPsweep	1	Sending `ICMP echo-requests` for live hosts	Source address checking	2
2	Probe	2	Probe of live IP's to look for the `sadmind` daemon running on Solaris hosts	Strengthen awareness, Scrutinize port activity, ignore suspicious source IP	3,3 3
3	Break-in	5	Breakins via the `sadmind` vulnerability, both successful and unsuccessful on those hosts	block access port, sleep/kill exploited process, application or service	3,5
4	Install Virus	10	Installation of the trojan `mstream` DDoS software on three hosts using telnet	sleep/kill connection/installation utility, delete payload/directory	5,8 8
5	DDoS	20	Launching the DDoS attacks	halt operations, alert admin., recovery	10,10,20

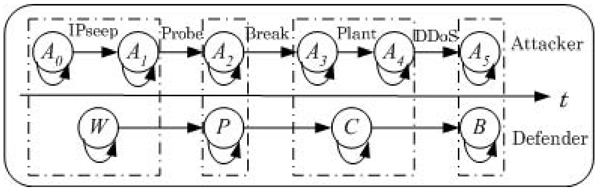

Fig. 4. Potential State transitions of attacker and defender

TABLE II
ASSUMED ASSETS IN DIFFERENT SUBNETS

	DMZ			Inside			
Hosts	loud	solomon	others	mill	pascal	locke	others
D_s	20.0	10.0	1.0	50.0	5.0	10.0	2.0

are shown in Table I. For attacker, D_a=1 if it accesses the network normally, otherwise D_a=2, 5, 10, 20 for the four actions respectively. With the assumption, defender's reward function Eq. (2) is thus dominated by $Cost_f$ as $Cost_m$ is always 0. For attacker reward, we calculate it as $Reward = GrossProfit - Cost$, where we let $GrossProfit$=$D_s \times D_a$ and $cost_T$=$\sum_{t=0}^{T} \xi_t D_s$. In this scenario, the value of ξ_t does not mean much on the evolution of attacker reward, so we set it 0 for an easier analysis.

Parameter setting We must also specify matrices $\mathbf{H}$ and $\mathbf{Q}$ for attacker and defender respectively. In practice, many of the transition probabilities in $\mathbf{H}_d$, and all of the measurement distributions& observation probabilities in $\mathbf{Q}_d$ can be empirically estimated from the training data set with the assistance of analytical methods [9]. However, there is no such a compelling need to accurately determine $\mathbf{H}_d$ and $\mathbf{Q}_d$ here, since our assumed countermeasures u_i have no actual effects on state transitions when $alerts$ were triggered. For elements in $\mathbf{H}_d$, $Pr(s_j|s_i, u_i)$, the only nonzero transition probabilities during attack operation in this scenario are for self-transitions in all states, as well as the transitions between successive steps of the attack scheme, resulting in the final state B. The potential state transitions of attacker and defender is shown in Fig. 4. If defender really takes actions, the transitions from $\{C, P, B\}$ to W may occur. Since there is no similar training data available for empirically approximate state transition probabilities, we take the same data set as an example for illustration. The left portion of Fig. 5 shows the state transitions by simply counting the number of occurrences of all events represented by each conditional distribution. A transition matrices under active countermeasures can also be estimated empirically using training data. For the elements in $\mathbf{Q}_d$, $Pr(z_i|s_i, u_{i-1})$ turns into $Pr(z_i|s_i)$, which can be simply regarded as a direct mapping between state s_i and observation z_i (IDS sensor report). Among the elements, $Pr(z_i = alert|W)$ represents the false positive rate, and $Pr(z_i = silence|\neg W)$ represents false negative rate. Obviously, a good estimate on $Pr(z_i = alert|W)$ may avoid wrong actions caused by false alarms in real time defense.

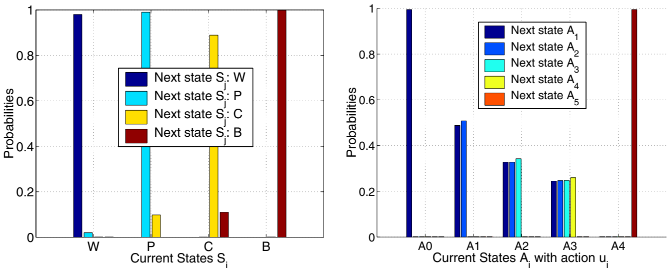

Fig. 5. State Transition Probabilities of Defender (left) and attacker (right)

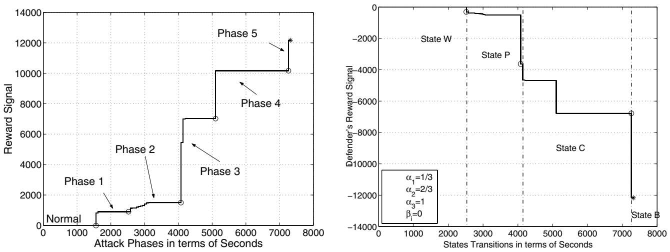

Fig. 6. Reward evolution during attack

Similarly, in order to determine $\mathbf{H}_a$ and $\mathbf{Q}_a$, we may examine attack phases required for a successful DDoS attack from attacker's perspective. Note DDoS attack in this scenario has five phases, while the system states estimated by the defender only have four, the mapping between them therefore is $A_0, A_1 \in W, A_2 \in P, A_3, A_4 \in C$, and $A_5 \in B$, where A_i represents ith attack phase. A general form for the element in $\mathbf{H}_a$ is,

$$Pr(s_{t+1}|s_t, u_t) = Pr(A_j|A_i, u_i) = \begin{cases} p_t, i <= j \\ 1 - p_t, i > j \end{cases} \quad (5)$$

where t denotes attack stage. In this scenario, while the attack was launched using a simple DDoS tool, `Mstream DDoS`, the network was equipped with poor security measures, so p_t should be large. In particular, we can estimate p_t using the data set at each attack phase in this scenario by replaying the traffic via Netpoke [15]. The right part of Fig. 5 shows the empirical transition probabilities $Pr(A_{i+1}|A_i, u_i)$ $(i \in [0, 4])$ of the attacker. Note at each state the attacker always knows the exact action that should be taken for the next state transition, thus U is a unit matrix here. That is, at stage i, the value of ith element of u_i close to 1, the others are nearly 0.

B. Rewards Evolution and Their Implications

As the original experimental data has been processed with the NIDS sensors to generate alerts in IDMEF format, our simulation was essentially conducted with the labeling data. The sensor data was generated from both network tcpdump (DMZ and inside networks) and Solaris (2.7) BSM Audit Data (host "mill" and "pascal"). For easier observation and calculation, we measured each attack phase in terms of seconds. The original data were collected over nearly 3 hours (from 09:25 AM to 12:35 PM). For example, probing started at 10:07:07 and ended at 10:17:10, lasting nearly 600 seconds.

The left portion of Fig. 6 shows attacker's reward signal during the attack, which increased monotonously. It is not surprising since the attacker was not countermined and achieved goals step-by-step. We also observed that the biggest signal increase occurred between attack phase 2 and 3 with 263.3% (from 1500 sec. to 5450 sec.), which was the key step of the whole attack. Since attacker has already achieved his final goal, the reward at phase 5 is neglected here. In addition, we may see that phase 4 has a very smooth reward update, this is because the attack sessions during this period was collected in a very short time (22 inside network sessions and 19 DMZ sessions appeared in one second) while the attack effects were measured in terms seconds.

We also observed the defender behavior during attack. In particular, we examined the intrusion alerts collected by NIDS sensors in DMZ and inside network (two sensors in 'Mill' and 'Pascal' were not considered), and we utilized information ”Time, SessionDuration” and ”Address” regardless of specific protocols or ports. We need maintenance cost $Cost_m = 0$ and failure cost $Cost_f$ to calculate the reward. Since defender kept inactive, $Cost_m$=0, while $Cost_f$ depends on D_s that is determined by the destination address of intrusion alerts. Referring to Eq. (1), we set $\alpha = (0, 1/3, 2/3, 1)$ to represent the attack progress at states W, P, C, B respectively, and $\beta_i = 0$ means no response was taken. Moreover, since there was no false positive appeared the original alerts data, the misdiagnosis case never occurred, i.e., $Cost_p = 0$. Thus Eq. (1) is simplified as $Cost_s = \sum_{i=0}^{n} \alpha_i Cost_f(s_i)$. With the parameters, the right part of Fig. 6 shows the update of defender reward, which is nearly opposite to that of attacker. Obviously, the most significant changing of reward occurred at state C (A_3 and A_4 for attacker), decreasing from -3633.3 to -6780.0. The state B also has a large negative reward -12170.0, which means that the network has been completely compromised at this time. The reward decreased dramatically from the star point as time advances. We observe that one promising characteristic of the model is that it treats the network as a whole and measures the security in terms of the consequences of attacks associated with assets inside the network by abstracting and representing the inter-hosts dependencies as high-level states.

C. Policy Inferences in an Attack-Defense Scenarios

In a more practical attack-defense scenario, both defender and attacker will choose optimal policies to maximize their rewards. However, the original experimental data only allows us to observe attacker behavior, we have to assume several countermeasures (Table I) to enrich our simulation. While each IDS sensor report may serve as observation for estimating system state of a particular host, we are more concerned about the whole network states, and thus handle the multiple sensors as a whole. Formally, we assume there are total N NIDS sensors, and each alert falls into M categories, then the entire observation should be a matrix $N \times M$ for each network state. In our scenario, there are total four sensors deployed in the network (two are network-based, two are host-based), and we assume their alerts fall into five categories: Silence, Scan, In-Host, Compromised, and Outbound. Each element in $\mathbf{Q}$ thus can be observed via *a prior* knowledge or a training phase. For example, for network state W, the corresponding observation with naive defense measure can be initiated as,

$$Z_w^{u_d=\varnothing} = \begin{pmatrix} 0.88 & 0.087 & 0.00 & 0.001 & 0.032 \\ 0.76 & 0.1995 & 0.00 & 0.0005 & 0.040 \\ 0.9799 & 0.00 & 0.009 & 0.0001 & 0.011 \\ 0.9927 & 0.00 & 0.004 & 0.0003 & 0.003 \end{pmatrix}$$

where the fist column represents the detection accuracy, and the sum of remaining elements in each row denotes the false positive rate. Similarly, $Z_P^{u_d}, Z_C^{u_d}$ and $z_B^{u_d}$ can be also estimated under different defense mechanisms u_d. Note that different defense postures may cause different observations, and a large number of IDS sensors may produce overwhelming alerts to process. To handle this, we introduce a time window, called *Report Update Cycle* (RUC for short), to periodically maintain and update the observation vector.

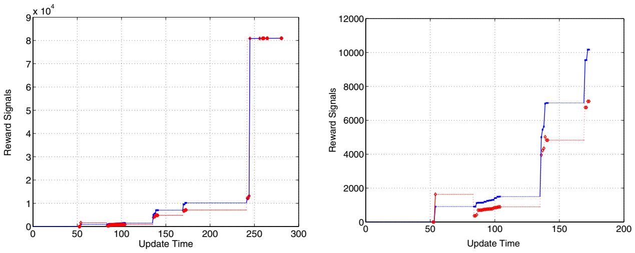

Fig. 7. Reward evolution under actual actions

In addition, we generalize the assumed actions into three categories: *passive, neutral*, and *positive*, which are denoted as u_P, u_N, u_A respectively. As Eq. (2) shows, the costs $Cost_s$ associated with defender policy is essentially dominated by the parameter β_i balancing maintenance cost and failure cost. Therefore, the rewards $\mathbf{R}_{ij}(u_i) = -Cost_s$, and β_i can be simply defined as follows,

$$\beta_i = \begin{cases} \beta_0 = 0.8, & s_i = W, u_i \in \{u_P, u_N\} \\ \beta_1 = 0.2, & s_i \in \{P, C\}, u_i \in \{u_P, u_N\} \\ \beta_2 = 0.95 & s_i \neq B \text{ and } u_i \in u_A \\ \beta_3 = 0.05, & s_i = B \text{ and } u_i \notin u_A \\ \beta_4 = 0.5, & \text{otherwise} \end{cases} \tag{6}$$

A reward function can be defined as well to correlate $\mathbf{R}_{ij}(u_i)$ and actions directly, while it is not a compelling need to do so in this scenario considering the high-accurate sensor alerts. We see β_0 and β_1 is a pair used to balance the maintenance cost and failure cost due to inappropriate and appropriate actions in $\{u_P, u_N\}$; β_2 and β_3 is another pair balancing the costs associated with inappropriate and appropriate actions in u_A.

The defender's policy therefore can be inferred with model-based parameter values **H, Q, R**, and the controlling parameters α, β, RUC. Provided the exact same sensor report, Fig. 7 depicts the defender action with the anticipated policy. That is, the defender would take actions with the original data that was generated under “naive” defending policy, while the actions have no actual effect on the observations. We set $(\alpha_0, \alpha_1, \alpha_2, \alpha_3) = (0, 1/3, 2/3, 1)$, $URC = 30$ sec., while **H, Q** keep unchanged.

Due to the high-accurate reports of IDS sensor, system sates are estimated perfectly by the observations, making the results be much similar to those generated by a “heuristic” policy. The comparison between Fig. 6 (right) and Fig. 7 (the left part is an entire view and the right part zooms in the former four attack phases) shows that an anticipated policy has much potential to enhance system survivability,

- When the attacker initially launched the attack, the defender did not take any action even though IDS sensor triggered alert. A simple reason is that the maintenance cost is much greater than failure cost. For example, at update time 54, the failure cost was 906, and the overall cost $Cost_s$ of defender was 1630.8.
- When attack advanced to A_2 when RUC=84, the defender took actions in u_p to mitigate the attack based on the high-confident system estimate $\mathcal{B}_t$. If the action had actual effect, the attack would have been successfully thwarted. Note that since countermeasures in u_P and u_n have different cost D_d, the action candidate were not taken immediately while they were concerned as far as future decision-making and should be able to minimize the long-term rewards. For example, defender took actions `ignore suspicious source IP` or `Scrutinize port activity` instead of `kill exploited`

process/service for the sake of minimizing defender's long-term reward.

- When the system evolved into state C, the significant update of reward signal required the defender to not only initiate kill connection, but also quickly diagnose the system by taking action delete payload. Only by taking such actions the increase of reward signal can be stopped. In state B, the defender would have definitely taken recovery actions no matter how much the maintenance cost, since the failure cost is much larger.

VII. Concluding Remarks and Future work

In this paper, we applied a variant of MRM to characterize and infer the behaviors of defender and attacker for probabilistic security evaluation, which also aims at bridging the gap between system dependability evaluation and security evaluation. Two objective-oriented models were developed respectively to unify the attacker's various intents and costs during the particular attack strategies, and to capture the defender's cost-sensitive concerns by quantifying different cases in adopting countermeasures. The objectives, actions and their aggregated effects were then integrated using a Partially Observable Markov Decision Process (POMDP) for fine-grained characterization of the behavior of defender and attacker, as well as their practical inference. We conducted a simulation-based experiment using real trace data to validate our methodology. In particular, we show how to determine model-based parameters, and how to evaluate system security on the basis of action effects of attacker and defender, as well as the inferences of their optimal action policies.

Nevertheless, our work is still preliminary and needs more exploration. These open issues are taken as future work and will be done along the line as follows: (1) we would investigate better objective-oriented models which provide more accurate quantification of attacker intents, costs, and actions (in more coordinated manner among multiple attacker parties), as well as the defender's cost-sensitive consequential actions in a complete multi-stage attack scenario; (2) we would design more practical methods to guide construction of and provide input parameter values (including those model-based elements S, Z, U and probability transitions **H, Q, R**) for models, and abstract them from case-specific to more generic which may still require details for meaningful representation; (3) we would investigate more efficient algorithms and methods to validate the inferred behaviors of attacker and defender, including sensitive and accuracy analysis, and use some policy iteration methods to optimize those behavior to enhance their scalability in more complicated systems and more sophisticated attack scenarios.

References

[1] Douglas Aberdeen, "A Survey of Approximate Methods for Solving Partially Observable Markov Decision Processes", *National ICT Australia Report,* Canberra, Australia, 2003.

[2] A. Avizienis, J. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", *IEEE Trans. on Dependable and Secure Computing*, Vol.1, No.1, Jan.-Mar. 2004.

[3] L. Cloth, J.-P. Katoen, M. Khattri, and R. Pulungan, "Model Checking Markov Reward Models with Impulse Rewards," In *Proc. Int'l Conf. Dependable Systems and Networks(DSN2005)*, pp.722-731, 2005.

[4] S. Derisavi, P. Kemper, and W. H. Sanders, "Lumping Matrix Diagram Representations of Markov Models", In *Proc. Int'l Conf. Dependable Systems and Networks(DSN2005)*, pp.742-751, 2005.

[5] J. B. Dugan, and M. R. Lyu, "Dependability Modeling for Fault-Tolerant Software and systems", *Software Fault Tolerance,* M.R.Lyu, ed., Chichester:John Wiley & Sons, pp. 109-138, 1995.

[6] E. Jonsson, and T. Olovsson, "A Quantitative Model of the Security Intrusion Process Based on Attacker Behavior," *IEEE Trans. on Sofware Engineering,* vol. 23, no. 4, pp. 235-245, Apr. 1997.

[7] O. P. Kreidl, and T. M. Frazier, "Feedback Control Applied to Survivability: A Host-Based Autonomic Defense System," *IEEE Trans. on Reliability,* Vol.53, No.1, pp.148-166, 2004.

[8] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok, "Toward Cost-Sensitive Modeling for Intrusion Detection and Response", *Journal of Computer Security*, Vol.10, Issue 1-2 (2002), pp. 5-22.

[9] W. Lee, and D. Xiang, "Information-Theoretic Measures for Anomaly Detection," In *Proc. of the IEEE Symposium on Security and Privacy (S&P'01),* Oakland, CA, May 2001.

[10] B. Littlewood, S. Brocklehurst, N. Fenton, et al., " Towards Operational Measures of Computer Security," *Journal of Computer Security*, vol. 2, pp. 211-229, 1993.

[11] P. Liu, W. Zang, and M. Yu, "Incentive-Based Modeling and Inference of Attacker Intent, Objectives, and Strategies", *ACM Trans. on Information and System Security,* Vol. 8, No.1, Feb. 2005, Pages 78-118.

[12] K. Lye, and J. M. Wing, "Game Strategies in network security", *International Journal of Information Security*, (2005) 4: 71-86.

[13] B. Madan, K. G. Popstojanova, K. Vaidyanathan, and K. Trivedi, "Modeling and Quantification of Security Attributes of Software Systems," In *Proc. Int'l Conf. Dependable Systems and Networks(DSN 2002)*, pp.505-514.

[14] MIT LLDDoS 1.0, http://www.ll.mit.edu/IST/ideval/data/2000 /LLS_DDOS_1.0.html

[15] MIT Lincoln Lab, http://www.ll.mit.edu/IST/ideval/tools/tools_index.html

[16] D. M. Nicol, W. H. Sanders, and K. S. Trivedi, "Model-Based Evaluation: From Dependability to Security," *IEEE Trans. on Dependable and Secure Computing*, Vol.1, No.1, pp.48-65, January-March 2004.

[17] R. W. Ritchey, P. Ammann, "Using Model Checking to Analyze Network Vulnerabilities," In *Proc. of IEEE Symposium on Security and Privacy (S&P'00)*, pp.156-165, May 14-17, 2000.

[18] S. M. Ross, *Introduction to Probability Models,* 8th Edition, Academic Press, Elsevier Science, 2002.

[19] S. Singh, M. Cukier, and W. H. Sanders, "Probabilistic Validation of an Intrusion-Tolerant Replication Systems," In *Proc. Int'l Conf. Dependable Systems and Networks(DSN2003)*, pp. 616-624, June 2003.

[20] N. Ye, S. M. Emran, Q. Chen, and S. Vilber, "Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection," *IEEE Trans. on Computers,* Vol. 51, No. 7, July 2002.

[21] Z. Zhang, and H. Shen, "Constructing Multi-Layered Boundary to Defend Against Intrusive Anomalies: An Autonomic Detection Coordinator," In *Proc. Int'l Conf. Dependable Systems and Networks(DSN 2005)*, pp.118-127, June 2005.