J. Innovative Eng. Nat. Sci., vol. 4, no. 1, pp. 245-262, 2024.
http://doi.org/10.61112/jiens.1389871

JIENS
Original Paper
Journal of Innovative Engineering and Natural Science
(Yenilikçi Mühendislik ve Doğa Bilimleri Dergisi)
journal homepage: https://jiens.org

Optimal defense strategies against intelligent cyber attacks

Mehmet Ertem[a,*] and Vicki M. Bier[b]

[a]*Department of Industrial Engineering, Eskişehir Osmangazi University, Eskişehir26040, Turkey.*
[b]*Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53705, USA.*

ARTICLE INFO

Article history:
Received 16 Nov 2023
Received in revised form 22 Dec 2023
Accepted 21 Jan 2024

Available online

Keywords:
cybersecurity
stochastic programming
defender attacker model
bilevel optimization
network interdiction
integer L-shaped method

ABSTRACT

We propose a comprehensive game-theoretic model pertaining to the security of computer networks, specifically addressing the interaction between defenders and attackers. The model incorporates attack graphs to outline potential attacker strategies and defender responses. To account for the attacker's capacity to execute multiple attempts, we introduce a probabilistic element, wherein the success or failure at any arc of the attack graph is treated as stochastic. This characterization gives rise to a multi-stage stochastic network-interdiction problem. In this problem formulation, the defender strategically interdicts a set of arcs in anticipation of the likely actions of the attacker, who, in turn, can make multiple attempts to traverse the network. We mathematically articulate this scenario as a stochastic bilevel mixed-integer program with a "min-max" objective. The defender's aim is to minimize the probability of the attacker's success, while the attacker seeks to maximize the probability of successfully traversing the network across multiple attempts. The defender's stochastic bilevel optimization model is solved using the integer L-shaped method. Upon analyzing the defender's perspective, we observe the anticipated trend that the overall success probability of the attacker diminishes with an increasing level of defense. Notably, in the sensitivity analysis involving relatively small attack graphs, we discover that the optimal defense strategy against a myopic attacker often aligns with that against a non-myopic attacker. Furthermore, in instances where deviations exist, the disparity in performance is generally marginal. However, our findings demonstrate a potential divergence in optimal defense strategies when the available attack paths share numerous common arcs.

I. INTRODUCTION

In an era dominated by technology and connectivity, the safeguarding of cyber networks against intelligent and persistent attacks has become paramount. Cybersecurity threats, driven by sophisticated attackers employing advanced techniques, demand innovative and robust defense strategies. The significance of cybersecurity cannot be overstated, given the escalating frequency and complexity of cyber-attacks. According to the Cybersecurity & Infrastructure Security Agency (CISA), there was a 33% increase in reported cyber incidents in 2022 compared to the previous year [1]. This surge highlights the pressing need for effective defense mechanisms against evolving threats.

Computer security has been studied extensively in literature. The main tools for protecting networked systems include cryptography, “demilitarized zones” (designated places on a computer network where public services are located), virtual private networks (VPN), honey pots, vulnerability scanners, firewalls, and intrusion detection systems (IDSs) [2]. Encryption is a strong tool for securing data packets, but it is not always realistic to encrypt every packet in a network, and in any case encryption keys can often be stolen or acquired through “social engineering” even if the encryption code cannot be “cracked” by brute computational force. A VPN is a way to connect securely with the network a private network from outside; however, this connection can also provide a

*Corresponding author. Tel.: +90-544-367-3441; e-mail: mertem@ogu.edu.tr

way for attackers to get inside the network, if they are able to obtain the privileges of an authorized user. Honey pots are decoys designed to attract the interest of attackers; however, if an attacker can identify the honey pots, this could provide knowledge of the network structure. Vulnerability scanners are used to scan a given network for potential weaknesses, but their findings are not always reliable. Finally, firewalls are designed to protect networks from malicious activities, and IDSs identify attack signatures and detect anomalous system behavior. However, if cyber attackers know their general capabilities, they may be able to take advantage of security holes or gaps. Attackers can also exploit the complexity of computer networks by finding a multiplicity of ways to attack them [3].

When computer networks were first introduced, they were not designed to protect against potential attacks [2]. Moreover, with the exponential growth of computer and internet usage, computer networks became extremely complex. Therefore, it is difficult if not impossible to secure computer networks completely, and intelligent attackers can always take advantage of new weaknesses or combinations of existing vulnerabilities. Moreover, operational decisions about updating network security often involve patching newly discovered security holes after an attack. In other words, once a new virus or cyber-attack has been detected, firewalls, intrusion-detection algorithms, and other defense tools are updated accordingly. However, game-theoretic defense approaches could in principle facilitate a more proactive approach to computer security, by anticipating the responses of potential attackers [4].

Given the intricate nature of cyber networks, attack graphs have emerged as a method for comprehensively representing all potential attack scenarios within a specific network context [5]. These graphs serve as concise visualizations, illustrating the vulnerabilities inherent in each cyber network. Consequently, numerous scholars have incorporated attack graphs into their research, employing them as foundational elements to construct intricate security models for cyber networks [6].

An attack graph serves as a comprehensive representation of all conceivable attack scenarios within a computer network [5]. By utilizing attack graphs, one can construct defender-attacker models aimed at assisting defenders in identifying optimal or nearly optimal defense strategies. Additionally, these models can be tailored to defend systems against various attacker types, including hackers, spies, terrorists, corporate raiders, and professional criminals. Each attacker type may have distinct objectives; for instance, opportunistic attackers seek easy targets and may switch focus if a specific target becomes too challenging or costly to attack successfully. Conversely, determined attackers, such as hostile governments, are less deterred by increased attack difficulty or cost.

In the realm of computer network security, numerous defender-attacker models have been developed, primarily leveraging game theory to pinpoint equilibrium strategies for both attackers and defenders. However, many of these models neglect the integration of attack graphs into their game trees. For example, Liu et al. [7] devised a game-theoretic formalization, featuring diverse models tailored to specific situations, such as accounting for the correlation among attack actions and the accuracy of intrusion detection. Lye and Wing [8] explore varied game strategies within network security, treating interactions between attackers and system administrators as a two-player stochastic game. They compute Nash equilibria and best-response strategies for both the attacker and the administrator using nonlinear programming techniques. By framing the problem as a general-sum stochastic game, Lye and Wing identify multiple Nash equilibria, aiding system administrators in anticipating the attacker's optimal attack strategies. Similarly, Xiaolin et al. [9] propose a Markov model to assess the risks of computer networks,

considering the current and anticipated future security status. They use a Markov chain to model the potential actions of attackers to assess the risk, and another Markov chain to represent possible defense strategies to decrease the number of system vulnerabilities. Furthermore, Nguyen et al. [10] propose a stochastic game-theoretic model for security and intrusion detection in computer networks and apply it to a small sample network. They model the security of the network using a weighted directed graph, with nodes representing combinations of security assets and vulnerabilities, and with edges representing relations among the nodes.

Additionally, there exist defender-based methodologies that do not explicitly consider attacker types or strategies. For instance, Sheyner et al. [5] propose an approximation method to identify the smallest subset of measures necessary to render a system "safe." They acknowledge the NP-completeness of finding such a subset and employ a greedy algorithm to derive an approximate solution. In their approach, they treat the identification of a minimum critical set in an attack graph akin to solving a minimum hitting-set problem. The resulting algorithm determines the minimum critical set covering at least one arc from each possible path on the attack graph. However, it is crucial to note that their model may not fully align with real-world scenarios where security systems are subject to budget constraints, a factor often influencing defense strategies.

One way to use attack graphs to develop defensive strategies is to find the optimal set of arcs (or, equivalently, attacker actions) for the defender to interdict. This critical set is likely to depend on factors such as interdiction costs, the defender’s budget constraint, the effect of an arc interdiction on the attacker’s future actions, etc. In the literature, many network-interdiction algorithms have been developed to find the optimal set of arcs to interdict on a given graph. Network-interdiction models have been studied in areas such as military applications and transportation security. During the Vietnam War, McMasters and Mustin [11] developed deterministic models to interdict enemy troops or material flows on a transportation network, using optimization to allocate a limited number of aircraft to best interdict the enemy’s supply line. More recently, [12-14] have all applied deterministic network-interdiction models to military problems, or to interdiction of illegal drugs or precursor chemicals. Similarly, [15-16] study the problem of maximizing the shortest path of an attacker, using a deterministic network-interdiction algorithm.

In situations where one or more components of a problem lack certainty, a stochastic network-interdiction problem arises. One of the pioneering investigations into stochastic network interdiction was conducted by Cormican et al. [17]. They introduced a stochastic network-interdiction model aimed at minimizing the anticipated maximum flow achievable by an adversary within a given network. Cormican et al. [17] employed binary random variables to represent interdiction success or failure and analyzed the ensuing problem as a two-stage stochastic integer program. In the initial stage, the defender's objective is to minimize the maximum network flow attainable by the adversary by interdicting a specified set of arcs. In the subsequent stage, the attacker selects the path on the network that yields the maximum flow. Cormican et al. [17] employed a sequential-approximation algorithm, generating lower and upper bounds for the optimal value of the objective function at each iteration. Israeli and Wood [18] focused on a shortest-path network-interdiction problem, wherein the defender interdicts a set of arcs to maximize the shortest path achievable by the attacker. The stochastic nature of their problem arises from the assumption that the defender lacks precise knowledge of the attacker's actual origin and destination, possessing only a probability distribution over potential origin-destination pair. They formulated the resultant bilevel max-min problem as a stochastic mixed-integer program. Bayrak and Bailey [19] extended the shortest-path network-interdiction

problem to cases where the interdictor and the evader possess varying levels of knowledge about the network. Their formulation transformed the problem into a stochastic nonlinear mixed-integer program, subsequently solved by converting it into a stochastic linear mixed-integer program. Pan and Morton [20] proposed a stochastic network-interdiction model where the interdictor selects a set of arcs for installing radiation sensors to minimize the reliability of the evader's maximum-reliability path, with the goal of maximizing the probability of detecting a potential nuclear smuggler. Like Cormican et al. [17], Pan and Morton [20] assumed that the interdictor only possesses a probability distribution for the evader's origin-destination pair. They formulated the resulting problem as a stochastic mixed-integer program and presented a decomposition method for its solution. Similarly, Dimitrov and Morton [21] developed various network-interdiction models and applied them to diverse problems. Recently, Morton [22] conducted a comparative analysis of deterministic and stochastic network-interdiction models and their respective solution algorithms. Additionally, they categorized the types of stochastic network-interdiction models based on the defender's information level regarding the attacker's source-terminal pair.

Stochastic network-interdiction models present a greater computational challenge compared to their deterministic counterparts, prompting the introduction of various solution algorithms and heuristics. Cormican [23] has classified available solution methods into two categories: decomposition methods and sequential-approximation algorithms. Generally, decomposition algorithms iteratively enhance the bounds of the objective function by solving both the attacker's and the defender's problems for every conceivable realization of the scenario. Examples of decomposition algorithms encompass [17-20, 24]. However, it is imperative for the objective function of the problem to be convex for decomposition algorithms to be applicable. Additionally, certain decomposition algorithms necessitate the dualization of the objective function, a task that is not always straightforward and, in some instances, may prove to be infeasible. Furthermore, when dealing with a large number of potential scenarios, computational challenges may arise with decomposition methods due to the escalating number of subproblems requiring resolution at each iteration, corresponding to the number of scenarios. In contrast, sequential-approximation algorithms exhibit enhanced computational efficiency in scenarios involving a substantial number of possible scenarios. Similar to decomposition, sequential-approximation algorithms establish upper and lower bounds on the optimal value of the objective function. In this approach, the genuine problem is approximated by progressively refining the partitioning of the sample space to create more representative subsets, thereby yielding tighter bounds on the optimal value of the objective function. Sequential-approximation algorithms utilize decomposition at each iteration to ascertain bounds on the optimal value of the objective function within each partition, thereby encountering analogous constraints as decomposition. Notable instances of sequential approximation algorithms are elucidated in [17, 25].

Recently, Janjarassuk and Linderoth [26] introduced a third method, sample-average approximation (SAA), for addressing stochastic network-interdiction problems. In SAA, the genuine problem is approximated by generating samples from the state space, representing all possible scenarios of the problem. Once the samples are generated, the SAA problem transforms into a deterministic optimization problem, effectively approximating the true stochastic problem. Deterministic optimization problems can frequently be solved optimally using suitable algorithms such as the L-shaped decomposition algorithm [27]. Examples of SAA methods are outlined in [28-33]. In their publication, Janjarassuk and Linderoth [26] conduct a comparative analysis of different methods for solving stochastic network-interdiction problems and conclude that SAA exhibits notable effectiveness for large problems, as a relatively modest number of samples proves sufficient to represent numerous possible scenarios.

This paper proposes an innovative defense approach grounded in the principles of stochastic network interdiction, an area of study gaining traction in cybersecurity research. By integrating bi-level stochastic optimization model, this approach aims to optimize the allocation of defensive resources, minimizing vulnerabilities and maximizing the network's resilience against intelligent attacks. To address the complexity of the problem, this study employs the Integer L-shaped method and SAA which provide a rigorous and efficient solution methodology.

II. THEORETICAL METHOD

In this study, a fully game-theoretic and non-myopic approach is adopted to represent attacker and defender behavior. We set up the game as a defender-attacker model because the defender moves first, and we assume that the attacker's objective is maximizing the probability of success while the defender minimizes the same objective function value with limited resources. In our model, we use attack graphs to represent attack success probabilities. Each arc on an attack graph represents the probability of successfully traversing one step of a complete attack (as represented by a path). We assume that the outcomes of attempting to traverse different arcs are independent. Our game is non-cooperative, because the defender wishes to minimize damage while the attacker attempts to maximize damage. Also, it is a stochastic game, because it is uncertain whether the attacker will be successful when attempting to traverse the network, and the defender develops his strategy anticipating this uncertainty. We can further categorize our game as a sequential game because the defender moves first to protect the network, and then the attacker launches an attack to traverse the protected network. Further, we assume a complete-information game between the attacker and the defender, which means that both the defender and the attacker know each other's possible strategies and payoffs. Finally, we assume a game of perfect information, where the attacker has knowledge of the defender's chosen protection strategy when choosing how to attack the network.

We formulate the attacker's problem as a multi-stage stochastic network-interdiction problem (MSNIP). When attacking the system, the attacker's success is uncertain. Thus, the attacker may need to make several attempts. The defender should anticipate this in choosing a defensive strategy. Thus, the problem becomes a bi-level stochastic programming problem when we consider both the defender and attacker objectives. In our model, each of the attacker's attempts represents a stage of MSNIP. For example, the attacker's first attempt represents the first stage of the MSNIP, the second attempt represents the second stage if the attacker fails at the first attempt, and the third attempt represents the third stage if the attacker fails at the second attempt, etc. MSNIP was first introduced by Ertem and Bier [34] and details of the model can be found in their study. For the purposes of this study, we formulated the defender's problem using the integer L-shaped method.

2.1 Integer L-shaped Method

Bilevel stochastic programming problems are difficult to solve. In literature, there are algorithms that can solve bilevel stochastic programming problems under certain conditions. However, to our knowledge, all these algorithms require the lower-level problem (the attacker's problem, in our context) to be a linear optimization problem [35]. By contrast, in our model, the attacker's problem is a stochastic mixed-integer program, because we use binary decision variables to define the optimal path for the attacker. For the defender's problem, we also use binary decision variables to define which arcs are interdicted by the defender. Several studies have developed

solution algorithms for specific stochastic integer programming problems [36-39]. However, Laporte and Louveaux [40] introduced the integer L-shaped method, which has general applicability to stochastic integer programs. In their study, Laporte and Louveaux [40] show that the integer L-shaped method finds the optimal solution of a problem (if it exists) in a finite number of steps, if the problem has a valid set of feasibility cuts and a valid set of optimality cuts. According to Laporte and Louveaux [40], stochastic integer programs have both valid feasibility cuts and valid optimality cuts. Thus, the integer L-shaped method can be used for the solution of stochastic integer programs provided that: 1) the lower-level expectation function $F(x, y, \xi)$ (the attacker's problem, in our model) is computable for a given upper-level (defender) decision variable *x*; and 2) the objective value of the lower-level optimal solution has a lower bound.

We can consider our problem a bilevel stochastic integer program where the defender interdicts a set of arcs at the upper level, and then the attacker makes multiple attempts to traverse the network at the lower level. The first assumption above (computability) is satisfied if there are enough paths for the attacker to traverse on a given attack graph. For the two-stage case, there should be at least two available paths, and for the three-stage case, at least three paths should be available, because the attacker must select another path in each attack stage if he fails at the previous attack stage. Moreover, for sufficiently large defender budget levels, the defender might be able to interdict all or most of the paths, in which case the attacker's problem $F(x, y, \xi)$ would not be computable. This situation is avoided, however, because of our assumption that if the defender interdicts arc *arc (i,j)* $\in A$ (*A*: set of arcs) on the attack graph, then the probability of successfully attacking *arc (i,j)* $\in A$ will be reduced from p_{ij} to q_{ij} > 0. The second assumption (boundedness) holds because the objective value of the attacker's problem (the probability of succeeding in multiple attempts) is naturally bounded below by 0.

For stage *K* of the integer L-shaped method, we define the master problem as follows:

$$min\ \theta \tag{1}$$

s.t.

$$\sum_{(i,j)\in A} c_{ij}x_{ij} \leq B \tag{2}$$

$$\theta \geq v(x_k) - v(x_k)(\sum_{(i,j)\in A:(x_{k(ij)}=0)} x_{ij}),\ k = 1,2,\dots,K \tag{3}$$

$$v(x_k) \geq 0, \theta \geq 0, k = 1,2,\dots,K \tag{4}$$

Here, variable θ represents an appropriate approximation of $F(x, y, \xi)$; i.e., an approximation satisfying Constraint 3, which represents the *K* identified optimality cuts. Function $v(x_k)$ represents the solution of the attacker's SAA problem with N scenarios [34] for a given defender solution x_k. Constraint 2 represents the defender's budget

limit. As discussed above, because our problem satisfied both computability and boundedness assumptions, the optimality cuts in Constraint 3 will always be valid.

We use the following steps to solve the defender's problem with the integer L-shaped algorithm:

- Step 0: Set $\beta = 0$. The variable θ is the lower bound for the objective function value of the defender problem, so is set equal to 0. An initial defender solution x^{β} is selected, and the corresponding attacker's problem $v(x^{\beta})$ is solved. Set $\bar{\gamma} = v(x^{\beta})$, where $\bar{\gamma}$ is the upper bound for the objective function value of the defender problem. If $\bar{\gamma} = 0$, then go to Step 4. Otherwise, set K = 1, add one optimality cut to Constraint 3, and go to Step 1.
- Step 1: Set $\beta = \beta + 1$. Solve the master problem in Eq. 1 through 4. Let $(x^{\beta}, \theta^{\beta})$ be an optimal solution.
- Step 2: Compute $v(x^{\beta})$ and set $\gamma^{\beta} = v(x^{\beta})$. If $\gamma^{\beta} < \bar{\gamma}$, set $\bar{\gamma} = \gamma^{\beta}$.
- Step 3: If $\theta^{\beta} = \bar{\gamma}$, then go to Step 4. Otherwise, set K = K + 1, add one optimality cut to Constraint 3, and return to Step 1.
- Step 4: Set x^{β} as the optimal defender solution and stop.

Figure 1 illustrates the flowchart depicting the application of the proposed integer L-shaped method to optimize the defender's problem based on the specified attacker's selection.

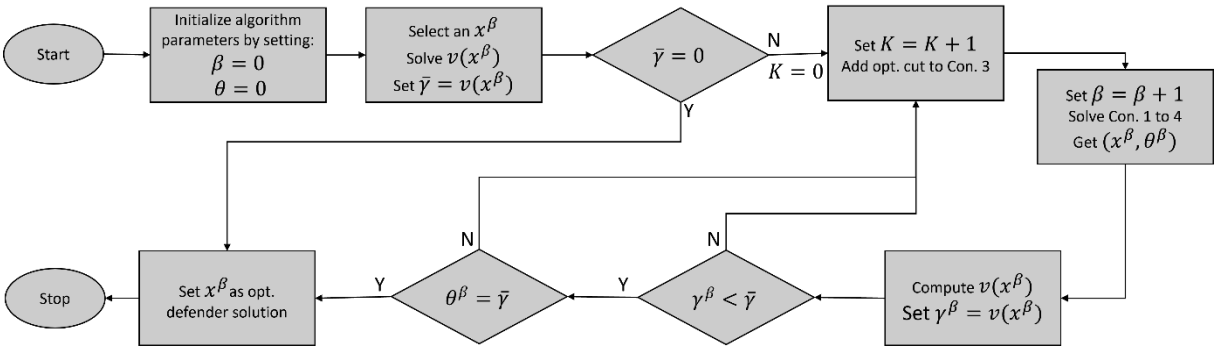

Figure 1. Flowchart of the proposed integer L-shaped method

III. RESULTS AND DISCUSSIONS

In this section, we explore the effect of the attack graph size and structure, attack success probabilities, attacker type, and defense level on the optimal solution of the defender problem. As discussed in Section 2, we will use the integer L-shaped method to solve the defender problem.

3.1 Sensitivity Analysis Setup

In our sensitivity analysis, we use three realistic network topologies (from literature [41-42]) and two levels of vulnerability densities per host. When selecting network topologies, our aim was to select common and realistic

topologies and use our model to obtain some insight into attacker behaviors and optimal defense policies. Therefore, we selected networks like the fully connected, and star-type networks for use in our sensitivity analysis. In a fully connected network, each of the nodes (hosts) is connected to every other node, and in a star topology, every node is directly connected to the central node. In total, we have six attack graphs (three topologies, with two vulnerability densities for each topology). The topologies and corresponding attack graphs for both one and two vulnerabilities per host are given in Figures 2 through 4. For each topology, we begin the analysis with one vulnerability per host, and then add another appropriate vulnerability to each of the nodes in the network.

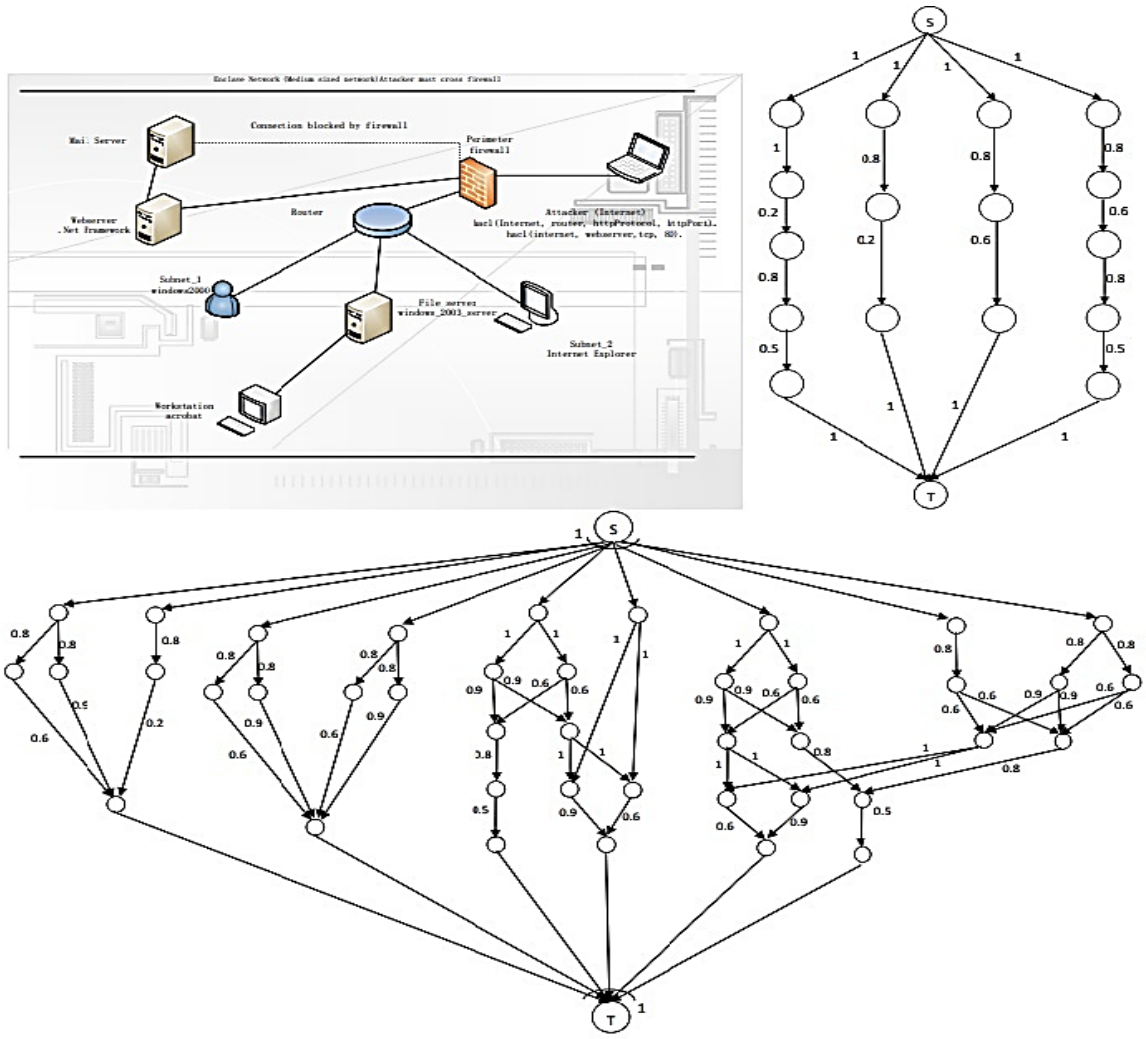

Figure 2. Case 1: Larger star-type network, and corresponding attack graphs with one vulnerability per host (top left), and two vulnerabilities per host (bottom)

The attack graphs used in the case study are generated using the MulVal software [43]. However, the resulting attack graphs include some types of nodes that are not necessary for our assessment. Therefore, in order to simplify the attack graphs, we deleted those nodes. Also, for each attack graph, we added dummy source and terminal nodes (since our model assumes a single source node and terminal node). We connect the source node to the initial nodes of the attack graph with arcs that have success probabilities of 1, and similarly for the terminal node and the final nodes of the attack graph. We start with a base case that assigns initial success probabilities to the arcs in each

attack graph. Initial success probabilities are assigned using the Common Vulnerability Scoring System (CVSS), which is an industry standard for assessing the severity of computer security vulnerabilities [44]. The National Institute of Standards and Technology tracks all emerging cyber-attacks and their corresponding CVSS scores in a National Vulnerability Database (NVD) [45]. In our case, the MulVal software automatically assigns success probabilities using the NVD database.

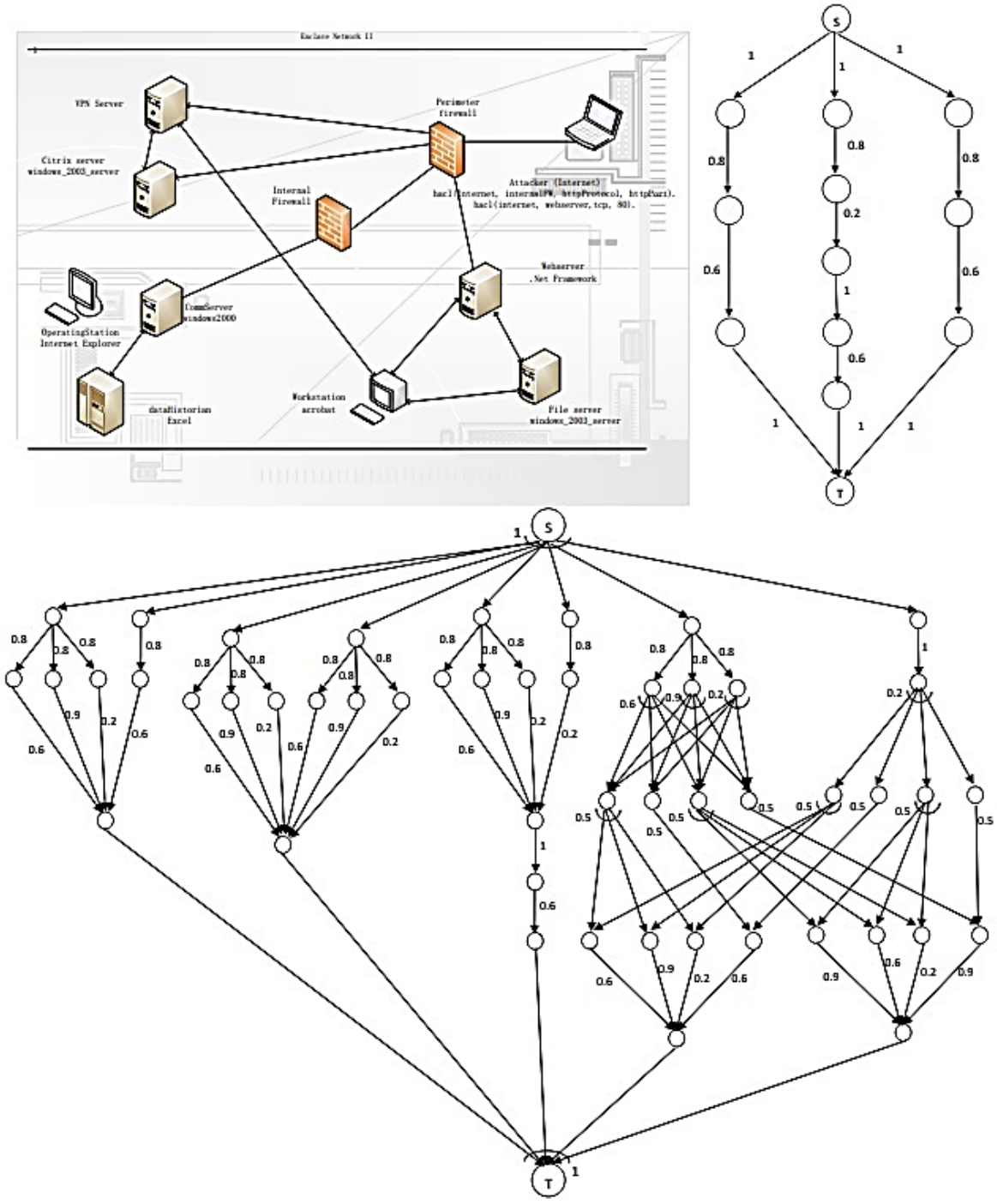

Figure 3. Case 2: Heavily connected network, and corresponding attack graphs with one vulnerability per host (top left), and two vulnerabilities per host (bottom)

To explore the impact of success probability changes on the optimal defense strategy, we also vary the success probabilities of arcs on the attack graphs. We could explore the effects of increasing the success probabilities of arcs, but making arcs more vulnerable does not seem realistic in the context of computer security. Thus, we decreased the success probabilities of arcs on all attack graphs by 50% in our sensitivity analysis. In addition to changes in the success probabilities of arcs, we also explore the effect of the defense level. We begin by solving each attacker problem with no defenses, and then increase the number of arcs that the defender can interdict, until enough arcs have been interdicted so that the attacker can no longer reach the target value.

We consider both myopic and non-myopic attackers and vary the number of attack stages. We consider a one-stage model (for which there is no difference between myopic and non-myopic attackers), as well as two-stage and three-stage models for both myopic and non-myopic attackers. However, we did not apply the three-stage model to cases 1 and 2 (given in Figures 2 and 3, respectively) with two vulnerabilities per host, because the attack graphs for those cases were relatively large and computationally demanding.

3.2 Application of the Integer L-Shaped Method

To solve the defender's optimization problem, we used the integer L-shaped method. As discussed in Section 2.1, the integer L-shaped method finds the optimal solution of the defender problem by adding optimality cuts to the master problem at each iteration. The number of iterations needed to solve the defender problem is an important measure for assessing the performance of this solution approach, because of the need to solve the attacker problem at each iteration.

For a one-stage attack, we first solve the attacker problem to find the optimal attack path for a given initial defender solution, find a candidate solution of the defender problem using the optimal result of the attacker problem, and then repeat these steps at each iteration until we find the optimal defender solution. For the two-stage myopic model, we first solve the attacker problem to find the optimal one-stage attack path for a given initial defender solution, then run the second-stage recourse model to assess the performance of that strategy, solve the master problem using the results of the recourse problem to find a candidate defender solution, and again repeat these steps at each iteration until we find the optimal defender solution.

For the two-stage non-myopic model, we first solve the two-stage attacker problem to find the optimal first-stage attack path, then run the second-stage recourse model to assess the performance of that strategy, solve the defender problem using the results of the recourse problem to find a candidate defender solution, and then repeat these steps at each iteration until we find the optimal defender solution. The three-stage myopic and non-myopic attacker problems are solved in a similar manner.

Based on the results of Case 1 shown in Table 1, the number of iterations needed to solve the defender problem seems to depend on the defender's budget level, and the selected initial defender solution (even though the table does not show the selected initial defender solution). The results are similar for the other cases. For smaller defender budgets, the integer L-shaped method performed no better than explicit enumeration, since it tried all possible defender solutions. For larger budget levels, the method performed significantly better than explicit enumeration. However, the sensitivity runs for the defender problem were still computationally demanding, especially for the larger attack graphs.

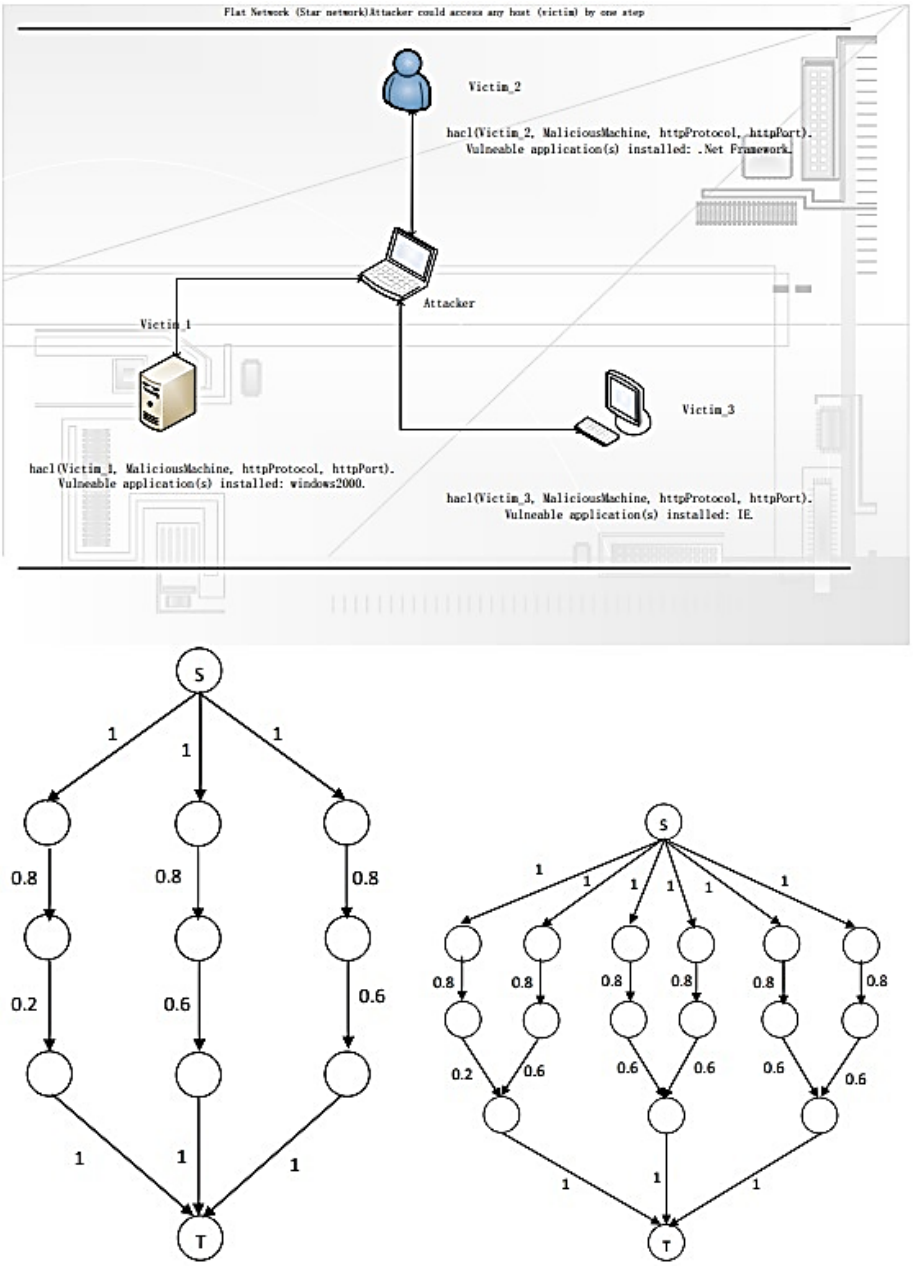

Figure 4. Case 3: Small star network, and corresponding attack graphs with one vulnerability per host (bottom left) and two vulnerabilities per host (bottom right)

5.3 Sensitivity Analysis Results

We use three realistic attack graphs with two vulnerability levels each to assess the sensitivity of the defender solution to attack graph type and size, attacker type (myopic or not), number of attack stages, and defense level. The case study results are illustrated in Figures 5 through 7. Based on the results, we see that the attacker's success probability is generally increasing in the number of attack stages, especially when few arcs are interdicted. As expected, smaller arc success probabilities yield lower overall attack success probabilities, as does an increase in the defender budget level, in general.

Table 1. Number of iterations needed to solve the defender problem using the integer L-shaped method versus using explicit enumeration

	Case 1: 1 vulnerability/host *(12 arcs)*		**Case 1: 2 vulnerability/host *(52 arcs)***	
Budget	**L-Shaped Method**	**Explicit Enumeration**	**L-Shaped Method**	**Explicit Enumeration**
B=1	12	12	52	52
B=2	144	144	2704	2704
B=3	220	220	2704	22100
B=4	N/A	N/A	2704	270725
B=5	N/A	N/A	22100	2598960
B=6	N/A	N/A	22100	20358520
B=7	N/A	N/A	2704	133784560
B=8	N/A	N/A	2704	752538150
B=9	N/A	N/A	2704	3679075400
B=10	N/A	N/A	2704	15820024220
B=11	N/A	N/A	2704	60403728840
B=12	N/A	N/A	52	206379406870
B=13	N/A	N/A	N/A	N/A

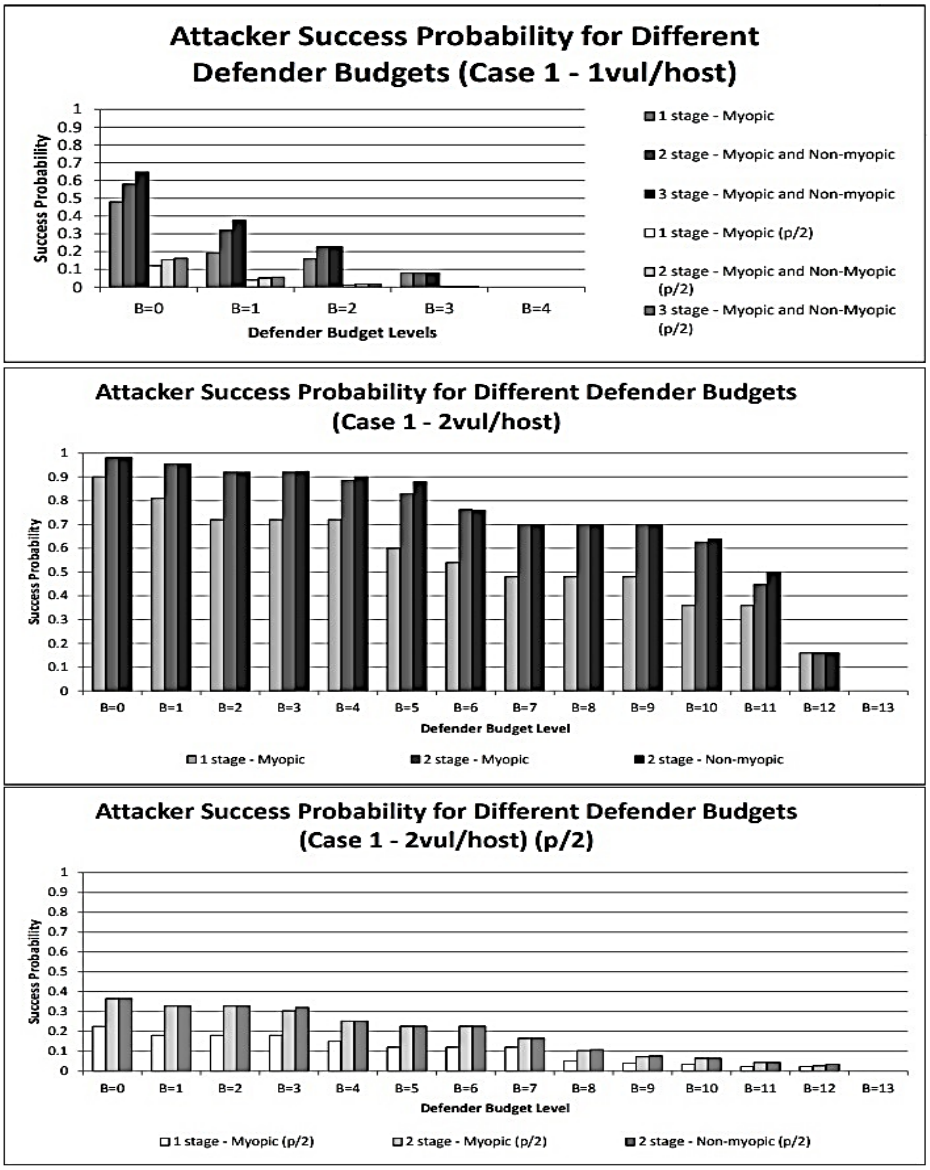

Figure 5. Sensitivity analysis results of the defender problem for Case 1 with one vulnerability per host for both with the base and 50% decreased arc success probabilities (top), with two vulnerabilities per host and base arc success probabilities (middle), with two vulnerabilities per host and arc success probabilities decreased by 50% (bottom chart)

However, when multiple paths in an attack graph have the same success probability, interdiction of an arc on one of those paths will not reduce the overall probability of success, because the attacker could select an equally good path.

In cases 1 and 2 with one vulnerability per host (Figures 5 and 6, respectively), and case 3 (Figure 7), the myopic attacker performs just as well as the non-myopic attacker. This is because in the corresponding attack graphs, the paths do not involve any common arcs, so there is no benefit to the attacker from thinking ahead. In cases 1 and 2 with two vulnerabilities per host (Figures 4 and 5, respectively), the non-myopic attacker does slightly better than the myopic attacker at some budget levels, because in these attack graphs at least some paths involve common arcs. However, even when the non-myopic attacker performs better than the myopic attacker, the defender's optimal interdiction strategy is the same for both attacker types. Thus, the added computational difficulty of solving the non-myopic attacker problem does not benefit the defender. Based on these results, we observe that anticipating the attacker's future actions is especially important when the available attack paths share more common arcs.

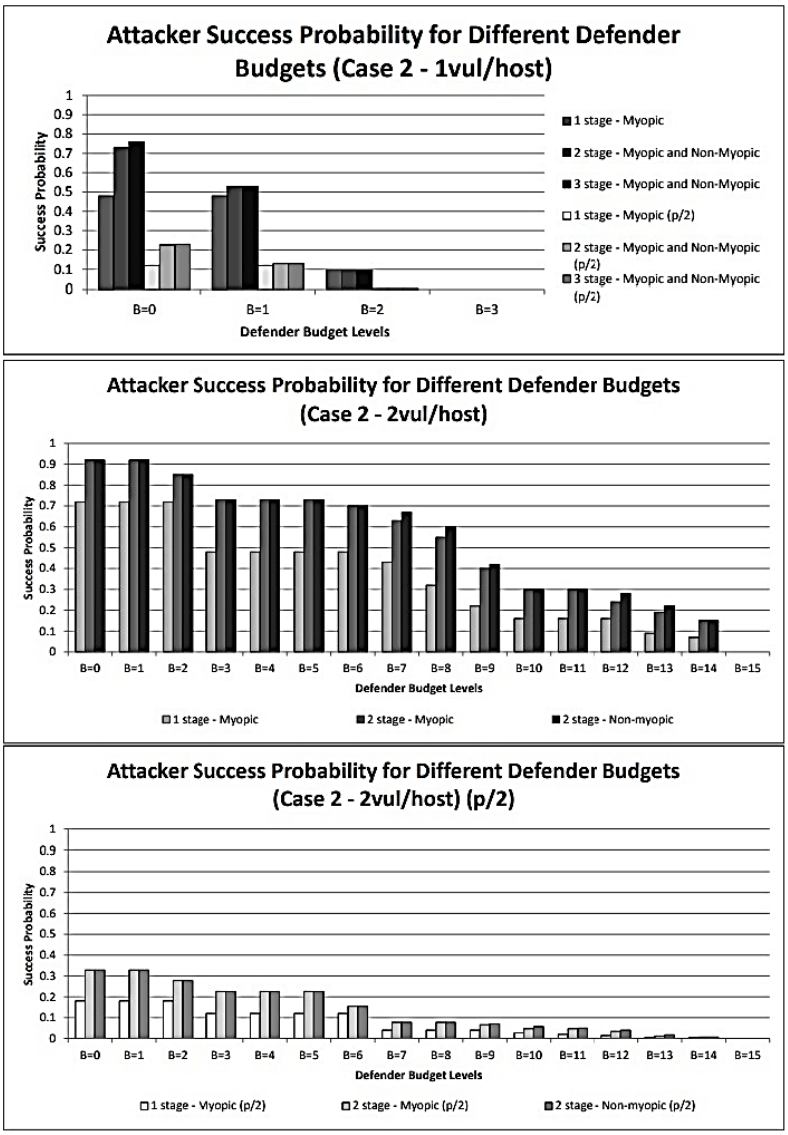

Figure 6. Sensitivity analysis results of the defender problem for Case 2 with one vulnerability per host for both with the base and 50% decreased arc success probabilities (top), with two vulnerabilities per host and base arc success probabilities (middle), with two vulnerabilities per host and arc success probabilities decreased by 50% (bottom chart)

Finally, we observed that the optimal defense is usually to protect arcs near the source node or end node of the attack graph. In our defender sensitivity analysis, we solved 91 defender problems on 7 different attack graphs. In all these runs, the interdicted arcs were next to either the source node or the end node, never at intermediate levels of the attack graph. For example, Figures 8 and 9 illustrate the interdicted arcs for cases 1 and 2 with two vulnerabilities for a defender budget of 10 arcs, respectively.

IV. CONCLUSIONS

In this study, we developed a general game-theoretic defender-attacker model for security of computer networks, using attack graphs to represent the possible attacker strategies and defender options. To represent the ability of the attacker to launch multiple attempts, we consider the attacker's success or failure on any arc of the attack graph to be probabilistic and describe the resulting security problem as a multi-stage stochastic network-interdiction problem. In our problem, the defender interdicts a set of arcs, anticipating the attacker's likely actions, and then the attacker can make multiple attempts to traverse the network. We formulated the resulting problem as a stochastic bilevel mixed-integer program with a "min-max" objective. Here, the defender attempts to minimize the attacker's success probability, and the attacker maximizes the probability of traversing the network successfully in multiple attempts.

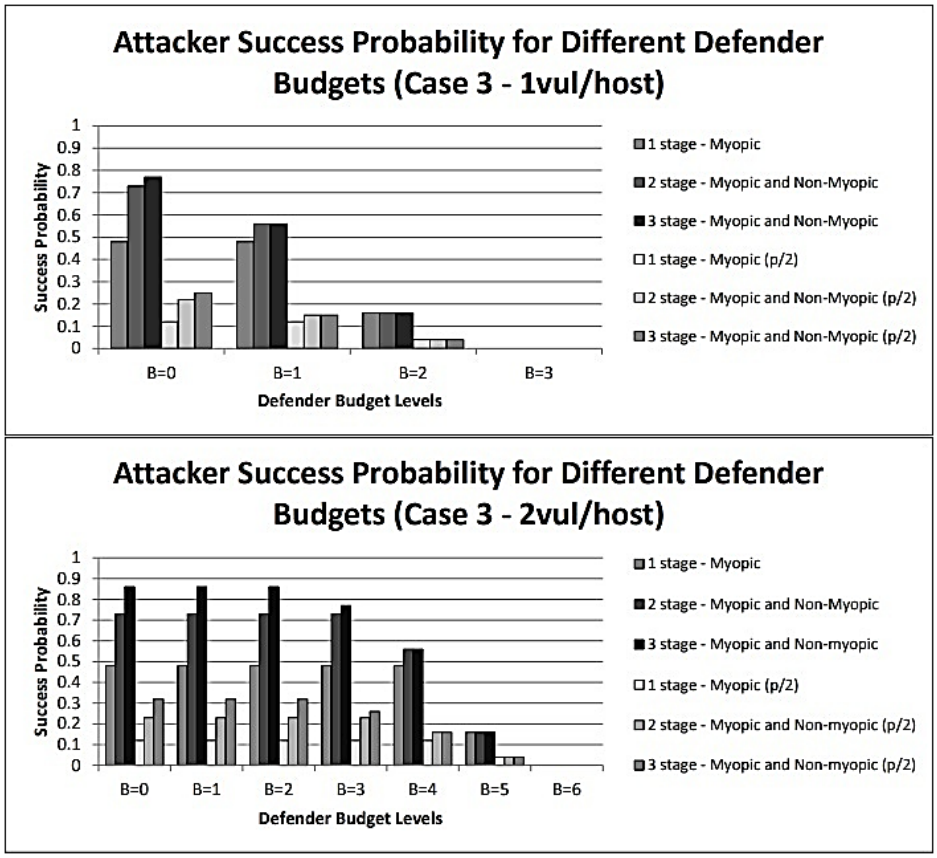

Figure 7. Sensitivity analysis results of the defender problem for Case 3 with one vulnerability per host for both with the base and 50% decreased arc success probabilities (top), with two vulnerabilities per host for both with the base and 50% decreased arc success probabilities (bottom chart)

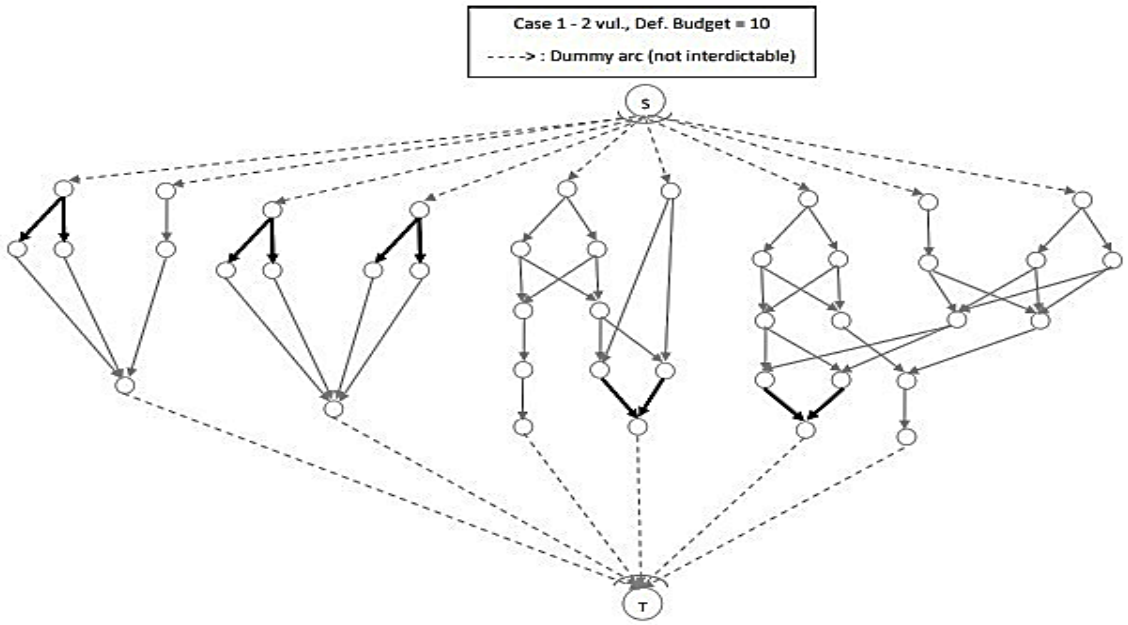

Figure 8. Illustration of the optimal defense strategy for case 1 with 2 vulnerabilities for a defender budget of 10 arcs

For the defender problem, we found, as expected, that the attacker's overall success probability decreases with the increase in the level of defense. More significantly, we found that for the relatively small attack graphs studied in our sensitivity analysis, the optimal defense against a myopic attacker is often the same as against a non-myopic attacker, and in any case rarely does too much worse. However, preliminary results suggest that this might not be the case when the available attack paths share numerous common arcs.

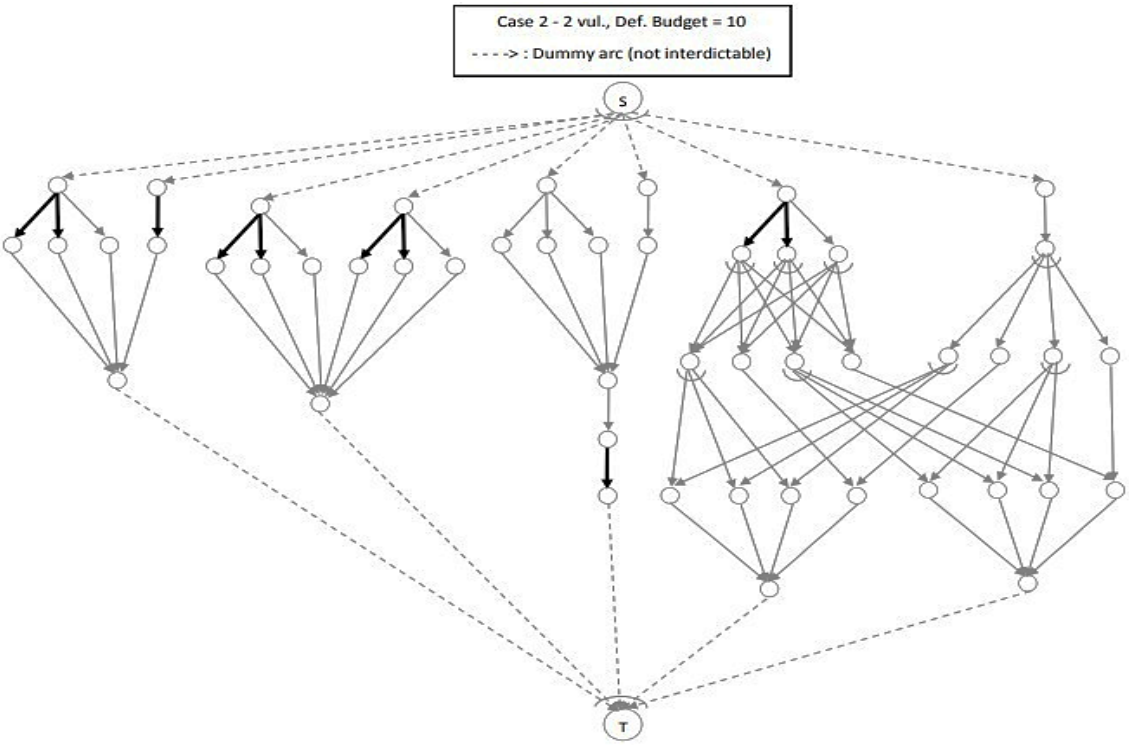

Figure 9. Illustration of the optimal defense strategy for case 2 with 2 vulnerabilities for a defender budget of 10 arcs

Overall, we were able to determine the optimal defense against a non-myopic attacker. Our method may not be directly applicable in practice, because the optimal defense is computationally intensive to find, and in many cases, is not significantly better than the defense against a myopic attacker. While this assertion holds a degree of validity, such findings undoubtedly offer valuable insights for the formulation of forthcoming defense strategies and policies. Cases where the benefit of defense against a non-myopic attacker is large may tend to involve large networks and may therefore be even more difficult to solve. Thus, while the goal of defending against a non-myopic attacker is significant, the methods we have used so far limit the applicability of our model. Thus, in future work, we plan to look for modeling and solution approaches that would help to solve our problem more efficiently even for large networks, such as dynamic programming.

Another important extension of this study involves transitioning from the idealized premise of a perfect information game to one that acknowledges imperfections in information. One prevalent avenue for addressing imperfect information games often involves adopting a worst-case scenario approach. Yet, employing such a method could prove exorbitantly expensive, notably within the domain of cyber defense. Consequently, the defender ought to explore more resource-efficient methodologies. Under this circumstance, enhancing the defender's defense strategies necessitates a broader spectrum of tactics, anticipating diverse attacker types and actions, thereby facilitating the development of a comprehensive defense strategy.

Despite these limitations, our results nonetheless yielded some recommendations of practical significance for the defense of cyber networks. First, for networks that are not well defended (i.e. highly vulnerable), defending against a myopic attacker may be adequate; the advantage of assuming a non-myopic attacker is greatest for well-defended networks. Similarly, defending against a myopic attacker may be adequate when the available attack paths share few common arcs, and when the success probabilities of attacks on different arcs vary widely. Thus, while the problem of defending against non-myopic attackers is difficult, it is encouraging to know that defense against a myopic attacker will frequently give results that are equally or almost equally good. Finally, our results suggest that the optimal defense will usually be to protect arcs near the source node or end node of the attack graph, rather than at intermediate layers of the attack graph.

REFERENCES

[1] Cybersecurity & Infrastructure Security Agency (CISA). (2022). "Annual Cybersecurity Report." https://www.cisa.gov/publications-library. Accessed 7 November 2023.
[2] Schneier B (2000) Secrets & Lies: Digital Security in a Networked World. 2nd ed. Wiley.
[3] Carin L, Cybenko G, Hughes J (2008) Cybersecurity strategies: The queries methodology. IEEE Computer 41(8) 20–26.
[4] Bier VM, Cox LA Jr., Azaiez MN (2009) Chap. 1: Why both game theory and reliability theory are important in defending infrasutructure against intelligent attacks. In Bier VM and Azaiez MN (ed) Game Theoretic Risk Analysis of Security Threats. Springer, 1–11.
[5] Sheyner, O, Haines J, Jha S, Lippmann R, Wing JM (2002) Automated generation and analysis of attack graphs. Proceedings of the IEEE Symposium on Security and Privacy. IEEE, 273–284.
[6] Lippmann RP, Ingols KW (2005) An annotated review of past papers on attack graphs. Tech. Rep. No. PR-IA-1, MIT Lincoln Lab, Lexington, MA.
[7] Liu P, Zang W, Yu M (2005) Incentive-based modeling and inference of attacker intent, objectives, and strategies. ACM Transactions on Information and System Security (TISSEC) 8(1) 78–118.
[8] Lye K, Wing JM (2005) Game strategies in network security. International Journal of Information Security 4(1) 71–86.

[9] Xiaolin C, Xiaobin T, Yong Z, Hongsheng X (2008) A Markov game theory-based risk assessment model for network information system. International Conference on Computer Science and Software Engineering, vol. 3. IEEE, 1057–1061.
[10] Nguyen KC, Alpcan T, Basar T (2009) Stochastic games for security in networks with interdependent nodes. International Conference on Game Theory for Networks, IEEE, 697–703.
[11] McMasters AW, Mustin TM (1970) Optimal interdiction of a supply network. Naval Research Logistics Quarterly 17(3) 261–268.
[12] Steinrauf, RL (1991) Network interdiction models. Master's thesis, Naval Postgraduate School, Monterey, CA.
[13] Phillips CA (1992) The network destruction problem. Tech. Rep. No. SAND-92-0186C, Sandia National Labs., Albuquerque, NM.
[14] Wood RK (1993) Deterministic network interdiction. Mathematical and Computer Modelling 17(2) 1–18.
[15] Fulkerson DR, Harding GC (1977) Maximizing the minimum source-sink path subject to a budget constraint. Mathematical Programming 13(1) 116–118.
[16] Golden B (1978) A problem in network interdiction. Naval Research Logistics Quarterly 25(4) 711–713.
[17] Cormican KJ, Morton DP, Wood RK (1998) Stochastic network interdiction. Operations Research 46(2) 184–197.
[18] Israeli E, Wood RK (2002) Shortest-path network interdiction. Networks 40(2) 97–111.
[19] Bayrak H, Bailey MD (2008) Shortest path network interdiction with asymmetric information. Networks 52(3) 133–140.
[20] Pan F, Morton DP (2008) Minimizing a stochastic maximum-reliability path. Networks 52(3) 111–119.
[21] Dimitrov NB, Morton DP (2013) Interdiction models and applications. Handbook of Operations Research for Homeland Security, Herrmann, J.W., Editor. Springer, 73–103.
[22] Morton DP (2011) Stochastic network interdiction. Wiley Encyclopedia of Operations Research and Management Science, Cochran, J. J., Editor. John Wiley & Sons, Inc.
[23] Cormican KJ (1995) Computational methods for deterministic and stochastic network interdiction problems. Master's thesis, Naval Postgraduate School, Monterey, CA.
[24] Held H, Hemmecke R, Woodruff DL (2005) A decomposition algorithm applied to planning the interdiction of stochastic networks. Naval Research Logistics 52(4) 321–328.
[25] Kall P, Wallace S (1994) Stochastic Programming. Wiley.
[26] Janjarassuk U, Linderoth J (2008) Reformulation and sampling to solve a stochastic network interdiction problem. Networks 52(3) 120–132.
[27] Van S, Richard M, Wets R (1969) L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics 17(4) 638–663.
[28] Shapiro A (2003) Monte Carlo sampling methods. Handbook in Operations Research and Management Science, Shapiro, A. and Ruszczynski, A., editors., North-Holland, Amsterdam, 2003 10 353–425.
[29] Verweij B, Ahmed S, Kleywegt AJ, Nemhauser G, Shapiro A (2003) The sample average approximation method applied to stochastic routing problems: A computational study. Computational Optimization and Applications 24(2) 289–333.
[30] Santoso T, Ahmed S, Goetschalckx M, Shapiro A (2005) A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research 167(1) 96–115.
[31] Ahmed S, Shapiro A, Shapiro E (2002) The sample average approximation method for stochastic programs with integer recourse. Preprint is available at www.optimization-online.org.
[32] Linderoth J, Shapiro A, Wright S (2006) The empirical behavior of sampling methods for stochastic programming. Annals of Operations Research 142(1) 215–241.
[33] Shapiro A, Xu H (2008) Stochastic mathematical programs with equilibrium constraints, modelling, and sample average approximation. Optimization 57(3) 395–418.
[34] Ertem M, Bier VM (2021) A Stochastic Network-Interdiction Model for Cyber Security. 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2021, pp. 171-176, DOI: 10.1109/ISMSIT52890.2021.9604681.
[35] Gumus ZH, Floudas CA (2005) Global optimization of mixed integer bilevel programming problems. Computational Management Science 2(3) 181–212.
[36] Bertsimas DJ, Jaillet P, Odoni AR (1990) A priori optimization. Operations Research 38(6) 1019–1033.
[37] Lageweg BJ, Lenstra JK, Rinnooy Kan AHG, Stougie L (1985) Stochastic integer programming by dynamic programming. Statistica Neerlandica 39(2) 97–113.
[38] Laporte G, Louveaux F, Mercure H (1992) The vehicle routing problem with stochastic travel times. Transportation Science 26(3) 161–170.
[39] Psaraftis HN (1984) On the practical importance of asymptotic optimality in certain heuristic algorithms. Networks 14(4) 587–596.

[40] Laporte G, Louveaux FV (1993) The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters 13(3) 133–142.
[41] Lippmann RP, Ingols KW, Scott C, Piwowarski K, Kratkiewicz KJ, Artz M, Cunningham RK (2005) Evaluating and strengthening enterprise network security using attack graphs. Tech. Rep. ESC-TR-2005-064, MIT Lincoln Laboratory, Lexington, MA.
[42] Lippmann RP, Ingols KW, Scott C, Piwowarski K, Kratkiewicz KJ, Artz M, Cunningham RK (2006) Validating and restoring defense in depth using attack graphs. Military Communications Conference, MILCOM 2006. IEEE, 1–10.
[43] Ou X, Govindavajhala S, Appel AW (2005) MulVal: A logic-based network security analyzer. 14th USENIX Security Symposium. 1–16.
[44] Mell P, Scarfone K, Romanosky S (2006) Common vulnerability scoring system. Security & Privacy, IEEE 4(6) 85–89.
[45] NIST (2013) National Vulnerability Database. URL http://nvd.nist.gov/. Accessed 7 November 2023.