Efficient Defense Strategies to Minimize Attackers' Success Probabilities in Honeynet

Frank Yeong-Sung Lin
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
yslin@im.ntu.edu.tw

Yu-Shun Wang
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
d98002@im.ntu.edu.tw

Po-Hao Tsang
Department of Information Management
National Taiwan University
Taipei, Taiwan, R.O.C.
d91002@im.ntu.edu.tw

***Abstract*—In this paper, we consider the problem of minimizing attackers' success probability in a protected network subject to attacker profile/behavior constraints and defender resource/strategy constraints. Compared with previous research, the following two enhancements are made. First, we no longer assume that perfect knowledge regarding the network topology and defense resource allocation is fully available for attackers (a worst case scenario for the defender). Second, all combinations of attacker classes can be considered, where each attacker class may be associated with any number of attributes, including ratio, intelligence/experience level, available attack resource and sophisticated attack strategies. The problem is modeled as a generic mathematical programming problem, and a novel two-phase solution approach, which well combines mathematical programming and simulation techniques, is proposed. More specifically, in the "Evaluation Phase", efficient and effective simulations are conducted to evaluate the effectiveness of the current defense policy; whereas, in the "Defense Policy Enhancement Phase", specially-proposed and easy-to-collect information from the "Objective Function Evaluation Phase" is adopted to calculate gradients of the decision variables. From computational experiments on honeynet, applicability and effectiveness of the proposed framework and algorithm are clearly demonstrated.**

Keywords-Network Attack and Defense; Network Survivability; Honeypots; Imperfect Knowledge

I. INTRODUCTION

The rapid growth of the Internet has given rise to a wide variety of applications that make our lives easier. However, people's increasing dependence on the Internet allows cyber criminals to launch denial of service (DoS) attacks against systems with the highest connectivity. Moreover, by compromising the most important server, attackers can gather highly sensitive information.

According to the 2008 CSI Computer Crime and Security Survey [1], there is a distinct tendency for cyber criminals to focus numerous attacks on one potential victim. This kind of attack behavior is called a *target attack*. After attackers have compromised a target, the attacked organization may suffer financial losses and/or damage to its reputation. For example, from a monetary point of view, the CSI survey [1] noted that "the most expensive kind of incident on average was financial fraud, with an average reported cost of $463,100, followed by dealing with "bot" computers within the organization's network, reported to cost an average of $345,600 per respondent. Given the number of threats that an organization may encounter, our network cannot prevent all types of malicious attacks. Therefore, we need to define a metric to evaluate the network's performance under malicious attacks. Survivability is often used to measure a system's performance. The concept has been widely discussed in the literature, e.g., [2] [3] [4] [5] [6].

Although many approaches attempt to improve system survivability, defenders are still in a passive position. To address this problem, in this paper, we consider general defense resources (e.g. firewalls, IDS, and IPS), as well as another kind of defense technology called "honeypots", which are designed to deceive and distract attackers. This security tool leads attackers to believe they have successfully compromised the server, even the core node; however, in reality, they have wasted their attack resources and fallen into a trap set by defender. The concept of honeypot has been in continuous progressing, many different taxonomies and applications are gradually appeared [7] [8] [9] [10].

Recent studies also agree the feasibility of honeynet which is composed by many honeypots applying in real systems. For example, in [10], Dimitriadis first analyze the vulnerabilities of 3G operator's architecture. By applying the game theory, the author discovers both a 3G operator and his roaming partners will get benefit from deploying honeynet.

Besides, there are different perfectives on researching attack defense scenario. The cost-based concept gets more and more attention [2] [3] [4] [11]. For example, in [3], the authors assume attacker has to spend some "cost" to compromise one node in order to decrease the performance of targeted network. In this paper, we also adopt the concept of cost-based attack defense scenario and propose a two-phase solution approach to discover the defense resource allocation which minimizes attackers' success probabilities. As for [11], the definition of vulnerability authors adopted is transformed from [12], which is $\frac{T^m}{T^m + t^m}$. In attack defense scenario, T stands for total attack budget and t refers to total defense budget. According to m, it represents attacker–defender contest intensity which □ 0. In case of m = 0, the investments t and T have equal impact on the vulnerability. While $0 < m < 1$, there is a disproportional advantage of investing less than one's opponent. If $m = \infty$, it gives a step function where "winner-takes-all". Once an attacker invests his budget that equal or

slightly more than the defense budget on one node, it is compromised. In this research, we adopt the concept which similar to $m = \infty$ case. In other words, an attacker at least requires the same budget as the defender invested on one node to compromise it. The reason we choose this case is because once attackers found a vulnerability of the system, they get over-whelming advantage on compromising this target. This forms a winners-take-all situation which is similar to $m = \infty$ case.

For attackers, we categorize based on their attack budgets, capabilities and next hop selection criteria. An attacker's budget and capability are divided into three levels: high, medium and low. In this paper, budget comprises time, money, and human force. Based the concepts discussed in [13], we organize next hop selection criteria as three main strategies.

To the best of our knowledge, all previous works, e.g., [2] [3] [4] [5] [6], assume that an attacker has perfect knowledge or detailed information about the topology of target network and there is only one kind of attacker category. This helps the defender evaluate the worst case scenario. In real attack-defense situations, however, attackers usually have partial information; in other words, they only have imperfect knowledge. Therefore, attackers can only gather next hop information during attack. Furthermore, the system is attacked simultaneously. By taking these concepts into consideration, a defender can analyze the average case scenario.

Of course, adopting the imperfect knowledge assumption means that we cannot apply the solution approaches proposed in [2] [3] [4] [5] [6] to solve security problems. Therefore, based on the imperfect knowledge property, we develop a two-phase evaluation process which combines mathematical programming and simulation techniques to solve such problems.

It is worth to emphasis there is a great difference between perfect knowledge and imperfect knowledge. For example, most of shortest path algorithms and minimum cost spanning tree algorithms are based on the perfect knowledge assumption since we know all the nodes and links at first. If we relax the assumption on these algorithms, i.e., nodes and links will dynamically appear during searching for the optimal solution, well-known algorithms may not be feasible anymore. Although it is not required to relax this assumption in the shortest path algorithms and the minimum cost spanning tree algorithms, it is a necessary concern in our attack defense scenario.

From another viewpoint, the perfect knowledge case can be classified into the worst case scenario where attackers know all detail information about the target network. As to imperfect knowledge case, it can be categorized into the average case. Similar to analyze the complexity of an algorithm, the average case is more difficult than the worst case. Therefore, we propose a novel methodology to address this kind of problems in network attack and defense scenarios.

TABLE 1. COMPARISOM OF PERFECT KNOWLEDGE AND IMPERFECT KNOWLEDGE

Perfect Knowledge	Imperfect Knowledge
Complete information about topology	Only one hop information
Complete information about defense resource allocation	Only next hop defense resource information
Complete information about node attribute	Partial information about node attribute
Single category of attackers	Multiple categories of attackers
Information is gathered before an attacker launches an attack	Information is gathered during attack

Table 1 shows the comparison of perfect knowledge and imperfect knowledge. As long as one of the conditions mentioned in the left-hand side of above table is not satisfied, we can say it is an imperfect knowledge case. In this paper, we break three perfect knowledge assumptions at the same time. In other words, attackers only have one hop topological and defense resource allocation information. Even for the node attribute, attackers are not fully mastered. They require gathering information during attack.

II. PROBLEM FORMULATION

In this section, we describe the problem, propose a mathematical model of the attack/defense scenario with specific assumptions, and define the objective.

Each attacker's objective is to compromise the core node in the given network. The defender has perfect knowledge of the network that is targeted by several attackers with different budgets, capabilities, and next hop selection criteria. However, the attackers are not aware that the defender has deployed honeypots in the network; in other words, their knowledge of the network is imperfect. In addition, a node can only be attacked if a path exists from the attacker's position to that node, and all the intermediate nodes on the path have been compromised. A node is deemed to be compromised when the attack resources allocated to it are not less than the defense resources allocated by the defender. Furthermore, only malicious nodal attacks are considered.

As mentioned earlier, we classify attackers based on their attack budgets, capabilities and next hop selection criteria. The budget and capability components are divided into high, medium and low levels. For next hop selection criteria, following [13], we assume that attackers will choose the current node's neighbor that has the highest/lowest defense level; otherwise, they just select a candidate to attack at random. It may seem odd that an attacker would target the node with the highest defense level. However, since such nodes are more likely to contain highly sensitive information, attackers who want to steal valuable information will target them. Note that, even if two attackers belong to the same category, we cannot be sure that the attacks will achieve the same result. For example, one attacker may be distracted by a honeypot, while the other may successfully avoid being tricked by this security tool. For presentation purposes, we set the total number of attacker categories as 27. No doubt, we could apply more sophisticated classification mechanisms to categorize attackers.

We assume that each attacker only has information about the next hop, including the neighbors of the current node and their defense levels; hence, to compromise the core node, the attacker needs to gather information step-by-step. We model the above problem as a mathematical programming problem. The given parameters are defined in Table 2.

TABLE 2. GIVEN PARAMETERS

Notation	Description
M	The total evaluation frequency for all attacker categories
K	The total number of attacker categories
R_k	Rounded evaluation frequency of each type of attacker (where $k \in K$)
P_k	The proportion of the total number of attackers that are type k attackers (where $k \in K$)
D	All possible defense strategies
$\overline{A_k}$	An attacker's strategy, which is comprised of his budget, capabilities, and next hop selection criteria (where $k \in K$).
$S_{kj}(\overline{D}, \overline{A_k})$	1 if attacker j in the k^{th} attacker category can compromise the core node under $\overline{D}$ defense strategy, and 0 otherwise (where $k \in K$)
B	The defender's total budget
B_k	The total budget of the k^{th} type of attacker, where $k \in K$
F	The index set of honeypots that act as fake core nodes
I	The index set of all general nodes in the network

In the formulation we proposed, the attack category is denoted by K. The decision variables are defined in Table 3.

TABLE 3. DECISION VARIABLES

Notation	Description
b_i	The defense resource allocated to protect a node i, where $i \in I$
h_f	The defense resource allocated to honeypot f, which is the fake core node in the network, where $f \in F$
$a(b_i)$	The cost of compromising a general node i in the network, where $i \in I$
$a(h_f)$	The cost of compromising a honeypot f in the network, where $f \in F$

The objective is to minimize the probability of the core node being compromised by adjusting the defense budget allocated to each node. In this scenario, the defender tries to improve the system's survivability by using resources that can increase the defense level and by assuming that honeypots will distract attackers. Thus, we formulate the attack-defense scenario as an optimization problem.

Objective function:

$$\min_{\overline{D}} \frac{\sum_{k=1}^{K}\sum_{j=1}^{R_k} S_{kj}(\overline{D}, \overline{A_k})}{M} \quad \text{(IP 1)}$$

subject to:

$$\overline{D} \in D \quad \text{(IP 1.1)}$$

$$\sum_{i=1}^{k} R_k = M \qquad \forall k \in K \quad \text{(IP 1.2)}$$

$$\sum_{i \in I} b_i + \sum_{f \in F} h_f \le B \quad \text{(IP 1.3)}$$

$$0 \le b_i \le B \qquad \forall i \in I \quad \text{(IP 1.4)}$$

$$0 \le h_f \le B \qquad \forall f \in F \quad \text{(IP 1.5)}$$

$$\sum_{i \in I} a_i(b_i) + \sum_{f \in F} a_i(h_f) \le B_k \qquad \forall k \in K \quad \text{(IP 1.6)}$$

$$0 \le \sum_{i \in I} a_i(b_i) \le B_k \qquad \forall k \in K \quad \text{(IP 1.7)}$$

$$0 \le \sum_{n \in N} a_i(h_f) \le B_k \qquad \forall k \in K \quad \text{(IP 1.8)}$$

In this model, the defender's objective is to minimize the probability of the system being compromised. The probability is modeled as the number of times (frequency) that the system is compromised divided by the total attack time M. The frequency is governed by the summation of $S_{kj}(\overline{D}, \overline{A_k})$.

Constraint (IP 1.1) requires that the defense resource allocation should be part of a feasible strategy, which means each allocation must satisfy the budget constraint. Constraint (IP 1.2) stipulates that the summation of the rounded frequency of each type of attacker should be equal to M; otherwise, it will cause inconsistency, which may affect the model's accuracy. Constraints (IP 1.3) to Constraint (IP 1.5) are the defender's budget constraints; and Constraints (IP 1.6) to Constraint (IP 1.8) define the attacker's budget limitations.

III. SOLUTION APPROACH

A. Evaluation Process

Since our scenario and environment are very dynamic, it is difficult to solve the problem by mathematical programming alone. The proposed evaluation process enables us to better describe the behavior of different attackers. In each attacker category, there is some randomness in the behavior of the attackers, even though they are classified as the same type. The randomness is caused by honeypots. Recall that if an attacker compromises a honeypot, which is a fake core node, he may believe he has compromised the real core node and terminate the attack. Therefore, we cannot determine whether an attack has been successful until the end of the evaluation.

The total evaluation frequency is set to the same value as M, which is determined by experiment. First, we select an initial value, for example, 10 million. Then, we take 10,000 as a chunk to summarize the result and draw a diagram to depict the relationship between the compromise frequency and the number of chunks. If the diagram shows a stable trend, it implies that the value of M is ideal, since there is no obvious difference between the compromise frequencies of the chunks. On the other hand, if the diagram shows an unstable result, we assume that M is set too small; therefore, we set M to a larger number to run the test experiment.

After deciding the value of M, we can apply our evaluation process to discover the optimal solution. We

have an initial resource allocation configuration. Based on this, we run the evaluation *M* times with all 27 categories of attackers to derive the compromise frequency of the core node. Then, we divide this frequency by *M* to obtain the average probability that the core node will be compromised. We take this result as the benchmark to evaluate the performance of each enhancement process. Reiterate that we can reach more than 27 attacker categories, the 27 profiles employed here is for illustration purpose.

In the next step, we improve the quality of the solution by removing resources from unimportant nodes and adding resources to important nodes. Then, we run the evaluation another *M* times using the adjusted defense parameters and obtain the core node's compromise frequency. Again, we divide the frequency by *M* to determine the average probability that the core node will be compromised. Finally, we check whether one of the stopping criteria is satisfied. If it is, we terminate the enhancement process and compare the final result with the initial state.

The stopping criteria can be divided into two concepts. The first is the total enhancement time, which we set to be no more than *N*. The value of *N* is decided by a resource constrained approach. First, we set the period that defenders are willing to apply this process, e.g., 8 hours. Then, we divide that period by the time required to accomplish one round of policy enhancements. The resulting quotient is the value of *N*.

After determining the value of *N*, if we cannot find any other defense resource allocation scheme that achieves a better performance than the current one, we stop the policy enhancement process. Therefore, in each round we check both conditions to determine if the enhancement process should be terminated.

B. Policy Enhancement

The methodology we use to enhance resource allocation is based on two key concepts: a derivative-based strategy and a popularity-based strategy. The derivative strategy measures the "marginal effectiveness" of the allocation of defense resources. The marginal effectiveness or derivative value is the difference between the present resource allocation strategy and the enhanced strategy. Defenders can make decisions by evaluating the derivative value.

We obtain the derivative value through experiments. After finding the derivative value of one allocation, we reallocate the defense resources; then we evaluate the allocation strategy's performance. In our problem, the performance is represented by the compromise probability of the core node. The difference between the probability of the previous state and that of the enhanced state divided by the total amount of reallocated resources forms our derivative value.

The objective of policy enhancement is to reallocate resources in the network. First, we try to take a certain amount of resources from every node in the network. If all nodes contribute the same amount of resources, we can calculate the total amount of resources that are available for reallocation. (i.e. we multiply the number of nodes in the network by the amount of resources we take from each node).

Next, we use the popularity-based strategy to choose the node(s) that will receive reallocated resources. Then, we take the posterior compromise probability of the core node minus the prior state and divide the result by the amount of reallocated resources to get the derivative value. The above concept can be expressed as follows:

$$\frac{\text{posterior compromise probability of the core node} - \text{prior state}}{\text{the amount of reallocated resources}}$$

Since the numerator of the derivative is the compromise probability of the core node, ideally, the difference between the posterior state and the prior state should be a negative value. Therefore, we choose the scheme that has the lowest derivative to replace the current one.

The popularity-based strategy focuses on nodes that are attacked frequently. Several criteria can be used to determine the kinds of nodes that are "popular" among attackers. In this paper, we take the cost of attacking one node divided by the accumulated cost of attacking every node in the network as the metric for policy enhancement.

We combine the concepts of the derivative and popularity-based strategies to form our policy enhancement scheme. Specifically, we apply the popularity-based strategy to find the most popular nodes, and then calculate the derivative to evaluate the performance. In each evaluation, we first remove defense resources from nodes that have surplus resources, i.e., above a certain threshold. Then, in each enhancement, we use all the resources that the defender can reallocate as a decision metric. If the amount of resources is larger than a predefined threshold, we continue the follow-up procedure; otherwise, we go back to the beginning and change the value for another trial.

In the next step, we determine how to distribute the resources. By using the popularity- based strategy, we can generate a priority list (from high to low), and allocate the resources accordingly. The number of nodes that require extra defense resources is determined through experiments. We test different reallocation methods, such as adding all the resources to a single node or distributing the resources among many nodes in the network. Since the relationship between the defense budget and the defense level forms a concave function, it is difficult to judge whether concentrating or distributing the resources is better without experiments. Therefore, we calculate the derivative value of each reallocation scheme, and take the scheme with lowest derivative value to replace the current resource allocation scheme.

IV. COMPUTATIONAL EXPERIMENTS

A. Experiment Environment

First, we introduce our initial defense resource allocation algorithm, which is also the starting point of our evaluation. Rather than consider just one metric, the algorithm relies on two important pointers, namely, the number of hops from the core node and the link degree of each node.

We combine the two metrics and give a weight to each one. The summation of the two weights should equal 100%. Moreover, to determine which strategy is more important for the defender, we apply 11 ratio configurations to obtain various initial allocation schemes. The first allocation takes 0% of the link degree and 100% of the hop numbers as the initial defense resource allocation. Then, we use the allocation results as input for our evaluation process.

Before the evaluation process, we need to determine the value of *M*. Therefore, we run a number of experiments to find the proper value for our scenario. Specifically, ten thousand attack results are aggregated into a chunk and become one data point, as shown in Figure 1. The vertical axis represents the average compromise frequency of the core node, and the horizontal axis represents the total number of chunks. The results for 10 chunks and 100 chunks were not stable; however, as shown in the figure, we ran 10,000 chunks in the experiments and found that the trend became stable when the number of chunks exceeded 1,000. Therefore, we set *M* as 1,000 chunks, which yielded 10,000,000 evaluations.

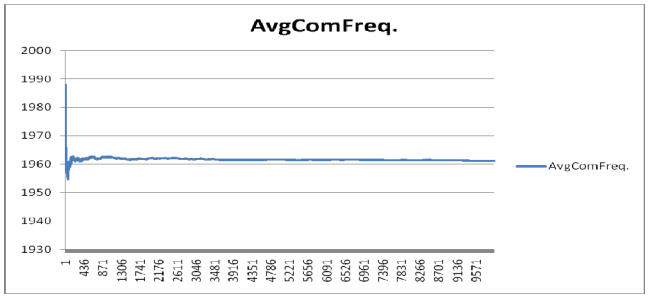

Figure 1. Experiment results when M is set to 10,000 chunks

TABLE 4. IMPORTANT PARAMETERS

Parameter	Value
Total number of attacker profiles	27
Attackers' budget levels	3
Attackers' capability levels	3
Next hop selection criteria	3
Defender's total budget	1,000
Total number of evaluations in one round	10,000,000

TABLE 5. ATTACKERS' BUDGET LEVELS

Types of attackers' budget levels	Multiple of minimum attack cost
High level	2
Medium level	1.5
Low level	1

TABLE 6. PROBABILITY OF DISTRACTING ATTACKERS

Types of attackers' capability levels	Probability of distracting attackers with false targets
High level	30%
Medium level	50%
Low level	70%

Next we discuss the initial allocation scheme and the corresponding policy enhancement scheme. Table 4 lists the key parameters in our experiments. There are 27 attacker profiles, each comprised of 3 budget levels, 3 capability levels and 3 next hop selection criteria. Recall that we apply 27 attacker categories just for illustration purpose. Table 5 shows to the attackers' budget levels; and Table 6 details the probability of distracting attackers. We tried different combinations of the number of hops and link degrees. However, in each combination, the minimum cost of attacking the topology also changes. Recall the method used to determine an attacker's budget. We use multiples of the minimum attack cost to generate the budget. Therefore, when this cost changes, the attacker's budget will be affected. To avoid bias, the benchmark for deciding the attacker's budget is fixed at 443, as shown in Figure 2. This is the minimum attack cost, which comprises 20% of the number of hops and 80% of the link degree in the initial allocation. The curve indicates its performance. The vertical axis represents the compromise probability of the core node and the horizontal axis represents the percentage of defenders that apply the hop count to determine the initial allocation. The figure shows that applying 40% of the hop count and 60% of the link degree yields the best defense result in the initial allocation. Clearly, the performance will be poor if the defender does not apply a hop count strategy. However, we find that applying a larger proportion of the hop count is less effective than applying a smaller proportion.

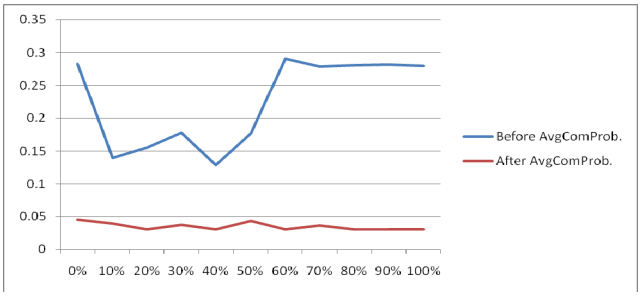

Figure 2. Performance comparison when the benchmark is 443

In Figure 3, the attackers' budgets are set as multiples of the benchmark at 50% of the number of hops and 50% of the link degree. Similar to the previous case, our policy enhancement approach improves the performance of the defense resource allocation scheme and reduces the compromise probability of the core node to 5%.

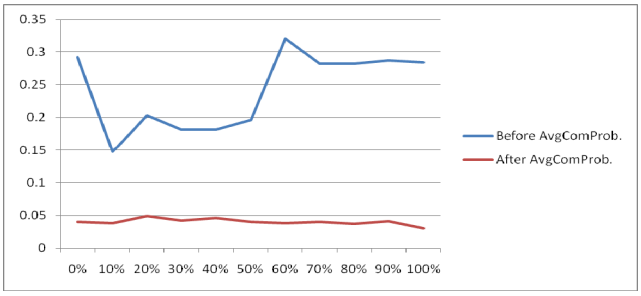

Figure 3. Performance comparison when the benchmark is 480

In Figure 4, the minimum attack cost of 80% of the number of hops and 20% of the link degree is taken as the benchmark to form attackers' budgets for the target topology. In this case, an attacker with a high budget level has more attack power than the defender's total defense budget. Hence, in this scenario, the enhanced allocation approach does not perform as well as in the previous cases. There are some fluctuations in the figure because at least 33% of the attackers can compromise the core node if they are not distracted by false targets, i.e., honeypots. However, our approach can still improve the performance even though the attackers' budgets are larger than the defender's budget.

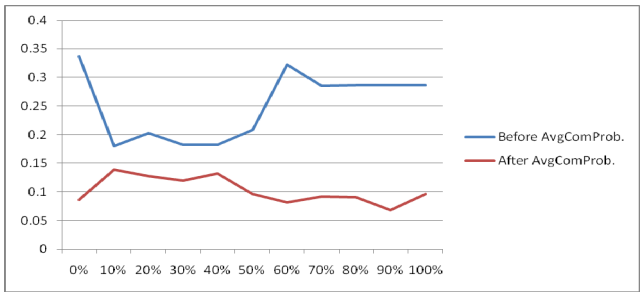

Figure 4. Performance comparison when the benchmark is 515

B. Discussion

The proposed approach can significantly improve a network's survivability, as shown by the performance comparisons in Figures 1, 2 and 3. In most cases, the compromise probability of the core node is reduced to less than 5%. We achieve this result by finding some "important nodes" in the network. For instance, we begin by concentrating the defense resources on one node. If attackers apply the lowest defense level criterion, they will never attack this resource concentrated node. However, our algorithm also uses topology features, so we put a small amount of resources on the node before the resource-concentrated node to trick an attacker applying the lowest defense level criterion to that node. After compromising the node, the attacker will discover that there are no links connecting to the node's neighbors, except the one that leads to the resource-concentrated node. In this way, we can force the attacker to compromise a node that he did not plan to attack originally.

However, this mechanism only works well if the attackers' budgets are not more than the defender's budget. In the scenario where the attackers are more powerful than the defender, our algorithm prefers to allocate resources to multiple nodes whose locations are critical. These nodes form a defense front that can block attackers more effectively, since about 33% of attackers can compromise the core node if they are not distracted by false targets.

V. CONCLUSIONS

In this paper, we relax the commonly made "perfect information assumption for attackers" in previous research and propose a mathematical model to evaluate network survivability. Another feature of this work is that we consider a more realistic environment where multiple classes of attackers may exist, and that attackers from different classes may be of distinct attributes, behaviors and strategies. We then intend to evaluate and improve the proposed metric under the average case (considering the profile of all possible attackers).

Our main contribution is that we combine mathematical programming with simulations and develop a novel approach to solve problems with the imperfect knowledge property. This mechanism helps us extend the scope of problems we can solve. Moreover, our approach can be applied in most cases. It achieves almost the same performance in terms of network survivability, which is approximately 95%, even if the defender applies distinct initial allocation schemes. In cases where the attackers' total budgets are higher than that of the defender, our method still can reallocate defense resources effectively and reduce the compromise probability of the core node.

REFERENCES

[1] R. Richardson, CSI Director, "2008 CSI Computer Crime & Security Survey," 2008.

[2] P.H. Tsang, F.Y.S. Lin and C.W, Chen, "Maximization of Network Survival Time in the Event of Intelligent and Malicious Attacks," Proc. IEEE ICC'08, 2008.

[3] Xingang Wang, Shuguang Guan and Choy Heng Lai, "Protecting infrastructure networks from cost-based attacks," New Journal of Physics, Volume 11, Issue 3, pp. 033006, 2009.

[4] F.Y.S. Lin, P.H. Tsang and Y.L. Lin, "Near Optimal Protection Strategies against Targeted Attacks on the Core Node of a Network," Proc. ARES'07, 2007

[5] A. T. Murray, T. C. Matisziw, T. H. Grubesic, "Critical network infrastructure analysis: interdiction and system flow," Journal of Geographical Systems, Volume 9, Issue 2, pp. 103–117, 2007.

[6] L. J. Zhang, W. Wang, L. Guo, W. Yang, Y. T. Yang, "A Survivability Quantitative Analysis Model for Network System Based on Attack Graph," International Conference on Machine Learning and Cybernetics, Volume 6, pp. 3211–3216, 2007.

[7] C. Seifert, I. Welch, P. Komisarczuk, "Taxonomy of Honeypots," Technical Report CS-TR-06/12, 2006.

[8] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing, Volume 1, Issue 1, pp.11–33, 2004.

[9] H. Debar, F. Pouget, and M. Dacier, "White Paper: "Honeypot, Honeynet, Honeytoken: Terminological issues"," Institut Eurécom Research Report RR-03-081, 2003.

[10] C. K. Dimitriadis, "Improving Mobile Core Network Security with Honeynets," IEEE Security & Privacy, Volume 5, Issue 4, pp.40–47, 2007.

[11] G. Levitin and K., Hausken, "False targets efficiency in defense strategy," European Journal of Operational Research, Volume 194, Issue 1, pp. 155–162, 2009.

[12] Hausken, K., "Production and conflict models versus rent seeking models," Public Choice, Volume 123, pp. 59–93, 2005.

[13] Fred Cohen, "Managing Network Security: Attack and Defence Strategies," Network Security, Volume 1999, Issue 7, pp. 7–11, July 1999.