國立臺灣大學管理學院資訊管理研究所

碩士論文

Graduate Institute of Information Management

College of Management

National Taiwan University

Master Thesis

考慮智慧型攻擊者經驗累積及權限提升下

網路強韌性之最大化

Maximization of Network Robustness Considering the Effect of Escalation and Accumulated Experience of Intelligent Attackers

陳奐廷

Huan-Ting Chen

指導教授：林永松 博士

Advisor: Yeong-Sung Lin, Ph.D.

中華民國 97 年 7 月

July, 2008

考慮智慧型攻擊者經驗累積及權限提升下

網路強韌性之最大化

Maximization of Network Robustness Considering the Effect of Escalation and Accumulated Experience of Intelligent Attackers

本論文係提交國立台灣大學
資訊管理學研究所作為完成碩士
學位所需條件之一部份

研究生：陳奐廷　撰

中華民國九十七年七月

國立臺灣大學碩士學位論文
口試委員會審定書

考慮智慧型攻擊者經驗累積及權限提升下
網路強韌性之最大化

Maximization of Network Robustness Considering the Effect of Escalation and Accumulated Experience of Intelligent Attackers

本論文係 陳奐廷 君（學號 R95725012）在國立臺灣大學資訊管理學系、所完成之碩士學位論文，於民國九十七年七月十五日承下列考試委員審查通過及口試及格，特此證明

口試委員：

孫雅麗　林永松

所　　長：陳靜枝

謝誌

隨著兩年的研究所生涯過去，碩士論文的完成，人生同時邁入下一個階段。回想論文成形的一點一滴，過程中的酸甜苦辣，不斷探究問題，反覆精鍊修改，以求其完備無誤，我相信這樣的一個學習歷程，即便辛苦，卻能獲益良多，日後必能帶著微笑回想這一段經過。

這兩年中，非常感謝支持我並不斷打氣的家人及朋友；感謝我的父母陳立文先生和謝美蓉女士，讓我在愛與包容的自由環境下，走過這一段學習過程；而在這求學過程中，最要感謝的是林永松老師不厭其煩的教導，老師總是給我們恣意揮灑的自由空間，更總在我們困惑的時候，適時的點醒我們，給予我們許多的指引、教誨以及鼓勵；此外也感謝本系孫雅麗教授、莊裕澤教授、清大通訊趙啟超教授、輔大資工呂俊賢教授在口試時對這篇論文寶貴的建議以及指正，讓這篇研究可以更完善。

感謝柏皓學長，從論文初步的構想，模型的建構，到最後投影片的製作，學長都給了我很多寶貴的建議，更給了我許多的方向，每每在不知所措的時候，總是學長的一番話，才能擁有繼續下去的動力與勇氣，可以在學長的建議與鼓勵下度過研究生涯，真的是很幸運的一件事！感謝霈語在整個研究中，和我討論了許多，解答了我諸多的問題，也給了我很多中肯的建議，尤其是英文上的協助，真的幫了我許多。感謝一起努力兩年，同為資安研究的至浩、孜謙、政祐，想想最後大家的奮鬥，我們終於成功的完成論文了！感謝俊維、岦毅與承賓學長提供的寶貴經驗與不管是生活上或是論文上的諸多幫助。感謝睿斌、竣韋、培維、猷順、冠瑋、宴毅及友仁，因為有你們，讓我可以無後顧之憂的全心準備口試。

感謝明芳總在過程中，給予我很多的鼓勵以及建議，讓我在每次心煩意亂的時候，可以整理好自己思緒，重新回到軌道，繼續奮戰！感謝奕仔與立穎總是聽我抱怨一堆，聽了我滿肚子的苦水，讓我可以平靜心情，大步向前。

最後感謝在這兩年中，曾經鼓勵過、幫助過我的所有人，也非常感謝能有機會進入這裡學習，因為有這一切，我才能走過這充實，也是難忘的兩年，要感謝的真的太多太多了，那就謝天吧！！謝謝老天讓我有一群替我加油的朋友，陪我走過這兩年。

陳奐廷 謹識
于台大資訊管理研究所
民國九十七年七月

論文摘要

論文題目：考慮智慧型攻擊者權限提升及經驗累積下網路強韌性之最大化

作者：陳奐廷　　　　　　　　　　　　　　　　　　九十七年七月

指導教授：林永松 博士

網路的日益普及，帶來了日常生活上的便利，卻也伴隨而來更多的網路犯罪，因此網路安全及其強韌性之衡量已逐漸受到重視；對網路營運者而言，如何能有效的評估攻擊者行為及威脅也已日趨重要。

在本篇論文中，我們提出一個兩階的數學規劃模型來描繪網路攻防情境以及攻擊者行為；其中內層問題，我們探討攻擊者欲利用最小攻擊成本來攻克網路上多個核心節點，而在其攻擊過程中，會不斷的累積攻擊經驗，使未來的攻擊成本有效的降低；此外，在攻擊者攻克某一節點後，亦可在此節點上進行權限提昇，如此攻擊者便可擁有足夠的權限來探測更多此節點上的資訊；在此，亦衡量這些資訊可能會對網路所造成的影響，亦即，攻擊者在攻克多個核心節點時，會同時讓這些資訊所造成的影響，達到一定程度的傷害；而在外層問題中，目標網路的管理者則能有效配置其有限防禦資源，使攻擊者需花費的攻擊成本最大化。為了求得此問題的最佳解，我們採用以模擬退火法為基礎的演算法來處理此問題，並設計出多種不同的初始解以及尋找鄰近解的方法，藉此獲得近似最佳解。

關鍵詞：網路攻防、存活度、多核心節點、權限提升、累積經驗、模擬退火法、最佳化

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT NATIONAL TAIWAN UNIVERSITY

NAME：HUAN-TING CHEN　MONTH/YEAR：JULY/2008

ADVISER：YEONG-SUNG LIN

Maximization of Network Robustness Considering the Effect of Escalation and Accumulated Experience of Intelligent Attackers

Internet has become much more important and worldwide, but it gives cyber criminals opportunities to crash a network system and conduct other cyber-crimes. Therefore, the issues of network security and robustness have come into notice. It is necessary for a network operator to understand the attacker behavior in order to efficiently allocate his limited budget.

In this thesis, we propose a two-level mathematical programming model to describe the network attack and defense scenario. In the inner problem, an attacker's objective is to compromise multiple core nodes using the minimum total attack cost. During the attack actions, the attacker may gain some experience from previous attacks to further reduce the attack costs in the future. Moreover, he can also pay extra fee to escalate on a compromised node to get higher user privileges, so that he will have higher authority to access more information on the node. We also measure the impact incurred by such information leakage in our model. As a result, the attacker will try to compromise multiple core nodes and collect valuable information, so that the total impact incurred by information leakage will exceed a threshold. Meanwhile, in the outer problem, the network operator of the target network allocates limited defense resources appropriately to maximize the total attack cost of the attacker. We adopt some Simulated Annealing-based algorithms to solve the problem and develop some initial solutions and several kinds of methods for searching neighbor solutions.

Key words: Network Attack and Defense, Survivability, Multiple Core Nodes, Escalation, Accumulated Experience, Simulated Annealing, Optimization.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

Internet has become worldwide and indispensable in our daily lives. It provides us with many convenient services, such as online chatting, video conference, file transfer, and E-commerce. However, it also brings us some threats. Cyber criminals can connect to others' computers to steal information and modify some important data via Internet. They can intrude vital systems and crash servers in the competitive corporations. Therefore, the research of network security is more and more important.

According to the CSI/FBI Computer Crime and Security Survey (2007) [1], the number that the respondents detected more than 10 incidents jumps from 9 to 26 percent in 2007. In addition, observing from Figure1-1 [1], we may find out the trend that organizations start using some technologies to protect their information systems. It may indicate that the network security issues have already come into notice. Since these network security incidents are often caused by attackers, if a network operator can understand the attacker behavior and the patterns of the attacks well, he may be able to maintain networks and resist malicious attacks more efficiently.

The issues of network security usually focus on the situation that whether systems are compromised or not. Therefore, the states of these issues were often defined as safe or compromised [2]. Because the information systems are usually in unbounded environments [3], the attackers may use the vulnerabilities of a system to reach the purpose that interrupts the service it provides. Thus, we start to pay more attention to how a system can sustain normal service under malicious attacks or random error conditions. As a result, the binary definition of network security is no longer sufficient describing the availability of information system nowadays. Thus, the concept of network security has been considered as the conditions of the availability of information service under malicious attacks and generalized as a subject of survivability [4, 5, 6] in recent years.

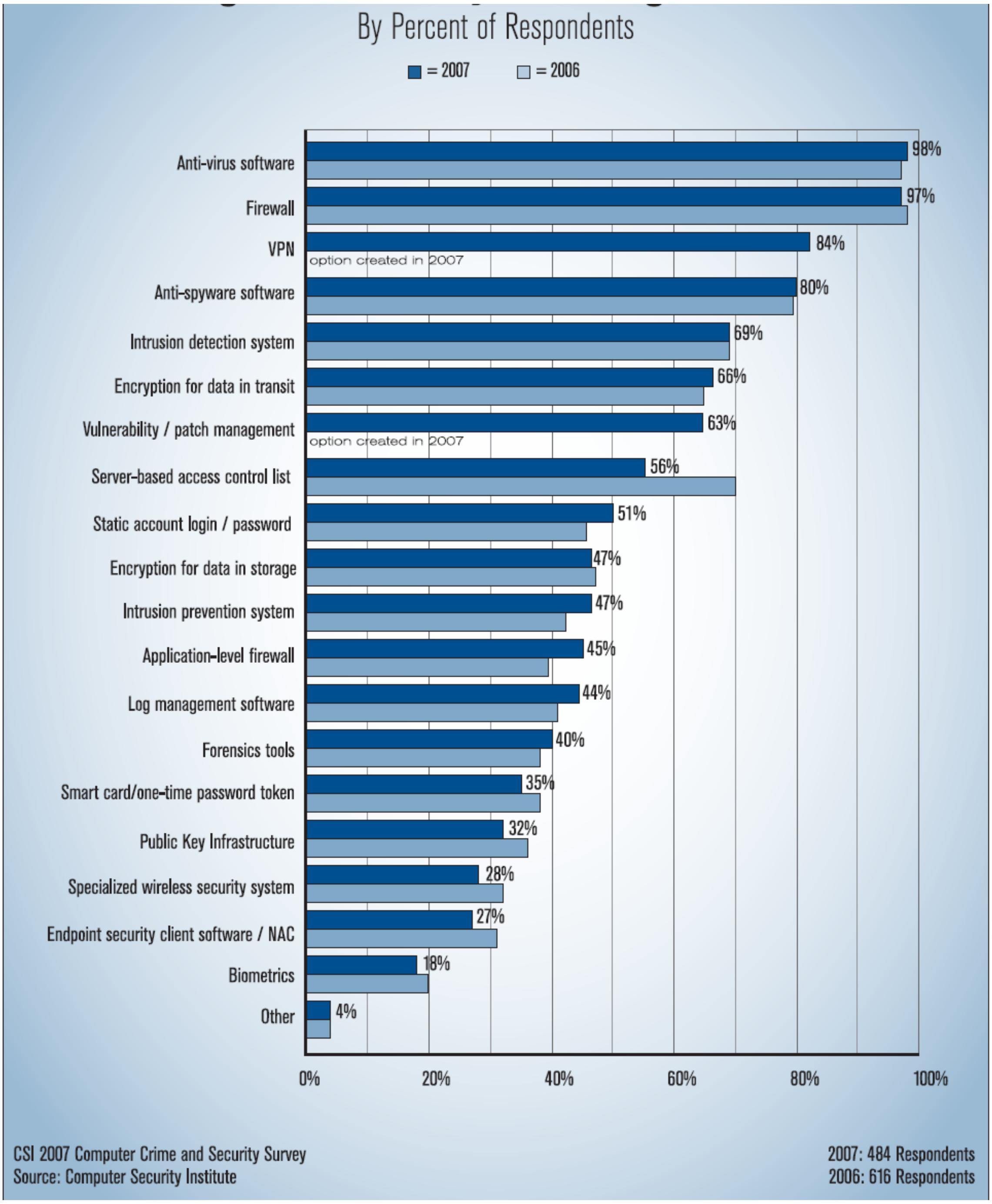

Figure 1-1 [1] Security Technologies Used

Although a great deal of research has been conducted on the field of survivability, unfortunately, the definitions of survivability are too diverse to unify into a general one. The most familiar definition proposed by Ellison et al. [7] is "*the capability of a system* *to fulfill its mission, in a timely manner, in the presence of attack, failures, or accidents.*" The more details about survivability will be discussed in Section 1.3.2.

1.2 Motivation

Since the Internet has become essential to business operations, more and more trades are accomplished through computer network. Thus, some critical data are usually stored in computers or other multimedia devices. Therefore, protecting these data is an important issue for enterprises. To ensure systems can continuously provide service or quickly recover from some disaster or malicious attacks, remote backup is a useful way. Remote backup means that organizations may regularly backup their critical data at different places. By applying this manner, the critical data of enterprises can be copied and stored in different centers, and all of these centers can provide the main service individually. It can avoid that all of the data destroy at the same time. Consequently, an attacker who wants to completely crash business service or networks may need to compromise all of these centers which have the capability of providing essential service. For this reason, the attacker's target may be multiple core nodes but single core node. For example, the company that the most quickly recovered from the 911 terrorist attacks is Morgan Stanley, a global financial service firm. That is because the company has a backup center at New Jersey.

Furthermore, in an attack scenario, the attacker may decide the best strategy to conduct a malicious attack. After compromising a node, the attacker can choose whether to probe this node or not. He can not only treat the compromised node as an intermediate node of the selected attack path, but also try to raise his control of the node in order to access more valuable information. For instance, the attacker may gain some useful information like the routing tables which can help him to get the whole picture of the network topology or help him to know more the outgoing and incoming links of the compromised node, which he did not know initially. Another example is that the attacker may access some vital information like the customer data of an E-commerce company or the secret information about business operations. When this kind of information is stolen, it not only causes some privacy issues but also leads to financial loss. For these reasons, to avoid the attacker enhancing his rights to access more information is very important.

In order to get more control of a compromised node, the attacker may need to conduct some escalation to enlarge his authority of the node. It means that the attacker can pay extra attack power to gain higher user rights/privileges. Once the attacker escalates to a higher level of user rights, he will be able to access more information on the compromised node. The higher level of user rights the attacker gains, the more costs he should pay. For example, operation systems, Windows, have different levels of user rights. More powerful user level represents that users at this level can do more tasks than he at a lower one, such as modify secret files and create new accounts. It also indicates that a higher level of user rights includes all the rights of lower levels. If the attacker gets a higher level of rights, he needs not to spend another budget getting a lower level of rights. Table 1-1 [8] lists several levels of user rights of Windows 2003.

While the attacker compromises a node, he may learn some experience from this attack. Besides, he can also gain another kind of experience from escalation. Using the experience efficiently he can reduce the attack costs of future attacks. For example, when the attacker compromises a node, he may learn how to intrude other systems via the same kind of vulnerabilities on the compromised node or he may be able to infer that other systems, which is near the compromised node in the same intranet of an enterprise, might use the same security mechanism with the compromised node, such as firewall or intrusion detection system. So the attacker can make use of this information to compromise another node more quickly and effectively, in other words, he can reduce the attack costs in the future.

Table 1-1 [8] Default user rights of Windows 2003

Group	**Description and Default user rights**
Administrators	**Description:** Members of this group have **full control of the server** and can **assign user rights and access control permissions to users** as necessary. The Administrator account is also a default member. When this server is joined to a domain, the Domain Admins group is automatically added to this group. Because this group has full control of the server, add users with caution
	Default user rights: Access this computer from the network; Adjust memory quotas for a process; **Allow log on locally**; Allow log on through Terminal Services; **Back up files and directories**; Bypass traverse checking; **Change the system time**; **Create a pagefile**; Debug programs; **Force shutdown from a remote system**; **Increase scheduling priority**; **Load and unload device drivers**; Manage auditing and security log; Modify firmware environment variables; Perform volume maintenance tasks; Profile single process; Profile system performance; **Remove computer from docking station**; **Restore files and directories**; **Shut down the system**; Take ownership of files or other objects.
Power Users	**Description:** Members of this group can **create user accounts and then modify and delete the accounts they have created**. They can **create local groups and then add or remove users from the local groups they have created**. They can also add or remove users from the Power Users, Users, and Guests groups. Members can **create shared resources and administer the shared resources they have created**. They **cannot take ownership of files, back up or restore directories**, load or unload device drivers, or manage security and auditing logs.
	Default user rights: Access this computer from the network; **Allow log on locally**; Bypass traverse checking; **Change the system time**; Profile single process; Remove computer from docking station; **Shut down the system**.
Users	**Description:** Members of this group can perform common tasks, such as **running applications, using local and network printers**, and locking the server. **Users cannot share directories or create local printers**. By default, the Domain Users, Authenticated Users, and Interactive groups are members of this group. Therefore, any user account created in the domain becomes a member of this group.
	Default user rights: Access this computer from the network; Allow log on locally; Bypass traverse checking.
Backup Operators	**Description:** Members of this group can back up and restore files on the server, regardless of any permissions that protect those files. This is because the right to perform a backup takes precedence over all file permissions. They cannot change security settings.
	Default user rights: Access this computer from the network; Allow log on locally; Back up files and directories; Bypass traverse checking; Restore files and directories; Shut down the system.
Guests	**Description:** Members of this group will have a **temporary profile created at log on, and when the member logs off, the profile will be deleted**. The Guest account (which is disabled by default) is also a default member of this group.
	Default user rights: No default user rights.

From a defender's view, while a business considers the network security issues, it has become to expand its strategies to risk management. That is because risk management not only measures the security events and the potential threats but also evaluates the values of asset and the impact incurred by network events. By applying such management, a business can more efficiently set up the defense strategies which protect some mission-critical systems from malicious attacks and random errors.

In addition, because the defender only has finite budget, it is important for him to allocate his budget efficiently. In order to decide the best defense budget allocation strategy, the defender must consider the best attack strategy the attacker would adopt. For these reasons, it is necessary for the defender to understand the attacker behavior. However, there are seldom theoretical studies modeling the attacker behavior and the offense-defense scenarios in mathematical ways [9]. Therefore, we propose a two-level mathematical model considering defense resource allocation strategy and attacker behavior in terms of risk management. It describes and formulates the attack-defense scenarios and provides the defender with defense strategies to allocate limited budget efficiently to maximize the network robustness.

1.3 Literature Survey

In this section, we review some related works about risk management, survivability, and attacker behavior.

1.3.1 Risk Management

Nowadays, how to efficiently allocate limited budget to the assets in the purpose of defending some critical resources has raised more notices at organizations and society. Therefore, risk management has become an important issue recently. Risk management should be concerned with a series of process which describes the relationship with the potential threats of particular vulnerabilities, the values of critical assets and the impact or damage incurred by losing such assets under some events.

Therefore, we discuss the vulnerabilities on a system first. An attacker usually exploits unpatched vulnerabilities to intrude a system and he can also use vulnerabilities to escalate from a lower privilege level to a higher privilege level [10]. In [11], the author used a quantitative evaluation of risk reduction estimation on a system. They proposed a compromise graphic where each node represents an attack event and each edge represents the expected time an attacker would need to gain some level of privilege on the corresponding system device. The expected time is modeled as a function of different types of vulnerabilities, the number of vulnerabilities and the attacker skill level. This research also indicated that the number of vulnerabilities which are exploitable externally would influence the risk level of a system. Similarly, according to [12], the author measured the effect of vulnerabilities by the total number of them on a system and proposed a time-based model to quantitatively evaluate the relationship between the time of discovering vulnerabilities and the number of vulnerabilities. Therefore, managing and patching vulnerabilities well would be an important mission in computer security. In [13], the author summarized patch management of system vulnerabilities and proposed a model of vulnerability life cycle based on response time and related risk level. Besides, as shows in Figure 1-2 [13], it pointed out that the number of reported vulnerabilities has increased quickly in the recent years. In accordance with [13], it takes about 5.54 days on average to patch a vulnerability which has been evaluated its importance. It also indicated that if we manage the vulnerabilities on a system well, we can reduce the costs and risk level of this system and increase system availability. Thus, we use the number of vulnerabilities on a system as our measurement metric.

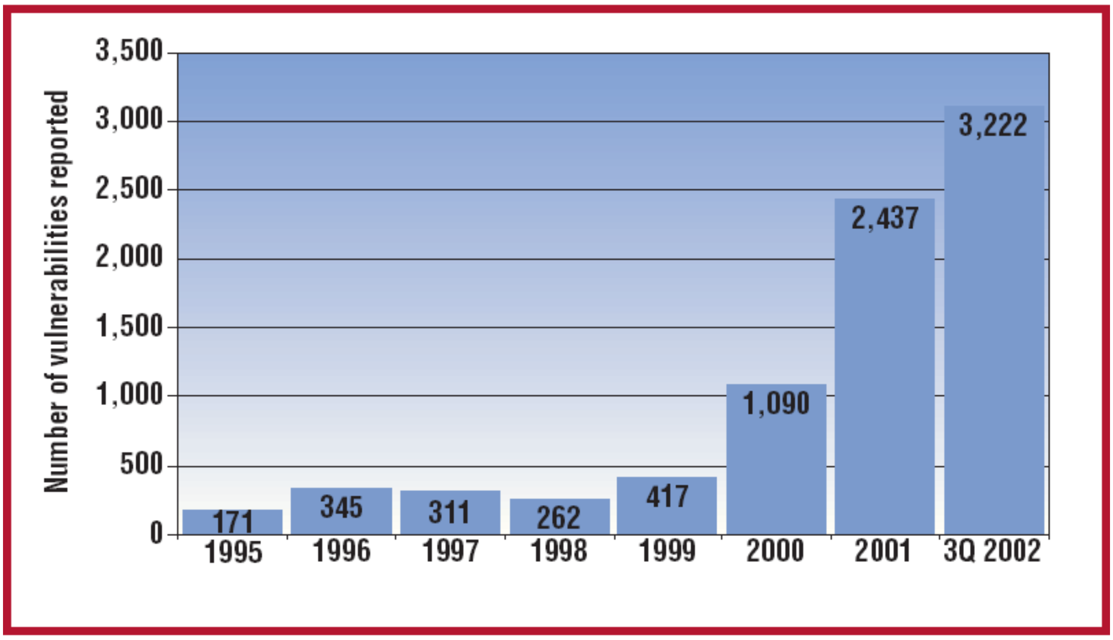

Figure 1-2 [13] Number of vulnerabilities reported from 1995 to 2002

Next, we consider some risk management approaches. In [14], the risk management is measured by a mixed qualitative and quantitative approach in order to describe attack scenarios and evaluate the risk of a system accurately. They analyzed the attacker behavior by using the defense trees which are modified from the attack trees. Traditionally, the attack tree is used to describe the attack strategy which is represented by the relationships among the vulnerabilities, the goal of an attack, the subgoal of this attack process, and the probability associated with each attack. The subgoal means the intermediate process of an attack, i.e., the attacker would need to combine different types of attacks to achieve the attack goal. Therefore, the defense tree is an extension of the attack tree with some countermeasures against different attacks. They also use some quantitative indexes to compute the return on investment value (ROI) for defenders and the return on attack value (ROA) for attackers. This approach can be useful for the defender to understand the efficacy of each countermeasure and provide information for the defender to make a more efficient decision during the risk management process.

Similarly, a different method to evaluate the ROI and other risk metrics is proposed in [15]. In accordance with [15], the impact of an attack to the critical network asset is evaluated by money lost in some network events. These kinds of quantitative approaches, which measure the economic impact of security risk, are also presented in other researches such as [16].

1.3.2 Network Survivability

Since the robustness of a network cannot be clearly and definitely measured, we use a quantification method, survivability, to evaluate it. Thus, the higher survivability means the higher robustness of networks.

Although there is no consistent definition of the concept of survivability, it still can be generalized as a context-specific one proposed by Westmark [17], which is "*the ability of a given system with a given intended usage to provide a pre-specified minimum level of service in the event of one or more pre-specified threats.*" In addition, Westmark also generalized measurements of survivability as three major categories: connectivity, network performance, and a function of other quality or cost measures.

Thus, the definitions of survivability under other considerations are listed as Table1-2.

Table 1-2 Survivability Definition Summary

Researcher	Definition	Year
Knight and Sullivan [4]	A system that has the ability to continue to provide service (possibly degraded or different) in a given operating environment when various events cause major damage to the system or its operating environment.	2000
Ellison, Fisher, and Linger [2]	Survivability is the capability of a system of a system to fulfill its mission, in a timely manner, in the presence of attack, failures, or accidents.	1999
Liew and Lu [18]	If the selected feature of network is quantified by x, survivability S is measured by the fraction of x that remains after an instance of the disaster type under consideration has happened.	1992
Westmark [17]	Survivability is the ability of a given system with a given intended usage to provide a pre-specified minimum level of service in the event of one or more pre-specified threats.	2004

As it was mentioned above, the concept of survivability can be summarized as the availability of a system that is under abnormal conditions, i.e., an ideal survivable system should provide continuous service after some malicious attacks or random errors.

Because higher risk level usually means lower survivability level and in order to evaluate risk level of the network, we consider survivability using the point of view of risk management in our research and measure the impact factor by the loss and damage resulting from information leakage. Therefore, we use the minimized total attack cots as our evaluation metric to measure the network survivability and robustness.

1.3.3 Attacker Behavior and Privilege Escalation

In order to set up the defense strategy, it is important for a network operator to understand the attacker behavior well. In [19], the author conducted some intrusion experiment to get empirical data. By analyzing the collected data, the author split the intrusion process into three phases based on attacker behavior: the learning phase, the standard attack phase, and the innovation phase. They pointed out that the most of the breaches were occurred at the standard attack phase, and the statistical test on the collected data also indicated that the times between consecutive breaches during standard attack phase are distributed exponentially. Similarly, McDermott et al. [20] pointed out that potential intelligent intruders will more probably attack the target as time goes by. Therefore, the intruder can not compromise the target today may be more likely to compromise the target in the future.

The attacker behavior may be a series of intrusions, in other words, to reach the goal the attacker must perform different types of attacks to collect the information that he needs. In [10], the author analyzed the attack behavior and found some characters: an attacker in the low user-level may usually exploit several vulnerabilities on a computer system to get a certain privilege escalation, and then he will be able to reach the high user-level without authorization. They also indicated that the attacker at a certain user-level owned the corresponding user privileges and resources of that system. Similar issues can be found on other research. According to [21], while the attacker successfully compromised a system, he would like to continuously perform other attacks to gain root privileges so that he can access the system resources which he is really interested in. Similarly, McQueen et al. [11] also indicated that attacks can be divided into several parts. One of them is escalation on a machine via the corresponding vulnerabilities. Once the attacker escalates to the root level, he can gain all other privileges. In [22], the author combined the evaluation of the potential risk and user privileges. They pointed out that the typical goal of the attacker is to change the security configuration from State A to desired State B, where State B is an escalated level and has more privileges than

State A. Thus, they proposed a Risk Potential Formulation that measured the potential risk on an operating system by quantify four metrics. According to the formulation, there are a direct proportional relationship with the number of privileges the user can gain at the new access level (state) and an inverse proportional relationship with the total number of rights the user can acquire at the original level. As a result, we define the process of an attacker raising his user rights and gaining more information as a term, escalation, in this research.

1.4 Proposed Approach

We propose a max-min mathematical programming model to describe the Defense Resource Allocation problem (DRA) and the attack strategy problem in which we consider the effect of Accumulated Experience and Escalation of attackers (AEE) in a quantitative way.

In the DRA Model, we formulate that a network defender would try to find the best defense resource allocation strategy to maximize the total attack cost. The more costs an attacker pays, the more robust the network is. In the AEE Model which is the inner problem of the DRA Model, we assume that the attacker may accumulate attack experience and escalate on each compromised node. Thus, the attacker's objective is to compromise multiple core nodes and minimize the total attack cost, which includes the cost of compromising nodes and escalating on compromised nodes. We also use a term, *impact*, to evaluate the total loss incurred by information leakage. We apply the Simulated Annealing method to solve this problem.

1.5 Thesis Organization

The remainder of this thesis is organized as follow. In Chapter 2, a max-min mathematical formulation of the defense-attack scenario is proposed. In Chapter 3, solution approaches based on the Simulated Annealing methods are presented. The computational results of the problem are showed in Chapter 4. Finally, Chapter 5 is the conclusions and future work about this research.

Chapter 2 Problem Formulation of the DRA and AEE Models

In this chapter, we propose a two-level mathematical model with specific assumptions and problem objective. In this Model, we consider that a network defender would allocate resources appropriately to defend networks so that an attacker would need to pay more costs to compromise the target network. The attacker's objective is to compromise multiple core nodes in the given network and to minimize the total attack cost as possibly as he could. In addition, he may gain experience from his previous attacks to reduce the costs of the future attacks. During the attack actions, the attacker may not only compromise a node but also pay an extra fee to conduct some escalation on the compromised node to get more powerful user rights, so that he is able to access more useful information (e.g., routing tables or the network's topology) to further reduce the costs of attacks and accumulate impact incurred by information leakage. As a result, while the attacker decides his attack strategy, the network defender would adjust the resource allocation strategy again to resist the attacks. In response, the attack will change his strategy again to find the best policy. Thus, it is an interaction between the defender and the attacker.

2.1 Problem Description and Assumption of the DRA Model

The problem we consider here is how a defender can use his resources efficiently to resist an attacker to compromise all the core nodes in the Autonomous System (AS) level network under the consideration that the attack costs might be reduced by accumulated experience and the loss incurred by information leakage must be greater than a given threshold. By adjusting defense strategies continuously in the battle with the attacker, the defender could finally find the best strategy to defend the networks and enhance the network robustness. Thus, in order to defend the network efficiently, the defender may need to understand the attacker behavior well.

At the AS level, a network domain is represented by a node and an inter-domain connection is represented by an edge. To reach a destination node, the attacker must find a path from the source node to the destination node and compromise all the intermediate nodes on the path. That is, the attacker needs to choose an attack path for each core node.

Because the number of vulnerabilities on each node is different, an attacker who wants to compromise a node may need to pay different costs related to the defense budget allocated to the node and the vulnerabilities on it. Besides, the attacker will gain experience from compromising a node which could further reduce the attack costs, in other words, he can make use of the accumulated experience to compromise other nodes more efficiently.

Moreover, we also consider a more realistic situation. When the attacker launch attacks on a computer system, he may not only treat the compromised node as an intermediate one but also use the node as efficiently as possible. For this reason, the attacker could pay extra costs to do some privilege escalation to access more information and get more control power of the node. Knowing this information, the attacker can understand the network topology more clearly and collect some information that would help him to know the security mechanisms of the same systems or collect the partial information to get a whole picture.

As it was mentioned before, while the attacker conducts some escalation on a compromised node, he may know more clearly about the network topology. In other words, he could be conscious of more the outgoing and incoming links of the node that he did not know initially. It indicates that the attacker can compromise farther nodes which he cannot reach before escalating on the node.

Because there are several levels of user privileges on a system, an attacker could pay various levels of extra budget to do different levels of escalation. The more costs he pays, the more user rights he could gain. Besides, the attacker may also gain some experience from escalation. This kind of experience can help the attacker to reduce the further escalation costs. That is, the attacker could gain two kinds of experience, one comes from compromising a node, and the other comes from escalating on a compromised node.

Considering the information an attacker access from a compromised node, if it contains some important financial data of an enterprise or some secret files, such as personnel data, or the password of a network administrator, it may cause critical loss of the network and the enterprise. For this reason, we also consider the information value corresponded to an impact factor to evaluate the damage incurred by information leakage in this model.

We describe the attack scenario in detail by Figures 2-1 to 2-4. Initially, the attacker starts on node *o* (Figure 2-1) and begins the attack actions by compromising nodes (Figure 2-2). After compromising a node, he can choose whether to escalate or not (Figure 2-3). If he escalates to a higher privilege level, he might be able to access more information on the node, so that he could know some critical information or understand more the outgoing and incoming links of the node. Thus, he can use this information to attack other nodes more efficiently. Finally, for each core node, the attacker would find out an attack path and compromise all the intermediate nodes on the attack path, and then some impact related to information leakage will occurred during his attack actions. The attacker's objective is to compromise multiple core nodes using the minimum total attack cost under the consideration that the total impact must exceed a given threshold (Figure 2-4). Therefore, if the attacker exactly finds an attack path towards each core node and minimizes the total attack cost, all the attack paths will naturally join into an attack tree that consists of all the core nodes. During the attack procedure, the attacker may accumulate some experience about compromising nodes and escalation. Therefore, he can use this experience to reduce the further attack costs and escalation costs. Diagrams of the attack behavior are presented below.

Furthermore, the attack cost which the attacker needs to apply to each node to compromise it would depend on the budget allocated to it. That is, the more defense resources the defender allocates to a node, the more powerful defense capability the node has. Thus, how to distribute the defense resources effectively would be an important mission for the network operator.

Knowing the attacker behavior is helpful for a network operator to allocate limited defense budget, because an intelligent attacker will adjust his attack strategy to minimize the total attack cost, i.e., he will choose the best way to reach his goal. The process of defending against attackers is not static. The defender would allocate his budget to protect nodes and maximize the minimized total attack cost. In response, the attacker will search for another way to reach his goal based on the defense strategy. According to these reasons, the more attacker behavior the defender knows, the better strategy he can adopt. That is, know your enemy, know yourself.

For the reasons we discussed above, in this model we address the problem that the defender's objective is to maximize the minimized total attack cost of the attacker by allocating the defense resources well. Similarly, the attacker's objective is to compromise multiple core nodes using the minimum attack cost under the consideration that different attack sequence would result in different accumulate experience. Thus, the attacker may need to find the best strategy to decide which nodes, privilege levels and attack sequence he should adopt. Moreover, the robustness of a network will also be evaluated by the minimized attack cost in this problem. A more robust network means the attacker would need to pay more costs to compromise the target network. The description and assumptions of this model are given in Table 2-1 and Table 2-2.

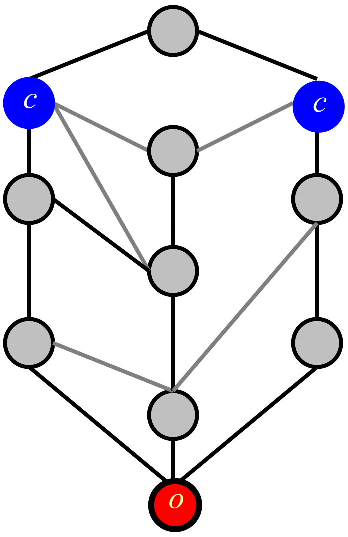

Figure 2-1 Initial State

The attacker is on node *o*, and *c* means the node is core node.

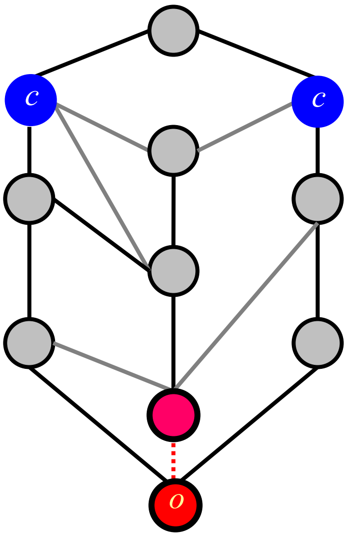

Figure 2-2 Attacking a Target

The attacker compromises a node successfully.

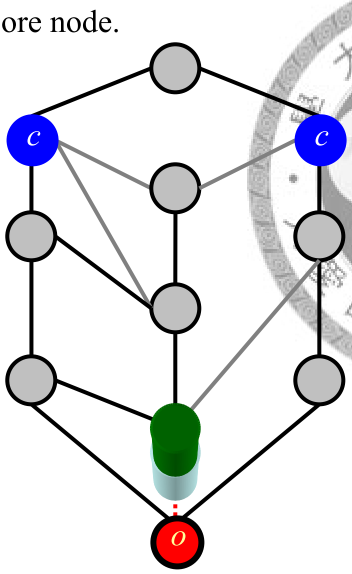

Figure 2-3 Escalation

The attacker escalates on the compromised node (nodes with multi-layers), and accesses information from it. After escalating, the attacker can know some links (gray lines) he did not know initially.

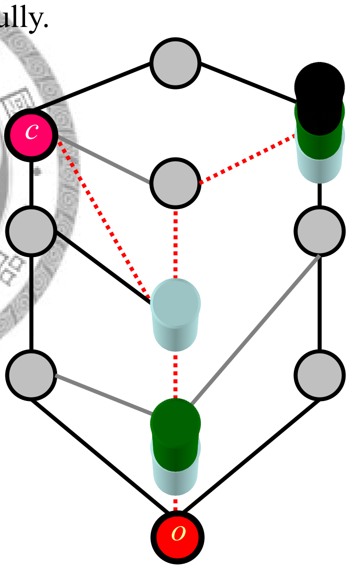

Figure 2-4 Successful Attack

Continuing the attack until the core nodes are compromised and the accumulated impact exceed the given threshold.

o	Attacker's initial position *o*	▮	The first level privilege
○	Uncompromised node	▮	The second level privilege
c	Core node	▮	The third level privilege
●	Compromised node	─	Reachable link
─	Link that is reachable only on some higher privilege levels	┈	Link on the attack path

Table 2-1 Problem Description

Given

- Core nodes
- The network topology and the network size
- The number of vulnerabilities on each node
- The experience the attacker could gain after compromising a node and/or escalating on a compromised node

Objective

- To maximize the minimized total attack cost

Subject to

- Only one node can be compromised at each stage
- The node to be attacked must be connected to the existing attack tree
- The total impact of the target network must be greater than a given threshold
- The sum of the defense budget allocated to protect nodes from being compromising and escalating must be no more than total defense budget

To determine

Defender：

- Budget allocation strategy

Attacker：

- Which nodes to attack
- Which levels to escalate on a compromised node
- Attack sequence

Table 2-2 Problem Assumption

Assumption

- The defender has complete information about the network.
- The defender has budget limitations.
- The attacker is on node *o*.
- There are some core nodes in the network which are the targets of the attacker.
- A node is subject to attack only if a path exists from node *o* to that node, and all the intermediate nodes on the path have been compromised (they can be viewed as hop sites for attacking the targets).
- Only nodal attacks are considered.
- Only malicious attacks are considered.
- The target network is at AS-level.
- The attack cost of a node is affected by the number of vulnerabilities on that node and the defense budget allocated to it.
- A node is compromised if the attack budget applied to the node is equal to or more than the defense capability of the node.
- After compromising a node, the attacker can pay extra attack budget to escalate to a higher privilege level, so that he can access more valuable information which may cause additional damage to the target network.
- The attacker can access different levels of information after obtaining different

levels of privileges.

- The higher privilege level includes all the privileges the attacker could gain at other lower levels on the same node.
- The higher privilege level the attacker tries to escalate to, the more budget he should pay.
- Total impact is measured by the sum of the damage incurred by information leakage after the attacker probing a compromised node.
- Total attack cost is the sum of the cost of compromising a node and the cost of escalating on a compromised node.
- The attacker gains and accumulates experience from compromising a node and/or from escalating on a compromised node to further reduce the costs of future attacks.

2.2 Problem Formulation of the DRA Model

We model the above problem as a max-min mathematical programming problem. The given parameters are defined as Table 2-3.

Table 2-3 Given Parameters of the DRA Model

Given Parameters	
Notation	**Description**
N	The index set of all nodes in the network
D	The index set of all core nodes in the network
W	The index set of all Origin-Destination pairs (O-D pairs), where the origin is node o; and the core nodes are d (where $d\in D$)
E_i	The index set of all the privilege levels on node i (e.g., 0, 1, 2, …), where $i\in N$ and level 0 means node i is compromised without escalation.
L_i	The index set of all the level on node i exclusive of level 0, where $i\in N$
P_w	The index set of all candidate paths of an O-D pair w, where $w\in W$
δ_{pijl}	An indicator function, which is 1 if node j (at privilege level l) is the pervious node of node i on path p, and 0 otherwise (where $i, j\in N$, $p\in P_w$, $l\in E_i$)

σ_{pil}	An indicator function, which is 1 if of node *i* (at privilege level *l*) is on path *p*, and 0 otherwise (where $p \in P_w$, $i \in N$, $l \in E_i$)
S	The index set of all stages
$S_{(k)}$	The index set of the stage 1 to stage *k*-1, where $k \in S$
e_{il}^{e}	The experience gained by the attacker after escalating to level *l* on node *i*, where $i \in N$, $l \in E_i$
e_{i}^{c}	The experience gained by the attacker after compromising node *i*, where $i \in N$
I_{il}	The impact incurred by accessing information from level *l* on node *i* after being escalated, where $i \in N$, $l \in E_i$
T	The threshold of total impact, which is the damage level that the attacker needs to reach.
B	The total defense budget

We will focus on which nodes and levels would be compromised and which attack order the attacker would adopt. In this formulation, the attack sequence is represented by a term, stage. Stage *n* means the attack is launched on the *n*-th step of the attack action. In other words, if there are *m* nodes in the network, we need at most *m* stages to represent the entire attack action. We define a set *S* to stand for these stages. The attacker's objective is to minimize the total attack cost by deciding which node and level should be attack at each stage. As noted earlier, once the attacker escalates to a higher level on a compromised node, he might know the network topology more, i.e., he might know some links he did not know before escalating to the level. Thus, the network we modeled here can be viewed as an artificial two-dimensional network. Higher levels on each node may have more links connecting to other nodes.

The decision variables are defined as Table 2-4.

Table 2-4 Decision Variables of the DRA Model

Decision Variables	
Notation	**Description**
y_{sil}	1 if node *i* is compromised at stage *s* and escalated to level *l* of the node, and 0 otherwise (where $s\in S$, $i\in N$, $l\in E_i$)
x_p	1 if path *p* is selected as the attack path, and 0 otherwise (where $p\in P_w$)
b_i^c	The defense budget allocated to protect node *i* from being compromised, where $i\in N$
b_{il}^e	The defense budget allocated to protect node *i* from being escalated, where $i\in N$, $l\in L_i$
$\hat{a}_i^c(b_i^c)$	The threshold of the attack budget required to compromise node *i*, where $i\in N$
$\hat{a}_{il}^e(b_{il}^e)$	The threshold of the attack budget required to escalate to level *l* on node *i*, where $i\in N$, $l\in L_i$

Our proposed model is as follows.

Objective:

$$\max_{b_i^c, b_{il}^e} \min_{y_{sil}, x_p} \left(\sum_{i\in N, l\in E_i} \hat{a}_i^c(b_i^c) \sum_{k\in S} y_{kil} \prod_{s\in S_{(k)}} \sum_{j\in N} \sum_{m\in E_i} \left(e_j^c y_{sjm} \right) + \sum_{i\in N, l\in L_i} \hat{a}_{il}^e(b_{il}^e) \sum_{k\in S} y_{kil} \prod_{s\in S_{(k)}} \sum_{j\in N} \sum_{m\in L_i} \left(e_{jl}^e y_{sjm} \right) \right) \quad \text{(IP 1)}$$

Subject to:

$$\sum_{p\in P_w} x_p \sigma_{pil} \le \sum_{s\in S} y_{sil} \qquad \forall\, i\in N, w\in W, l\in E_i \qquad \text{(IP 1.1)}$$

$$\sum_{p\in P_w} x_p = 1 \qquad \forall\, w\in W \qquad \text{(IP 1.2)}$$

$$x_p = 0\ or\ 1 \qquad \forall\, p\in P_w, w\in W \qquad \text{(IP 1.3)}$$

$$\sum_{l\in E_i} y_{kil} \le \sum_{s\in S_{(k)}} \sum_{j\in N} \sum_{l\in E_i} y_{sjl} \delta_{pijl} \qquad \forall\, i\in N, k\in S, p\in P_w, w\in W \qquad \text{(IP 1.4)}$$

$$\sum_{s\in S}\sum_{l\in E_i} y_{sil} \le 1 \qquad \forall\, i \in N \qquad \text{(IP 1.5)}$$

$$\sum_{i\in N}\sum_{l\in E_i} y_{sil} \le 1 \qquad \forall\, s \in S \qquad \text{(IP 1.6)}$$

$$y_{sil} = 0\; or\; 1 \qquad \forall\, i \in N, s \in S, l \in E_i \qquad \text{(IP 1.7)}$$

$$T \;\le \sum_{i\in N}\sum_{l\in E_i}\sum_{s\in S} I_{il}\, y_{sil} \qquad \text{(IP 1.8)}$$

$$0 \le \sum_{i\in N}\left(b_i^c + \sum_{l\in L_i} b_{il}^e \right) \le B\,. \qquad \text{(IP 1.9)}$$

Explanation of the mathematical formulation:

- ✓ **Objective function:** The objective is to maximize the minimized attack cost by adjusting the defense budget allocated to each node. In the inner problem, an attacker tries to compromise multiple core nodes using the minimized total attack cost, which includes the total compromised cost $\sum_{i\in N, l\in E_i}\left(\hat{a}_i^c(b_i^c,\ V_i^c)\sum_{k\in S} y_{kil} \prod_{s\in S_{(k)}}\sum_{j\in N}\sum_{m\in E_i}\left(e_j^c y_{sjm}\right)\right)$ and the total escalation cost $\sum_{i\in N, l\in L_i}\left(\hat{a}_{il}^e(b_i^e, V_i^e)\sum_{k\in S} y_{kil} \prod_{s\in S_{(k)}}\sum_{j\in N}\sum_{m\in L_i}\left(e_{jl}^e y_{sjm}\right)\right)$, by deciding which nodes and levels to escalate and which attack sequence to adopt, i.e., deciding the y_{sij} value of each node at each stage. The compromised costs and escalation costs would be reduced by experience factor, e_i^c and e_{il}^e, which are values between 0 and 1. The effect of the experience would be showed as accumulated multiplied forms,

$\prod_{s\in S_{(k)}}\sum_{j\in N}\sum_{m\in E_i}\left(e_j^c y_{sjm}\right)$ and $\prod_{s\in S_{(k)}}\sum_{j\in N}\sum_{m\in L_i}\left(e_{jl}^e y_{sjm}\right)$.

✓ **Constraint (IP 1.1)** requires that if a node is on the selected attack path, it must be compromised at one stage by the attacker, i.e., $\sum_{s\in S} y_{sil} = 1$.

✓ **Constraints (IP 1.2)** and **(IP 1.3)** enforce that there should be one and only one attack path for each core node.

✓ **Constraint (IP 1.4)** requires that if a node is compromised at stage *k*, i.e., $\sum_{l\in E_i} y_{kil} = 1$, the ancestor node of that node on the selected attack path must have been compromised at one of the stages 1 to *k*-1 before. It enforces that the attacker must find a path between the source node and the current target node, in other words, the attack action must be in sequence.

✓ **Constraints (IP 1.5)**, **(IP 1.6)** and **(IP 1.7)** enforce that only one node could be compromised and only one level on the compromised node could be escalated by the attacker at each stage.

✓ **Constraint (IP 1.8)** requires that the sum of the impact incurred by information leakage on each compromised node should be greater than or equal to a given threshold *T*.

✓ **Constraint (IP 1.9)** restricts the sum of defense resources allocated to each node must not exceed the total defense budget *B*.

Since the attack tree is naturally constructed by joining each attack path which is towards each core node, and the attacker's objective is to minimize the total attack cost, we do not need to set up some constrains which would enforce the absence of a cycle on the attack tree in this formulation. If the attacker chooses some paths that would form cycles, the attack resources would be wasted on unnecessary nodal compromise, i.e., in this situation, there will be more than one way to reach some nodes, but it violates the attacker's minimum cost objective.

2.3 Problem Formulation of the AEE Model

Solving the proposed two-level mathematical problem is difficult because the attack strategy is unknown. Thus, we formulate attacker behavior as an optimization problem, the AEE Model, which is the inner problem of the DRA Model. According to this problem, we can get some information to simulate the future actions of the attacker and then we can develop the best defense strategy for network defenders. Hence, we will use the result of the AEE Model as the input of the DRA Model to solve this two-level problem.

The assumptions and attack scenarios of the AEE Model are the same with the DRA Model. We model the above problem as a mathematical programming problem which is the inner problem of the DRA Model. The given parameters are defined as Table 2-5.

Table 2-5 Given Parameters of the AEE Model

Given Parameters	
Notation	**Description**
N	The index set of all nodes in the network
D	The index set of all core nodes in the network
W	The index set of all Origin-Destination pairs (O-D pairs), where the origin is node o; and the core nodes are d (where $d\in D$)
E_i	The index set of all the privilege levels on node i (e.g., 0, 1, 2, …), where $i\in N$ and level 0 means node i is compromised without escalation.
L_i	The index set of all the level on node i exclusive of level 0, where $i\in N$
P_w	The index set of all candidate paths of an O-D pair w, where $w\in W$
δ_{pijl}	An indicator function, which is 1 if node j (at privilege level l) is the pervious node of node i on path p, and 0 otherwise (where $i, j\in N$, $p\in P_w$, $l\in E_i$)
σ_{pil}	An indicator function, which is 1 if of node i (at privilege level l) is on path p, and 0 otherwise (where $p\in P_w$, $i\in N$, $l\in E_i$)
S	The index set of all stages
$S_{(k)}$	The index set of the stage 1 to stage k-1, where $k\in S$
e_{il}^{e}	The experience gained by the attacker after escalating to level l on node i, where $i\in N$, $l\in E_i$
e_{i}^{c}	The experience gained by the attacker after compromising node i, where $i\in N$
I_{il}	The impact incurred by accessing information from level l on node i after being escalated, where $i\in N$, $l\in E_i$
T	The threshold of total impact, which is the damage level that the attacker needs to reach.
B	The total defense budget
b_{i}^{c}	The defense budget allocated to protect node i from being compromised, where $i\in N$
b_{il}^{e}	The defense budget allocated to protect node i from being escalated, where $i\in N$, $l\in L_i$
$\hat{a}_{i}^{c}(b_{i}^{c})$	The threshold of the attack budget required to compromise node i, where $i\in N$

$\hat{a}_{il}^{e}(b_{il}^{e})$	The threshold of the attack budget required to escalate to level *l* on node *i*, where *i*∈*N*, *l*∈*L*$_i$

The cost function and budget allocated to each are decision variables in the DRA Model, but we treat them as given parameters in the AEE Model. In other words, we assume the attack would know the defense budget allocation strategy here. A node can be compromised/escalated if the attacker applies more resources than the $\hat{a}_{i}^{c}(b_{i}^{c})\,/\,\hat{a}_{il}^{e}(b_{il}^{e})$ to it.

The decision variables of the AEE Model are defined as Table 2-6.

Table 2-6 Decision Variables of the AEE Model

Decision Variables	
Notation	**Description**
y_{sil}	1 if node *i* is compromised at stage *s* and escalated to level *l* of the node, and 0 otherwise (where *s*∈*S*, *i*∈*N*, *l*∈*E*$_i$)
x_p	1 if path *p* is selected as the attack path, and 0 otherwise (where *p*∈ *P*$_w$)

Our proposed model is as follows.

Objective:

$$\min_{y_{sil},\,x_p}\left(\sum_{i\in N,l\in E_i}\hat{a}_{i}^{c}(b_{i}^{c})\sum_{k\in S}y_{kil}\prod_{s\in S_{(k)}}\sum_{j\in N}\sum_{m\in E_i}\left(e_{j}^{c}y_{sjm}\right)+\sum_{i\in N,l\in L_i}\hat{a}_{il}^{e}(b_{il}^{e})\sum_{k\in S}y_{kil}\prod_{s\in S_{(k)}}\sum_{j\in N}\sum_{m\in L_i}\left(e_{jl}^{e}y_{sjm}\right)\right)\text{(IP 2)}$$

Subject to:

$$\sum_{p\in P_w}x_p\sigma_{pil}\le\sum_{s\in S}y_{sil}\qquad\forall\, i\in N,\,w\in W,\,l\in E_i\qquad\text{(IP 2.1)}$$

$$\sum_{p\in P_w} x_p = 1 \qquad \forall\, w\in W \qquad \text{(IP 2.2)}$$

$$x_p = 0\ or\ 1 \qquad \forall\, p\in P_w,\, w\in W \qquad \text{(IP 2.3)}$$

$$\sum_{l\in E_i} y_{kil} \le \sum_{s\in S_{(k)}} \sum_{j\in N} \sum_{l\in E_i} y_{sjl}\delta_{pijl} \qquad \forall\, i\in N,\, k\in S,\, p\in P_w,\, w\in W \qquad \text{(IP 2.4)}$$

$$\sum_{s\in S}\sum_{l\in E_i} y_{sil} \le 1 \qquad \forall\, i\in N \qquad \text{(IP 2.5)}$$

$$\sum_{i\in N}\sum_{l\in E_i} y_{sil} \le 1 \qquad \forall\, s\in S \qquad \text{(IP 2.6)}$$

$$y_{sil} = 0\ or\ 1 \qquad \forall\, i\in N,\, s\in S,\, l\in E_i \qquad \text{(IP 2.7)}$$

$$T \le \sum_{i\in N}\sum_{l\in E_i}\sum_{s\in S} I_{il}\, y_{sil}\ . \qquad \text{(IP 2.8)}$$

Explanation of the mathematical formulation:

- ✓ **Objective function:** The attacker's objective is to compromise multiple core nodes using the minimized total attack cost by deciding which nodes and levels to attack and which attack sequence to adopt. The result of the AEE Model is the same with the result of the inner problem in the DRA Model.
- ✓ **Constraint (IP 2.1) ~ Constraint (IP 2.3)** are the same with **Constraint (IP 1.1) ~ Constraint (IP 1.3)** in the DRA Model, and together form the path constraints.
- ✓ **Constraint (IP 2.4) ~ Constraint (IP 2.8)** are equal to **Constraint (IP 1.4) ~ Constraint (IP 1.8)** in the DRA Model.

Chapter 3 Solution Approach

3.1 Simulated Annealing Method

The Simulated Annealing method is an approach that was used to solve a number of complex combinatorial problems. Because solving these problems directly is difficult and inefficient, many large-scale combinatorial optimization problems are often solved by some divide-and-conquer or iterative improvement approaches. The Simulated Annealing method belongs to the latter.

Simulated Annealing (SA) approach is proposed by Kirkpartrick et al. [23] to solve large-scale combinatorial problems. SA is a procedure that simulates the process of material cooling and crystallizing to steady state. The procedure is used to solve combinatorial problems. Its main concept is iterative improvement operated by standard rearrangement operations. In the annealing process, initially, the material is heated to higher temperature with a higher energy state, so that the structure of atoms is unsteady and the atoms are more unstuck. Then, the cooling procedure is controlled to lower the temperature slowly to yield crystal, so that atoms would have lower thermal mobility and the structure of atoms would be tighter. To reach a stable energy state and structure, the procedure needs to proceed long enough and reach equilibrium at each temperature. If the annealing temperature does not decrease slowly enough, the material might trap into an unsteady state and the material may crystallize with some defect. Therefore, the energy state may not be the lowest one.

For a minimization problem, the objective function is analog to internal energy state. Thus, each feasible solution and its objective value are treated as the state of material and the internal energy on the state.

Initially, the SA method randomly generates an initial feasible solution, and set the parameters which are related to cooling procedure, such as initial temperature, final temperature, cooling ratio. At each state, SA randomly generates a new solution which is a neighbor of the current solution, and the procedure will then examine the feasibility of the neighbor solution which is rearranged from the original solution; in other words, the new solution is generated based on the current solution. If the new solution satisfies all of the constraints and the objective value (energy state) of new solution is smaller than or equal to the original one, the current state will be changed and the new solution will be accepted. If the objective value of new solution is larger than original one, there will be a probability to decide to accept the solution or not. The accept probability is defined as $p = exp(-\Delta E/T_t)$, where ΔE is the difference of the energy state between the new state and the original one, i.e., the difference of the objective value, and T_t is current temperature. A random number is generated each iteration. If the random number is larger than or equal to the p, the current state will be changed. Therefore, the heuristic has some opportunities to accept worse new solutions. The purpose of the probability function is to avoid local optimum. Besides, two parameters α and β are used to control the number of iterations at each temperature, where $\alpha < 1$ and $\beta > 1$. At temperature T_t, the procedure will repeat $b(T_t)$ iterations, and then executes cooling procedure to assign T_{t+1} to $\alpha \times T_t$ and set $b(T_{t+1})$ to $\beta \times b(T_t)$. When the temperature reaches the frozen temperature T_f, the system will be frozen and an approximated optimal solution will be obtained.

The Simulated Annealing procedure is described as Figure 3-1.

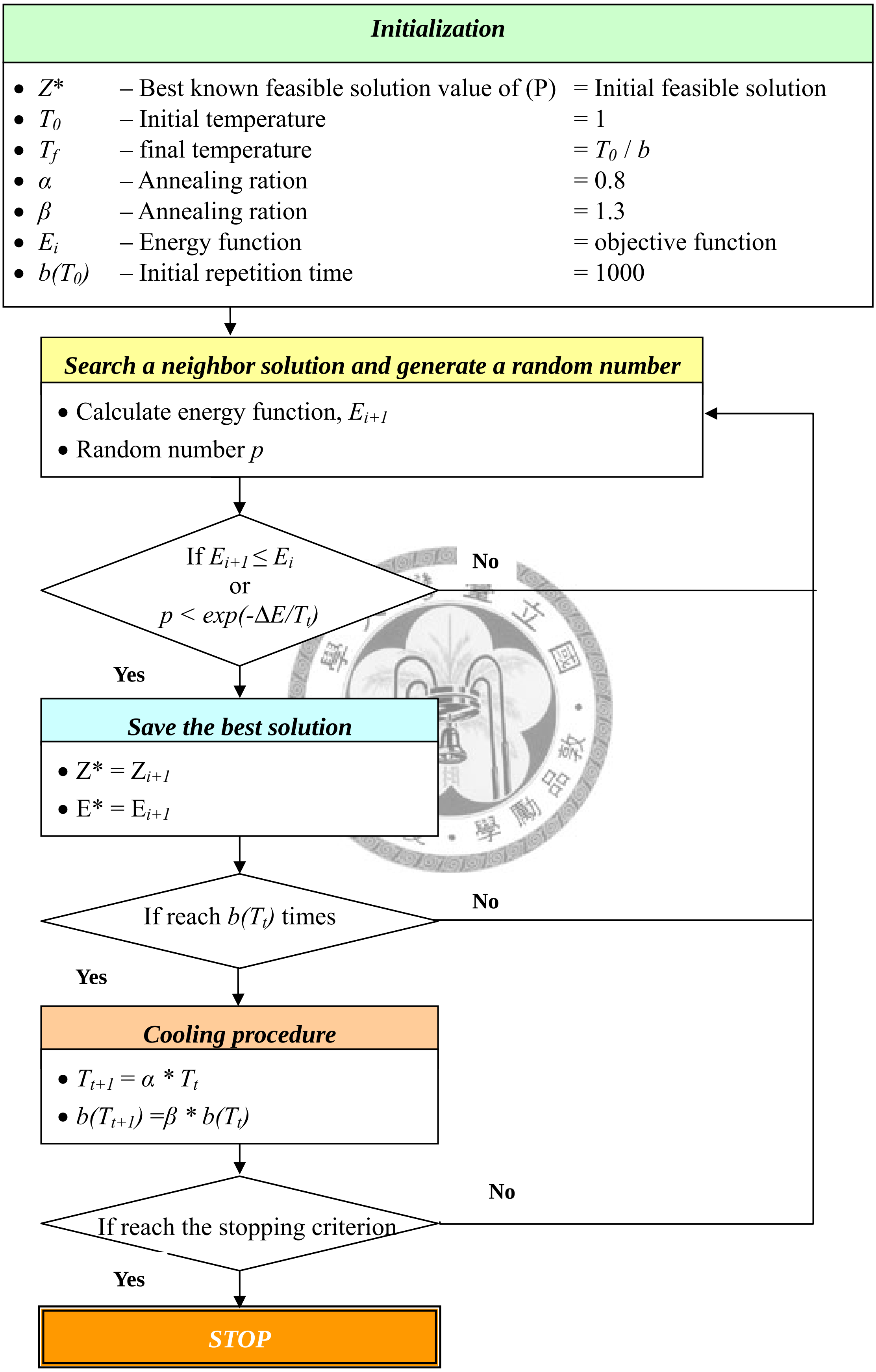
Initialization
• Z^* – Best known feasible solution value of (P) = Initial feasible solution
• T_0 – Initial temperature = 1
• T_f – final temperature = T_0 / b
• α – Annealing ration = 0.8
• β – Annealing ration = 1.3
• E_i – Energy function = objective function
• $b(T_0)$ – Initial repetition time = 1000
Search a neighbor solution and generate a random number
• Calculate energy function, E_{i+1}
• Random number p
If $E_{i+1} \leq E_i$
or
$p < exp(-\Delta E/T_t)$
No
Yes
Save the best solution
• $Z^* = Z_{i+1}$
• $E^* = E_{i+1}$
If reach $b(T_t)$ times
No
Yes
Cooling procedure
• $T_{t+1} = \alpha * T_t$
• $b(T_{t+1}) = \beta * b(T_t)$
If reach the stopping criterion
No
Yes
STOP

Figure 3-1 Simulated Annealing Method Procedure

3.2 Solution Approach for the AEE Model

In this section, we will solve the AEE Model by SA-based heuristics. Details about the algorithm are described as Table 3-1. We set the objective function as the energy state in SA and use some two-phase approaches to solve the problem. At the first phase, we initially generate an initial feasible solution by several approaches which we will discuss later. Then, we randomly generate a random number *p* and choose a neighbor solution and examine the feasibility of the new solution. If it is a feasible solution, we calculate the difference of the objective value between the new solution and the current one, i.e., Δ*E*. If Δ*E* is smaller than or equal to 0, the new solution is accepted. If Δ*E* is larger than 0, we compare *p* with *exp(-ΔE/T),* if *exp(-ΔE/T)* is larger than *p*, the worse new solution will be accepted. At the first phase, the approaches of generate neighbor solutions are not restricted, i.e., we would randomly search for new solutions by changing the sequence of an attack tree, the topology of the attack tree, or the escalation levels on the compromised nodes. By comparing the value of each accepted solution, we save the best ten solutions as the results of the first phase procedure. After conducting the complete first phase SA procedure, we run the second phase procedure. At this phase, we use the 10 best results of the first phase as the initial solutions. And we would search neighbor solutions by changing the sequence of the initial solutions only. That is, for a complete two-phase SA-based approach, we will run the SA procedure one time to search for the attack trees, the attack sequence and the escalation levels, and we will then use these results as the input of the second round SA procedure to adjust the attack sequence. Finally, we save the smallest objective value and its state as our best solution.

In our approaches, we use three different kinds of initial solutions and several methods for searching neighbor solutions. Among these, the first initial solution is an algorithm which is similar to Prim's Algorithm. Initially, we use the Prim's Algorithm to generate a minimum cost spanning tree. Next, we prune the unnecessary nodes of the spanning tree, i.e., nodes which are not core nodes and not the intermediate nodes on the paths towards core nodes. Finally, we adjust the escalation levels to satisfy the corresponding constraints. The second approach is a random-based algorithm. The difference between this solution and the first one is the criteria of choosing next node. Here, when choosing the next target node, we always randomly choose a reachable node as the next attack node instead of choosing the smallest weight node. The last one is also similar to the first approach. But it modifies the weight of each node from the attack cost to the ratio of the cost and experience. The time complexity of all the initial solutions is $O(|N|\log|N|)$.

Because the SA emphasizes that the neighbor solution is generated based on the pervious solution randomly, we will use these properties to develop our heuristics for searching neighbor solutions. The solutions could be divided into three parts which are the change of the attack sequence, the change of the attack tree, and the change of the escalation levels on compromised nodes. About the change the attack sequence, we use two different ways. One is to rearrange the whole traversal sequence. We start from the source node and randomly choose a compromised node that can be reached from the source node and re-label the sequence of that node. We will then repeat this process until all the compromised nodes are visited again. The other one is to exchange the attack sequence of two compromised nodes randomly. We also divide methods of changing the attack tree into two parts. One is large-range change and the other is small-range change. The large-range change means that we will randomly choose a compromised node and reset the nodes, which are compromised after the selected node on the attack tree by the attacker, to uncompromised. Then, we will start from the chosen node to find other paths randomly in order to complete the attack. Therefore, the new attack tree will be just the same with the original one before the chosen node. The small-range change is only to adjust small parts of the original attack tree. Here, we propose two different methods to do this. One is to change the path between two compromised nodes which are adjacent to each other in the attack tree and adjust the attack sequence if necessary. The other is to compromise an additional node which is not necessary for the original attack tree or remove an unnecessary node from the original attack tree. This is reasonable because the attacker may gain some additional experience from the extra attack and the experience of the attack may be helpful for him to reduce the future attack costs. Hence, it is not inevitable that the total attack cost of the attack tree with some additional unnecessary nodes will more than the total attack cost of the tree which only contains necessary nodes forming the original attack tree. Finally, we also develop two ways to change the escalation levels. One is to randomly exchange the escalation levels on two compromised nodes. The other is to escalate to a higher level or drop to a lower level on a randomly chosen compromised node. Although we develop several methods to search neighbor solutions, we will only randomly choose one approach for searching neighbor solution at each loop. The time complexity of searching for neighbor solutions is $O(|N|\log|N|)$.

All of the methods for searching neighbor solutions are based on the principles "random" and "neighbor." The property, random, is the reason that we always apply a random manner while adjusting the attack tree, the attack sequence and the escalation levels. Because of the property of neighbor, we need to divide the methods into three categories, so that we could apply only a small random change to searching new solutions and use the new accepted solution as the starting point of the next step. If we change the attack sequence, the attack tree and the escalation levels on compromised nodes at the same time, the new solution we obtained might be too far from the original one. In other words, the difference between the structure of new solution and the structure of the original one will be too huge.

For each initial solution, we will run a completely SA procedure with the proposed methods for searching neighbor solutions; in other words, we will use three SA-based heuristics and compare the results of them. The computational results will be described in chapter 4.

Table 3-1 Two-Phase SA-based Heuristic

```
// Start the first phase SA
// Set the initial configuration
Set the SA parameters, $t_0, t_f, \alpha, \beta$;
// Generate the initial feasible solution
According to the path constraints, choose an attack tree including all of the core
nodes;
Choose an escalation level on each compromised node and check the impact
constraint;
Calculate initial energy function $E_{old}$, $E_{min} \leftarrow E_{old}$; save the initial configuration as
the best solution;
$t \leftarrow t_0$, $b \leftarrow b_0$;
// Cooling procedure
While $t > t_f$ do
    Loop $b$ times
        // Search neighbor solutions
        Randomly alter the solution configuration (escalation level or topology of
    the attack tree or attack sequence);
14.         **If** the configuration does not violate any constraints, **then**
15.             Calculate $E_{new}$, and $\Delta E = E_{new} - E_{old}$;
16.             Generate a random number p, where $0 < p < 1$;
17.         **Else**
18.             Recover the action in Step (13);
19.         **If** $\Delta E \le 0$ or $p < exp(-\Delta E / t)$, **then**
20.             $E_{old} \leftarrow E_{new}$;
21.             **If** $E_{old} < E_{min}$, **then**
22.                 $E_{min} \leftarrow E_{old}$, save current configuration as the best solution;
23.                 Save the best ten solutions;
24.         **Else**
25.             Recover the action in Step (13);
26.     **End loop**
27.     $b \leftarrow b \times \beta$, $t \leftarrow t \times \alpha$;
28. **End While**
29.
30. // *End the first phase SA procedure and start the second phase SA procedure*
31.
32. Reset SA parameters to initial condition;
33. **Loop** 10 times
34.     Take one solution from the best ten solutions of the first phase SA as the initial
    solution;
35.     $t \leftarrow t_0$, $b \leftarrow b_0$;
36.     **While** $t > t_f$ **do**
37.         **Loop** $b$ times
38.             Randomly alter the solution configuration (attack sequence only);
39.             **If** the configuration does not violate any constraints, **then**
40.                 Calculate $E_{new}$, and $\Delta E = E_{new} - E_{old}$;
41.                 Generate a random number p, where $0 < p < 1$;
42.             **Else**
43.                 Recover the action in Step (38);
44.             **If** $\Delta E \le 0$ or $p < exp(-\Delta E / t)$, **then**
45.                 $E_{old} \leftarrow E_{new}$;
46.             **If** $E_{old} < E_{min}$, **then**
47.                 $E_{min} \leftarrow E_{old}$, save current configuration as the best solution;
48.                 Save current configuration as the best solution;
49.             **Else**
```

```
50.                 Recover the action in Step (38);
51.         **End loop**
52.         $b \leftarrow b \times \beta$, $t \leftarrow t \times \alpha$;
53.     **End While**
54. **End loop**
55. $Z_{ip} \leftarrow E_{min}$;
56. **End**
```

3.3 Solution Approach for the DRA Model

The main objective of the DRA Model is to maximize the minimized total attack cost. Thus, the best solution of the AEE Model can be used as an input of the DRA Model and we will then adjust the budget allocation strategy according to the current attack strategy. After the adjustment, we solve the AEE Model again to gain an attack strategy corresponding to the new defense budget allocation strategy. In other words, it is a battle between the defender and the attacker. While the attacker determines the best attack strategy, the defender would adjust the defense strategy against the attacks. In response, the attacker will then change his attack strategy again. Thus, the AEE Model is used to decide the best attack strategy, and the DRA Model is used to simulate the interaction between the attacker and the defender.

The main concept of the adjustment procedure is to extract a small proportion of the budget from the uncompromised nodes, and then allocate it to compromised nodes. The reason is that if a node is not compromised, it indicates that the node may be not profitable for the attacker or the budget allocated to the node may be too much.

In order to deduct some resources from uncompromised nodes, we assign a weight to each node to compare the importance of nodes. Because the problem we addressed here is to evaluate the experience from attacks, it is important to measure the sequence of attacks. Hence, we will consider the attack sequence as a factor of the weight of nodes. In addition, how many times that a node has been compromised during the interaction between the attacker and the defender may also be useful information to evaluate the weight of the node. Thus, we use the attack sequence and the number of times a node has been compromised as the metric of the weight.

We calculate the average frequency that a node has been compromised during the interaction between the defender and the attacker. If a node is compromised each iteration, its average frequency of being compromised is one. And then, we measure the impact of sequence. First, we sort nodes by their compromised sequence and classify every three nodes into a group and set the corresponding impact. For example, if there are 21 nodes in the attack tree, we would classify them into 7 groups. The first group contains three nodes which are the first, the second and the third nodes the attacker compromised in the attack tree. Next, the impact of the first group is set to 1 and the second group is set to 6/7. Thus, the impact of the last group is assigned to 1/7. After calculating the impact of average attack frequency and attack sequence, we consider the two factors jointly. That is, we set the weight of each node to $F_i \times S_i^a$, where F_i is the average frequency that node *i* has been compromised, and S_i^a is the impact of the attack sequence of node *i*. Consequently, the budget extracted from node *i* would be a proportion, $\alpha \times (1 - F_i \times S_i^a)$, of the budget allocated to it.

After extracting budget from nodes, we will then allocate the extractive budget to the nodes on the current attack tree. We propose three kinds of reallocation strategies here. The first one is to uniformly reallocate the extractive budget to compromised nodes and we denote this reallocation strategy as *R_Uni*. The second strategy denoted as *R_Deg* is to allocate budget according to nodal degree. The last one is a sequence-based strategy reallocating budget in accordance with the attack sequence and this strategy is denoted as *R_Se*. The time complexity of the adjustment procedure is *O*(|*N*|).

Chapter 4 Computation Experiments

4.1 Computation Experiments with the AEE Model

4.1.1 Simple Algorithms

To measure the effective of our proposed algorithms, we design the following simple algorithms.

Simple algorithm 1 is also an SA-based heuristic and it can be divided into an outer part and an inner part. Initially, at outer loop, we ignore the effect of experience and then run the Prim's Algorithm. Thus, we can know the total cost of the tree. Next, we reset the weight on each node using the experience factor. We start from the source node and replace the weight of the first node the attacker would compromise with the value that the original nodal cost subtracts the effect of its experience. The effect would be calculated by the experience factor of the current target node multiplying the total attack cost of all the nodes which are compromised after the current node on the attack tree. This weight means whether the effect of experience on a node can balance its cost or not. After repeating this action and applying this weight to each node, we run the SA procedure which is the inner loop of this heuristic to adjust the sequence and escalation levels. Then, modify the weight of each node again and repeat these actions several times. The pseudo code of simple algorithm 1, which is denoted as *S1*, is presented below.

Table 4-1 Simple Algorithm 1

```
While ( there is no improvement after b iterations)
    Prim's Algorithm(); // Generate a spanning tree
    For each node i
        // Update weight
        weight ← [cost - experience * (total cost of its descendants)];
    End For
    // Adjust sequence and escalation level
    SA procedure();
End While
End
```

Other simple algorithms are derived from our two-phase SA-based algorithms. We use the first phase SA procedures and each initial solution as the comparisons and evaluate the improvement ratio. We also use different initial solutions to distinguish our approaches. The first one is the Prim-based algorithm, which is denoted as *Prim_based*, and its corresponding one-phase and two-phase SA approaches are denoted as *SA_Prim* and *TSA_Prim* respectively. The second initial solution is the approach which randomly chooses the next node to compromise, and we denote it as *Random* and its corresponding SA solutions are denoted as *SA_Random* and *TSA_Random*. The last one, denoted as *Weight*, is the solution using the ratio between the experience and the cost of a node and its one-phase and two-phase approaches are denoted as *SA_Weight* and *TSA_Weight* respectively Therefore, we use *S1*, *Prim_based*, *SA_Prim*, *Random*, *SA_Random*, *Weight* and *SA_Weight* as our simple algorithms.

4.1.2 Experiment Environment

The proposed algorithms are coded in C++ and executed on a PC with Intel(R) Pentium 4 3.00GHz CPU and 512MB RAM. The SA parameter α is set to 0.7, and β is set to 1.3. The initial temperature T_0 is initialized to 1.0 and the final temperature is set to T_0/1000. At each temperature, we control the SA to repeat b_0 times, and initialize b_0 to 1000. We randomly assign the experience value and the number of vulnerabilities on each node. We assume that there are three escalation levels on each node.

In order to evaluate the quality of our approaches, we compare our solutions to the exhaustive search in three small size networks. The first network is a grid network which is a 3*3 square; the second one is a random network with 9 nodes; and the last one is a scale-free network with 9 nodes. We consider one escalation level at these three types of networks due to the efficiency of exhaustive search. Because these networks are small enough, we are able to use the exhaustive search to find the optimal solutions.

Thus, by comparing these optimal solutions with our solutions, we can measure the efficiency of our algorithms.

Besides, in other larger size networks, we use two ways to evaluate the quality of our solutions. One is to compare the solutions of our approaches with some simple algorithms and measure the improvement ratio of the two-phase SA procedures. The other is to design some particular networks in which we can find the optimal solutions intuitively. In other words, we can easily find the optimal attack trees and the optimal attack sequence in these networks.

For instance, in Figure 4-1, nodes of this topology can be divided into four types. We set the experience of the first type nodes to 1.0 and the costs to 20. The experience values of the second type nodes are set to 0.98 and costs are set to 5; the experience values of the third type are set to 1.0 and costs are set to 20; the fourth type are the nodes whose experience value are set to 0.96 and costs are set to 5. We set the second type and the third type nodes and the last node of the first type nodes as core nodes. Because the experience of first type nodes are 1 and these nodes are the necessary nodes to reach the core nodes, we can ignore the effect of experience of these nodes. Similarly, the third type nodes are all core nodes and the experience values are all one, so this type should be compromised after other nodes are all compromised. As the figure shows, there are two choices to reach node 3 from node 1. One is to go along with node 1, node 2 and node3. The other path is directly from node 1 to node 3. Comparing these two paths, if we do not consider the effect of experience, the better choice would be from node 1 to node 3 directly. Once we further to evaluate the effect of experience, the choice would not be the same. The attacker compromises node 2 can gain experience 0.96 and he only needs to pay cost 5. Therefore, this experience can be applied to compromising node 3, i.e., the attacker needs to pay cost 20 × 0.96 = 19.2 to compromise node 3 and an additional cost of compromising node 2. In this case, the attacker may still choose from the node 1 to node 3 directly, but if we consider the whole network, the choice might be changed. For example, if there are five nodes belong to the first type nodes and three nodes are the third type nodes in the network. The third type and the second type nodes are all needed to be compromised. Thus, if the attacker compromises node 2 first and then compromises other nodes which must be compromised in the network, he can reduce the cost from 160 ($20\times 5+20\times 3$) to 158.6 ($0.96\times(20\times 5+20\times 3)+5$). Thus, he would choose to compromise the additional node to reduce his total attack cost. The smallest case we measure here are totally 47 nodes where 16 first type nodes, 8 second type nodes, 8 third type nodes and 15 fourth type nodes. 8 third type nodes would be compromised last because of the experience values of them are 1. So, for the last node of the fourth type nodes, compromising it or not depends on the difference between its cost and how much experience the attacker can gain to reduce the future attack costs. Thus, its experience would be applied to 8 third nodes at least. Due to the calculation above, the attacker should choose to compromise this additional node. In other words, to reach the last node of the first type nodes, the attacker would choose a more winding path but a straightforward path. Next, consider the second type nodes, the costs of these nodes are 5 and experience values are 0.98. After compromising a first type node, the attacker can choose to attack a second type node or a fourth type node. Because the costs are the same, the decision would be based on the experience. That means the attacker would compromise a fourth type node first and then compromises a second type node. Thus, in this kind of network, we can easily evaluate the optimal attack tree and optimal the attack sequence.

Figure 4-2 is another example. In this network, we can use a backtracked method to calculate the optimal solution. In order to compromise the core node, the attacker must pass through each intermediate node. There are two paths from each intermediate node to the next one. One is to pass through the right path and the other is through the left path. The total attack cost of the descendants of an intermediate node would be the same no matter what path the attacker chooses. So, the attacker can decide to pass through which path to the next intermediate node just under the consideration that the node on the right side or on the left would bring better effect. For instance, if the attacker is at node 1 and the target is node 4, he can attack node 2 or node 1 to reach node 4. Assume that the cost of node 2 is 9 and the experience is 0.6; and the cost of node 3 is 8 and experience is 0.8. The cost of node 4 is 10. In this condition, the total attack cost of all the descendants of current node is 10 (the cost of node 4). If the attacker chooses the right path, he needs to pay cost 16 ($8+10\times0.8$), but if he chooses the other way, he only needs to pay cost 15 ($9+10\times0.6$). Clearly, a better choice would be the left path. After calculating these nodes, we add the cost 15 to the pervious intermediate node, here that is node 1. By applying this procedure, we can continuously repeat this manner to backtrack to the source node and find the optimal solution.

We also consider other networks such as grid networks, random networks and scale-free networks with nodes from 25 to 144 to evaluate the robustness of these networks. Because our purpose here is to evaluate some general scenarios, we would measure the networks with three escalation levels on each node.

There may be several types of the cost functions, such as linear functions, convex functions and concave functions. The cost functions here are set to concave functions.

Because the effect of additional budget allocated to nodes may declines as the defense budget increases. Thus, concave functions, e.g. log functions may describe the real situation more accurately. Besides, we evaluate the effect of the number of the vulnerabilities on each node in our cost function. The more number of the vulnerabilities a node has, the less cost the attacker should pay to compromise it. Similarly, the effect of the vulnerabilities should decrease as the number of the vulnerabilities increases. Thus, we define the cost function here as a form, $\ln(\frac{b_i \times M}{V_i}+1)$, where b_i is the budget allocated to node i and V_i is the number of the vulnerabilities on node i and M is a constant to adjust the proportion of b_i and V_i. The cost functions of different escalation levels on nodes are also defined as this form.

In addition, we also design different budget allocation strategies. The first policy is a uniform allocation strategy. In this strategy, each node is allocated the same defense budget. The second strategy is a degree-based budget allocation. Each node is allocated budget according to the percentage of its degree over the total degree of the network. The last strategy is a vulnerability-based budget allocation. Budget allocated to each node depends on the ratio of the vulnerabilities on each node and total vulnerabilities in the networks. Because there are several levels on a node, the budget allocated to each level on a node is also different. As noted earlier, the network can be viewed as a two-dimensional network. Thus, while allocating defense budget to escalation levels, we can treat the different levels on a node as different nodes in this artificial two-dimensional network. Consequently, we can easily use this property to allocate budget to each level in degree-based and uniform defense budget allocation strategies.

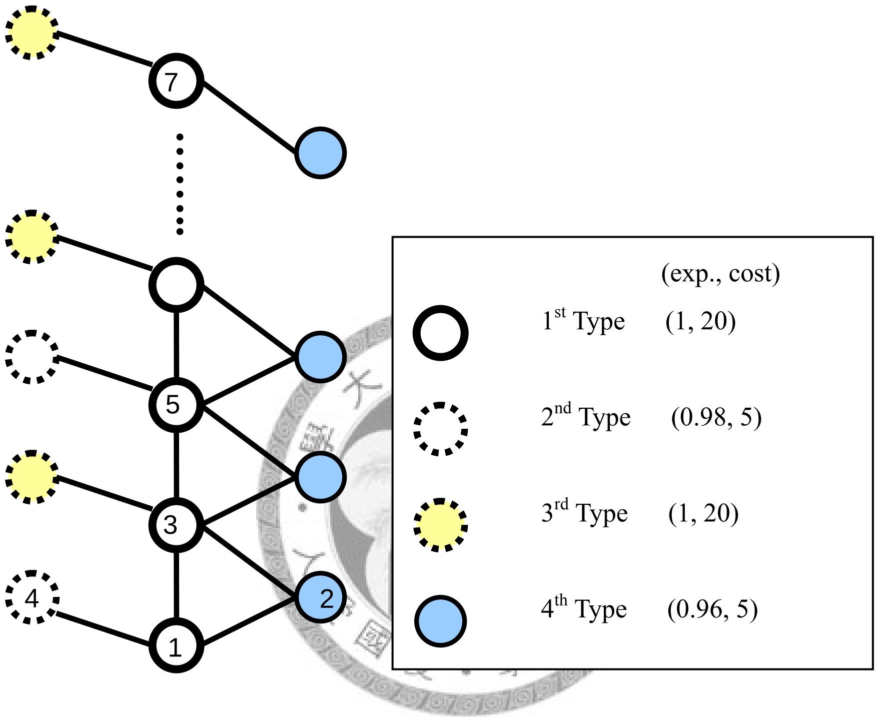

Figure 4-1 Experiment Topology 1

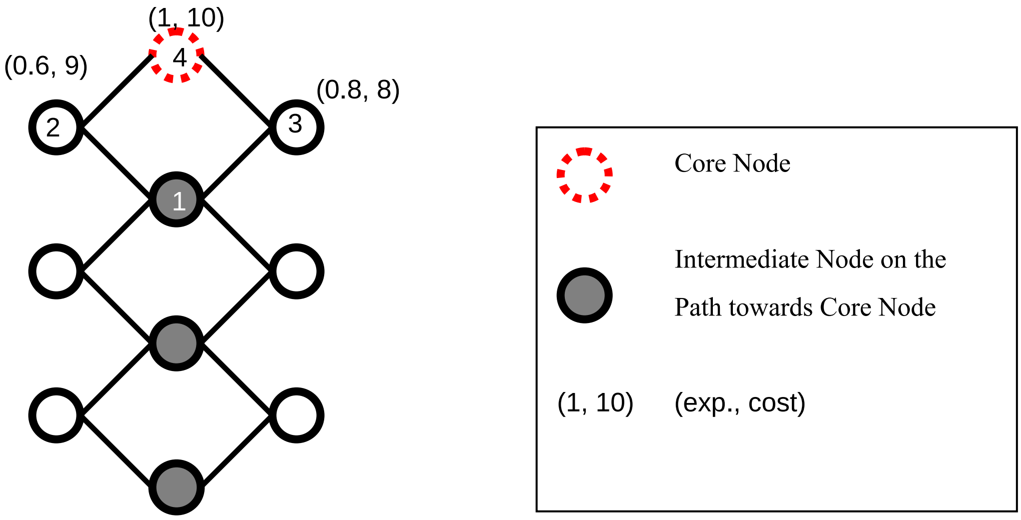

Figure 4-2 Experiment Topology 2

The parameters and related environment used in our experiments are detailed below.

Table 4-2 Experiment Parameter Settings

Parameters of SA	
Parameters	**Value**
Initial Temperature	1.0
Initial Iterations	$b_0 = 1000$
Final Temperature	Initial Temperature / b_0
Cooling Parameter	$\alpha = 0.7$ $\beta = 1.3$
Test Platform	CPU: INTELTM Pentium 4 3.0GHz RAM: 512 GB OS: Microsoft Windows XP

Parameters of the Model	
Parameters	**Value**
Testing Topology	Grid networks, Random networks, Scale-free networks
Number of Nodes |N|	9, 25, 49, 81, 100, 144
Total Defense Budget	Equal to 2 Times the Number of Nodes
Budget Allocation Strategy	Uniform allocation (Uni), Degree-based allocation (Deg), Vulnerability-based (Vul)
Defense Capability $\hat{a}_i(b_i)$	$\hat{a}_i(b_i, V_i) = \ln(\frac{b_i \times M}{V_i} + 1)$, b_i is the budget allocated to node i and V_i is the vulnerabilities on node i, $\forall\, i \in N$
Total Escalation Levels on Each Node	3

4.1.3 Experiment Results

To evaluate the robustness of different networks, we use the minimized total attack cost as our metric. That is, the higher minimized total attack cost the attacker pays, the more robust the network is. *TSA* value means the total attack cost calculated by the two-phase SA-based process. The *SA* value means the first phase results of our solutions. The *Ini.* value means the results of initial solutions. To evaluate the quality of *TSA*, we calculate the improvement ratio of *TSA* to *SA* and *Ini.* by $\frac{SA-TSA}{TSA}\times 100\%$ and $\frac{Ini.-TSA}{TSA}\times 100\%$ respectively.

Table 4-3 Experiment Results of Networks with 9 nodes and 1 escalation level

Network Topology	Budget Allocation	Initial Solution	TSA	Improvement Ratio to SA	Exhaust Search	Error Rate
Grid Networks	Uni	Prim_based	8.30945	0.00%	8.47228	0.00%
		Weight	8.30945	0.00%		0.00%
		Random	8.30945	0.00%		0.00%
	Deg	Prim_based	8.47228	0.00%	8.47228	0.00%
		Weight	8.47228	0.00%		0.00%
		Random	8.47228	0.00%		0.00%
	Vul	Prim_based	8.58804	0.00%	8.58804	0.00%
		Weight	8.58804	0.00%		0.00%
		Random	8.58804	0.00%		0.00%
Random Networks	Uni	Prim_based	7.30564	0.00%	7.30564	0.00%
		Weight	7.30564	0.00%		0.00%
		Random	7.30564	0.00%		0.00%
	Deg	Prim_based	7.6147	0.00%	7.6147	0.00%
		Weight	7.6147	0.00%		0.00%
		Random	7.6147	0.00%		0.00%
	Vul	Prim_based	7.80811	0.00%	7.80811	0.00%
		Weight	7.80811	0.00%		0.00%
		Random	7.80811	0.00%		0.00%
Scale-free Networks	Uni	Prim_based	6.91938	0.00%	6.91938	0.00%
		Weight	6.91938	0.00%		0.00%
		Random	6.91938	0.00%		0.00%
	Deg	Prim_based	7.17378	0.00%	7.17378	0.00%
		Weight	7.17378	0.00%		0.00%
		Random	7.17378	0.00%		0.00%
	Vul	Prim_based	6.89044	0.00%	6.89044	0.00%
		Weight	6.89044	0.00%		0.00%
		Random	6.89044	0.00%		0.00%

Table 4-4 Experiment Results of Experiment Topology 1 and Experiment Topology 2

Network Topology	Node Number	Initial Solution	TSA	Improvement Ratio to SA	Optimal Solution	Error Rate
Experimental Topology 1	47	Prim_based	390.176	0.14%	390.176	0.00%
		Weight	390.176	0.00%		0.00%
		Random	390.176	0.20%		0.00%
	80	Prim_based	488.518	0.01%	488.432	0.02%
		Weight	488.547	0.07%		0.02%
		Random	488.636	0.08%		0.04%
	101	Prim_based	584.31	0.00%	583.59	0.12%
		Weight	583.733	0.06%		0.02%
		Random	583.733	0.00%		0.02%
Experimental Topology 2	10	Prim_based	99.39	0.00%	99.39	0.00%
		Weight	99.39	0.00%		0.00%
		Random	99.39	0.00%		0.00%
	49	Prim_based	427.965	0.00%	427.965	0.00%
		Weight	427.965	0.00%		0.00%
		Random	427.965	0.00%		0.00%
	82	Prim_based	588.929	0.00%	588.929	0.00%
		Weight	588.929	0.00%		0.00%
		Random	588.929	0.00%		0.00%
	100	Prim_based	650.622	0.00%	650.622	0.00%
		Weight	650.622	0.00%		0.00%
		Random	650.622	0.00%		0.00%

Table 4-5 Experiment Results of Networks with 25 nodes

Network Topology	Budget Allocation	Initial Solution	TSA	Improvement Ratio to Ini.	Improvement Ratio to SA	S1
Grid Networks	Uni	Prim_based	17.067	49.36%	0.42%	16.3045
		Weight	16.3045	65.16%	0.00%	
		Random	17.067	67.30%	0.47%	
	Deg	Prim_based	16.1115	85.56%	0.00%	17.3475
		Weight	15.7374	71.60%	0.00%	
		Random	15.7374	41.14%	0.00%	
	Vul	Prim_based	13.6048	73.02%	3.99%	13.6048
		Weight	13.6048	81.26%	0.00%	
		Random	13.6048	80.74%	0.00%	
Random Networks	Uni	Prim_based	13.6477	12.04%	0.00%	15.1472
		Weight	13.6453	30.24%	0.00%	
		Random	13.6477	56.08%	0.17%	
	Deg	Prim_based	14.6404	90.17%	0.00%	15.4821
		Weight	14.6404	28.94%	3.05%	
		Random	14.6404	83.15%	0.00%	
	Vul	Prim_based	13.0879	40.85%	0.00%	13.0879
		Weight	13.0951	76.97%	0.06%	
		Random	13.1004	62.96%	-0.10%	
Scale-free Networks	Uni	Prim_based	11.793	17.63%	0.07%	14.3948
		Weight	11.793	54.38%	0.00%	
		Random	11.793	35.69%	0.00%	
	Deg	Prim_based	14.0495	72.12%	0.23%	15.4361
		Weight	14.0495	51.18%	0.00%	
		Random	14.0495	27.15%	0.00%	
	Vul	Prim_based	10.3934	1.42%	0.16%	11.9421
		Weight	10.3934	11.67%	0.16%	
		Random	10.3934	12.75%	0.00%	

Table 4-6 Experiment Results of Networks with 49 nodes

Network Topology	Budget Allocation	Initial Solution	TSA	Improvement Ratio to Ini.	Improvement Ratio to SA	S1
Grid Networks	Uni	Prim_based	18.3298	108.44%	3.07%	21.9722
		Weight	18.0553	88.82%	0.00%	
		Random	18.1337	53.19%	2.27%	
	Deg	Prim_based	18.1227	39.10%	0.00%	23.0121
		Weight	18.1227	65.41%	0.95%	
		Random	18.1227	95.13%	0.00%	
	Vul	Prim_based	16.8577	67.71%	0.00%	18.558
		Weight	16.8338	48.01%	0.14%	
		Random	16.8577	57.05%	0.78%	
Random Networks	Uni	Prim_based	17.5207	77.45%	2.18%	22.2319
		Weight	17.8796	76.80%	0.00%	
		Random	17.4353	120.80%	0.00%	
	Deg	Prim_based	18.2307	75.52%	0.00%	24.3793
		Weight	18.2307	102.88%	0.00%	
		Random	18.2307	79.87%	0.00%	
	Vul	Prim_based	16.4035	101.00%	1.45%	19.4409
		Weight	16.7009	95.87%	1.41%	
		Random	16.2982	73.11%	0.00%	
Scale-free Networks	Uni	Prim_based	17.0068	70.36%	1.30%	25.3068
		Weight	16.4659	84.38%	0.00%	
		Random	16.8559	38.94%	0.09%	
	Deg	Prim_based	18.2237	47.95%	0.00%	27.2766
		Weight	17.9234	73.27%	0.42%	
		Random	17.9234	73.86%	4.62%	
	Vul	Prim_based	15.2985	80.22%	3.53%	20.6346
		Weight	16.063	64.15%	0.34%	
		Random	15.8402	52.35%	0.23%	

Table 4-7 Experiment Results of Networks with 81 nodes

Network Topology	Budget Allocation	Initial Solution	TSA	Improvement Ratio to Ini.	Improvement Ratio to SA	S1
Grid Networks	Uni	Prim_based	24.1036	26.55%	0.00%	29.1466
		Weight	22.7427	96.82%	2.43%	
		Random	22.4084	112.19%	0.00%	
	Deg	Prim_based	22.0132	63.64%	3.80%	28.156
		Weight	22.9025	66.21%	2.41%	
		Random	22.5665	82.71%	3.04%	
	Vul	Prim_based	19.3422	67.24%	3.14%	28.0975
		Weight	19.3632	46.14%	0.00%	
		Random	19.3422	124.02%	0.00%	
Random Networks	Uni	Prim_based	23.165	136.49%	2.26%	29.4004
		Weight	22.2234	168.03%	0.00%	
		Random	22.665	122.46%	1.38%	
	Deg	Prim_based	22.7699	108.04%	3.31%	29.9849
		Weight	22.3458	141.07%	0.00%	
		Random	22.4739	48.81%	4.87%	
	Vul	Prim_based	19.8119	107.97%	2.03%	30.7614
		Weight	20.5151	116.71%	0.00%	
		Random	20.3446	118.00%	0.37%	
Scale-free Networks	Uni	Prim_based	18.3464	109.14%	0.93%	32.405
		Weight	19.2383	149.72%	1.53%	
		Random	18.3999	116.71%	3.68%	
	Deg	Prim_based	22.5509	98.87%	2.84%	33.5005
		Weight	21.2131	116.43%	4.98%	
		Random	21.2131	86.11%	6.69%	
	Vul	Prim_based	17.6088	104.44%	3.32%	26.4876
		Weight	17.0667	100.34%	7.64%	
		Random	17.9647	90.70%	2.48%	

Table 4-8 Experiment Results of Networks with 100 nodes

Network Topology	Budget Allocation	Initial Solution	TSA	Improvement Ratio to Ini.	Improvement Ratio to SA	S1
Grid Networks	Uni	Prim_based	31.5009	85.18%	0.00%	41.6078
		Weight	29.6952	80.86%	0.00%	
		Random	30.6794	69.42%	6.14%	
	Deg	Prim_based	32.3269	119.94%	0.00%	42.5538
		Weight	29.137	91.21%	0.86%	
		Random	30.0537	75.05%	4.43%	
	Vul	Prim_based	28.8378	104.50%	2.48%	34.6427
		Weight	25.2229	89.47%	0.00%	
		Random	26.4644	123.87%	1.04%	
Random Networks	Uni	Prim_based	29.1791	85.40%	1.97%	39.2367
		Weight	27.422	124.96%	0.00%	
		Random	27.9177	57.30%	0.38%	
	Deg	Prim_based	29.3654	150.85%	1.91%	46.0847
		Weight	29.594	115.00%	0.00%	
		Random	29.4845	53.55%	1.74%	
	Vul	Prim_based	23.2003	114.32%	0.00%	34.9472
		Weight	25.0353	119.93%	0.00%	
		Random	24.8418	79.92%	0.32%	
Scale-free Networks	Uni	Prim_based	26.7715	89.53%	0.00%	39.414
		Weight	26.121	116.65%	0.00%	
		Random	26.0372	65.72%	0.88%	
	Deg	Prim_based	29.333	65.91%	0.00%	38.0386
		Weight	28.3074	116.77%	0.00%	
		Random	28.0432	114.92%	3.71%	
	Vul	Prim_based	25.2489	112.34%	2.50%	34.1402
		Weight	24.2604	119.91%	2.15%	
		Random	24.2124	96.59%	8.73%	

Table 4-9 Experiment Results of Networks with 144 nodes

Network Topology	Budget Allocation	Initial Solution	TSA	Improvement Ratio to Ini.	Improvement Ratio to SA	S1
Grid Networks	Uni	Prim_based	38.0503	125.06%	1.14%	50.3497
		Weight	39.5596	101.20%	0.27%	
		Random	38.511	74.08%	1.26%	
	Deg	Prim_based	39.782	82.85%	0.00%	50.3017
		Weight	39.0366	82.51%	0.30%	
		Random	39.5608	79.85%	0.59%	
	Vul	Prim_based	36.4206	52.13%	0.00%	41.9704
		Weight	35.7813	50.39%	0.04%	
		Random	35.3611	88.45%	0.21%	
Random Networks	Uni	Prim_based	34.9907	116.76%	0.04%	50.6679
		Weight	34.4164	103.76%	0.43%	
		Random	33.9935	67.84%	0.00%	
	Deg	Prim_based	41.4724	109.41%	2.37%	60.0891
		Weight	37.128	121.73%	0.36%	
		Random	39.5733	70.20%	0.00%	
	Vul	Prim_based	34.4716	56.87%	1.51%	40.4979
		Weight	33.3454	57.28%	0.30%	
		Random	31.1273	89.82%	0.31%	
Scale-free Networks	Uni	Prim_based	33.14	74.24%	1.04%	50.6356
		Weight	32.1484	77.87%	0.00%	
		Random	35.2476	62.99%	1.51%	
	Deg	Prim_based	36.9738	117.34%	0.15%	50.2134
		Weight	37.1385	111.05%	0.00%	
		Random	41.1268	48.58%	2.44%	
	Vul	Prim_based	32.4377	81.54%	0.39%	44.0077
		Weight	27.5028	116.66%	0.63%	
		Random	32.3247	74.60%	0.74%	

Table 4-10 The Elapsed Time of the Proposed Approaches

Network Topology	Number of Nodes	SA (Sec.)	TSA (Sec.)
Grid Networks	25	25	157
	49	46	238
	81	88	412
	100	129	748
	144	284	1334
Ransom Networks	25	26	155
	49	60	265
	81	103	424
	100	144	604
	144	309	1250
Scale-free Networks	25	26	153
	49	56	217
	81	90	335
	100	135	763
	144	265	1113

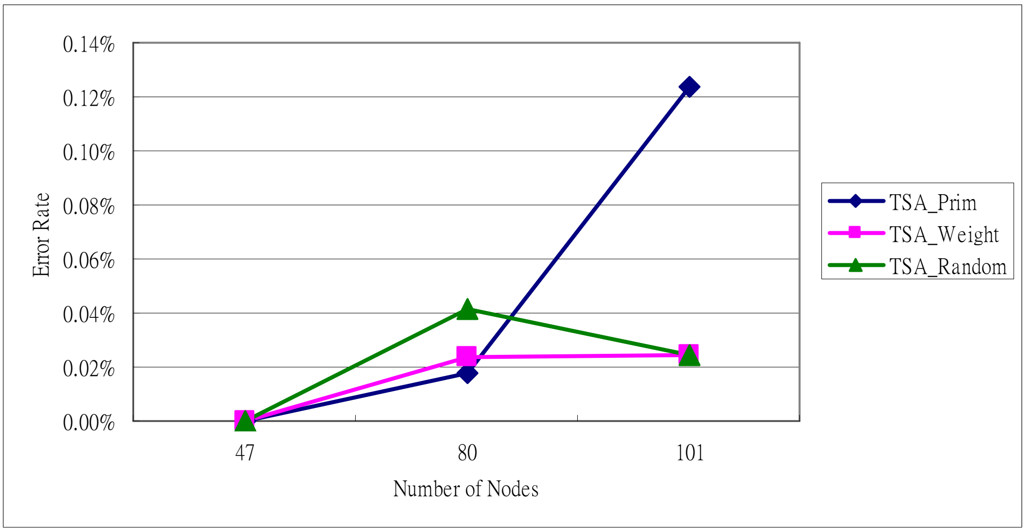

Figure 4-3 The Error Rate of Different Solution Approaches under Experimental Networks 1

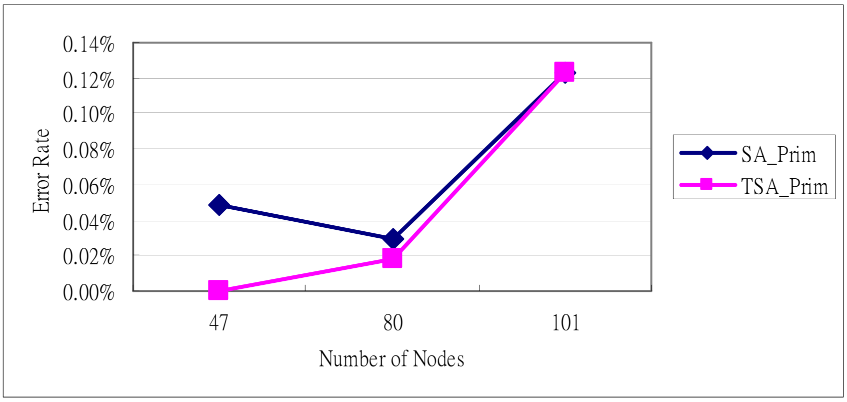

Figure 4-4 The Difference of the Error Rate between the *SA_Prim* and *TSA_Prim* under Experimental Networks 1

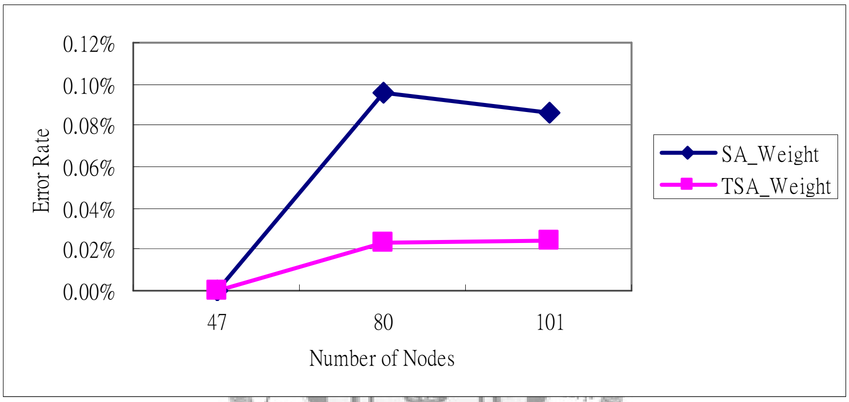

Figure 4-5 The Difference of the Error Rate between the *SA_Weight* and *TSA_Weight* under Experimental Networks 1

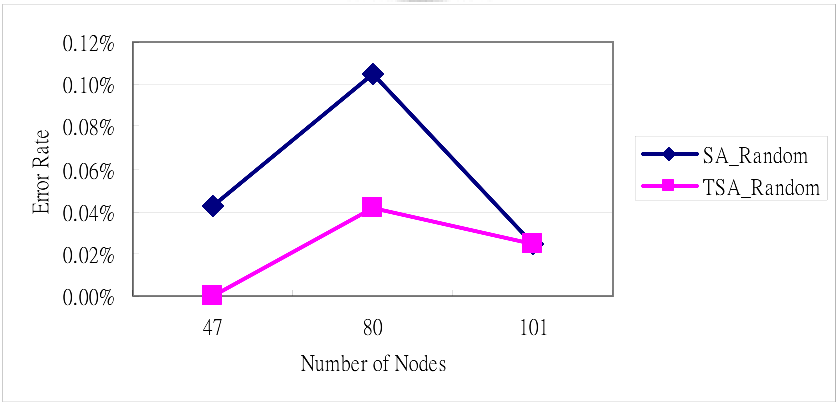

Figure 4-6 The Difference of the Error Rate between the *SA_Random* and *TSA_Random* under Experimental Networks 1

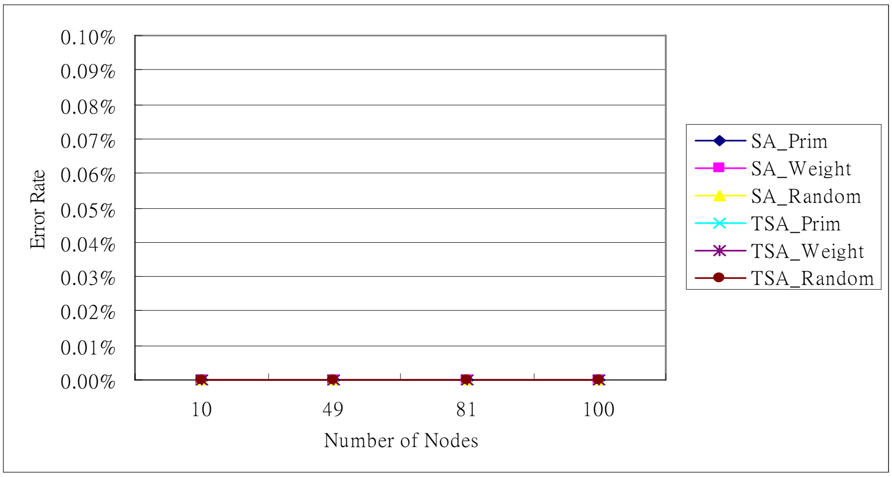

Figure 4-7 The Error Rate of Different Solution Approaches under Experimental Networks 2

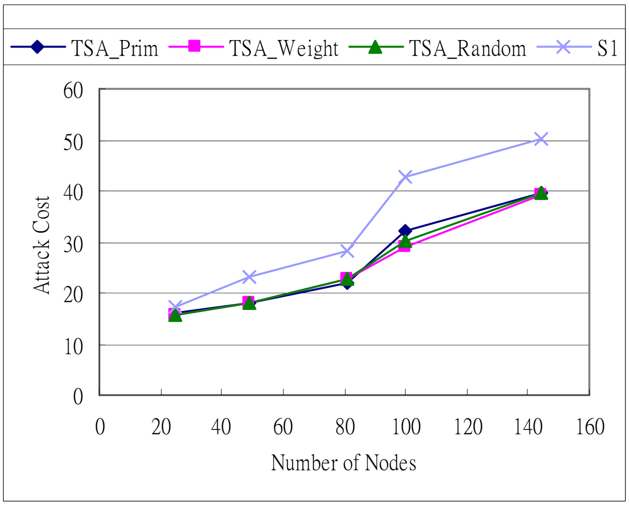

Figure 4-8 Total Attack Costs of Different Solution Approaches under Grid Networks with Degree-based Defense Budget Allocation Strategy

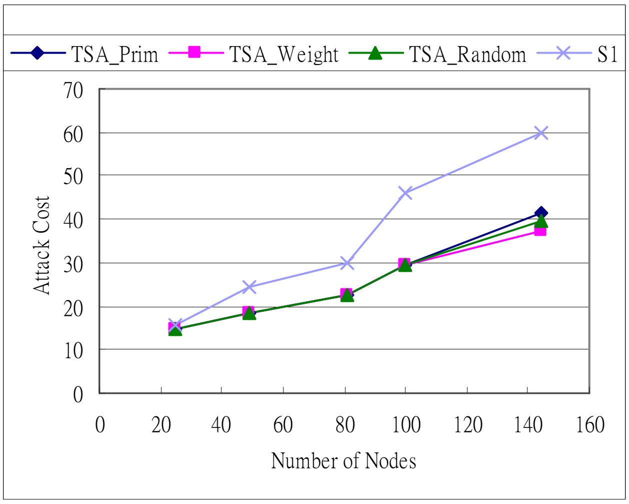

Figure 4-9 Total Attack Costs of Different Solution Approaches under Random Networks with Degree-based Defense Budget Allocation Strategy

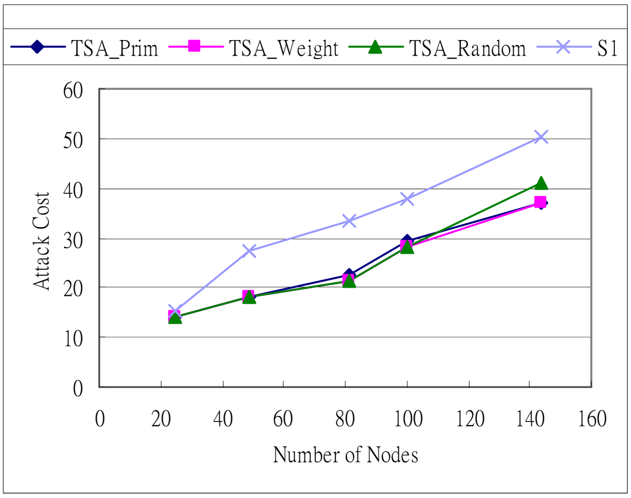

Figure 4-10 Total Attack Costs of Different Solution Approaches under Scale-free Networks with Degree-based Defense Budget Allocation Strategy

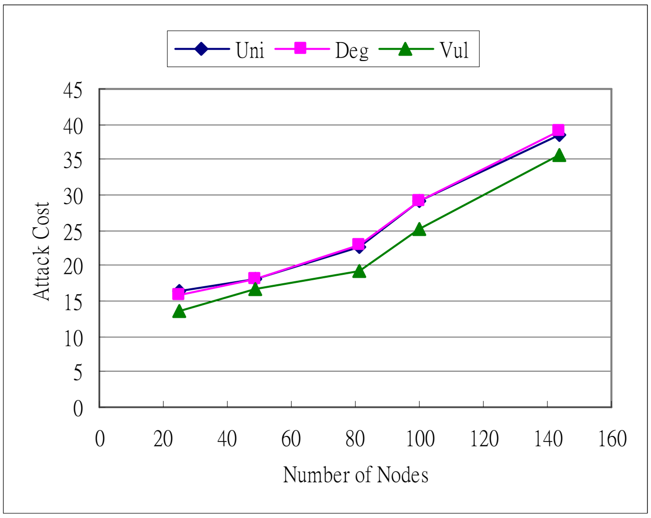

Figure 4-11 Total Attack Costs of Grid Networks under Different Defense Budget Allocation Strategies

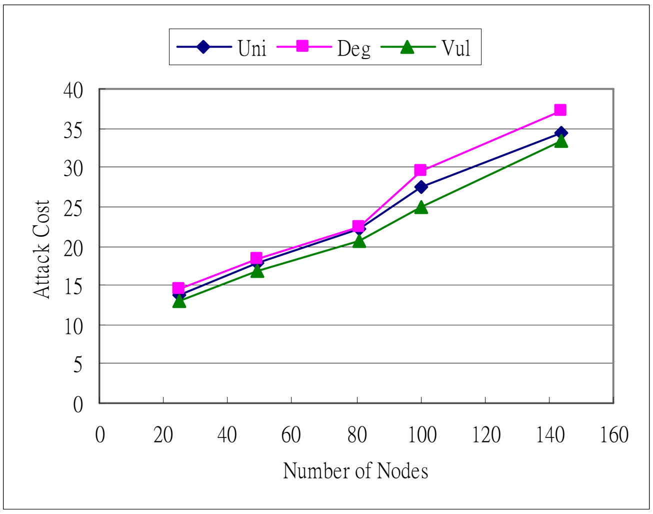

Figure 4-12 Total Attack Costs of Random Networks under Different Defense Budget Allocation Strategies

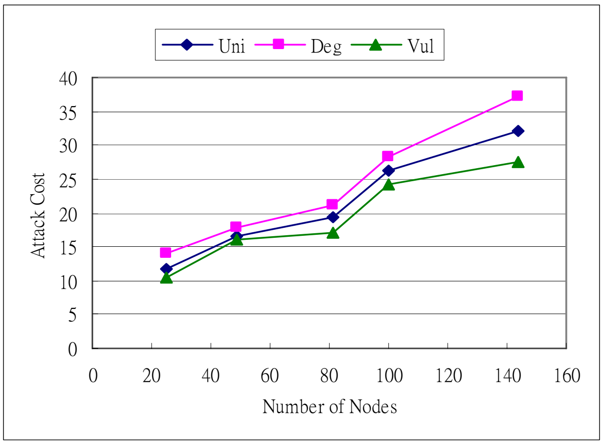

Figure 4-13 Total Attack Costs of Scale-free Networks under Different Defense Budget Allocation Strategies

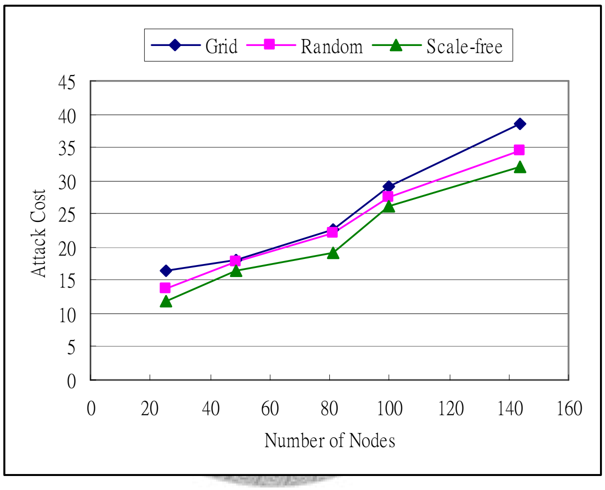

Figure 4-14 Total Attack Costs of Uniform Defense Budget Allocation Strategies under Different Networks

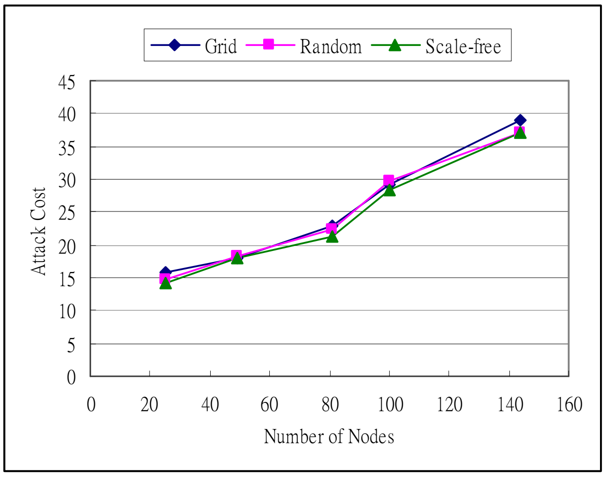

Figure 4-15 Total Attack Costs of Degree-based Defense Budget Allocation Strategies under Different Networks

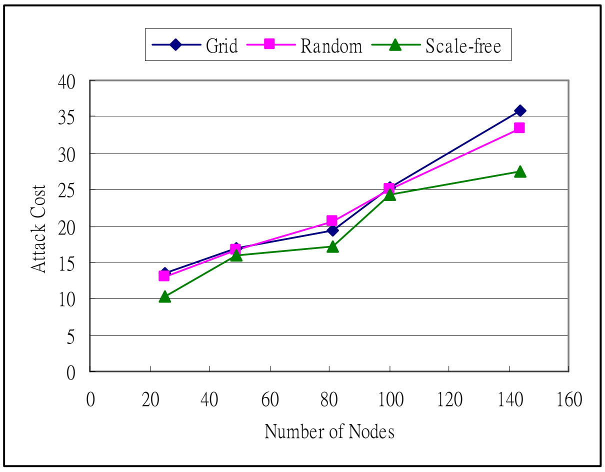

Figure 4-16 Total Attack Costs of Vulnerability-based Defense Budget Allocation Strategies under Different Networks

4.1.4 Discussion of Results

Tables 4-3 to 4-4 and figures 4-3 to 4-7 show the quality of our solutions under the target networks. Because it is unpractical to compare our approaches under some general networks with exhaustive search, we could use these special cases as an alternative measurement. From these figures and tables, we observe:

- The error rate of our solutions of these particular experiment networks is approximate under 0.1%. Besides, from Table 4-3, we can find that the results of our solutions under small networks with 9 nodes and 1 escalation level are all the same with the results of exhaustive search under these networks. These results show that our approaches under these networks could obtain near-optimal solutions. It also indicates that the quality of our solutions under these scenarios is quite good.

Figures 4-8 to 4-10 compare the quality of the proposed SA-based algorithms with simple algorithm 1 (*S1*) under the degree-based budget allocation strategy in different networks.

- In all kinds of network topologies, our heuristics perform better than simple algorithm 1 obviously. The *S1* also performs well in small size networks with 25 nodes, but when the network becomes large, the gap between our solutions and *S1* increases in most cases.
- Generally, the total attack cost increases with the growth of the networks. It shows a monotone increase. This is due to the growth of attack tree. When the networks become large, the attack path towards each core node would also become more complex.
- On average, the quality of the results of the approach *TSA_Weight* is better than other approaches. But the variations of the three proposed approaches are quite slight. Since these approaches are only different in initial solutions, it might be concluded that the initial solution of SA in this problem is not a very vital factor affecting the quality of the solutions.

Thus, we use the approach *TSA_Weight* as the solution approach in the following comparisons. For Figures 4-11 to 4-13, we could observe several trends.

- Networks with degree-based defense budget allocation strategy are the most robust. It means that the attacker who wants to compromise these networks needs to pay the most cost. In other words, it is the most difficult for the attacker to compromise. This finding is reasonable. The defense budget should allocate to the vital nodes in the networks, i.e., budget should be allocated according to the importance of each nodes. This allocation is also based on the characteristics of networks. If a node with more connectivity, it may be also a shortcut in a network. Thus, the attacker could use this node to reach his targets more quickly. According to this, if the defender protects these kinds of nodes more, it would become more difficult to reach the target nodes from the source node for the attacker.
- For grid networks, the robustness of the degree-based and uniform defense budget allocation strategies is close. That is because the characteristic of grid networks that the degree of each node is almost the same except the edge nodes. Thus, the budget allocated to each node under uniform and degree-based strategy is similar.
- Therefore, we could observe the effect of allocating budget in accordance with the characteristics of network topologies more clearly in scale-free

networks. We could find that the degree-based defense budget allocation is obviously more robust than other two budget allocations in the scale-free networks. This is due to the characteristics of networks we discussed above. If the defender allocated more defense budget in those higher degree nodes, the attacker would loss the shortcuts to the target nodes. Therefore, the degree-based budget allocation strategy could make scale-free networks more difficultly to compromise.

- The vulnerability-based defense budget allocation is the most vulnerable way to protect all kinds of networks. That is because this allocation strategy is based on the fragility of a node but network. The attacker's objective is to compromise some targets, so he would need to compromise some other nodes first to reach his goal. It means that the attacker would consider the attack strategy in a more general way. For instance, if a node is very vulnerable, it may be allocated much budget in this budget allocation strategy. But it may be meaningless. If the node is on the edge of the network and the attacker could also reach his goal without compromising it. Thus, the budget allocated to the node is wasteful. So, it is reasonable that this budget allocation strategy is the most vulnerable.

For Figures 4-14 to 4-16, we could also find some results:

- Under uniform defense budget allocation strategy, scale-free networks are less robust and easier to attack. On the other hand, grid networks are the most robust under this strategy. This finding is consistent with the results we mentioned above. The reason is also due to the characteristics of networks.
- Degree-based budget allocation causes similar total attack cost in these three kinds of networks. This result also shows that the degree-based allocation could provide similar defense level in all networks. Again, it is because this allocation could defend the networks according to the characteristics of the networks.

The experimental execution elapsed time is presented as Table 4-10. All the numbers showed in Table 4-10 represent the total elapsed seconds while executing our approach coded by us.

- Although the time complexity of the proposed two-phase SA-based approaches is the same with the only one-phase SA procedures, the two-phase approaches would need more time to obtain a solution, but these approaches could improve about 0-5 % of the quality of the solutions. Thus, it may be a trade-off for the network operator. If the quality of the solutions is a strict demand for the operator and he has adequate resources to adopt the two-phase approaches, he could choose these kinds of approaches to decide his defense budget allocation strategies. On the other hand, if the quality is not restricted strictly, it may be sufficient for the defender to use the one-phase SA-based approaches. This is because the difference of the solutions between the one-phase SA-based approaches and two-phase SA-based is slight.

4.2 Computation Experiments with the DRA Model

4.2.1 Experiment Environment

The proposed algorithms for the DRA Model are coded in C++ and executed on a PC with Intel(R) Pentium 4 3.00GHz CPU and 512MB RAM. The SA parameter α is set to 0.7, and β is set to 1.3. The initial temperature T_0 is initialized to 1.0 and the final temperature is set to T_0/1000. At each temperature, we control the SA to repeat b_0 times, and initialize b_0 to 1000. The iteration counter is set to 50. We randomly assign the experience value and the number of vulnerabilities on each node. We assume that there are three escalation levels on each node.

In the DRA Model, the attacker would try to compromise multiple core nodes using the minimized attack cost. Thus, we use the degree-based defense budget allocation strategy, which is the best of the three given strategies, as the initial budget allocation strategy in this model. In addition, the performance of the proposed approach *TSA_Weight* is better than the performance of other approaches on average. We use the approach as the solution approach for the inner problem of this model to generate attack strategy.

After each attack, the defender would adjust his allocation strategy according to the attack strategy. Here, three reallocation strategies are chosen to adjust the budget allocated to each node. The strategies are uniform, degree-based and sequence-based reallocation strategies which we discussed before.

Table 4-11 Experiment Parameter Settings

Parameters of the DRA Model	
Parameters	**Value**
Testing Topology	Grid networks, Random networks, Scale-free networks
Number of Nodes |N|	25, 49, 81, 100, 144
Total Defense Budget	Equal to 2 Times the Number of Nodes
Initial Budget Allocation Strategy	Degree-based allocation
Budget Reallocation Strategy	Uniform allocation (R_Uni), Degree-based allocation (R_Deg), Sequence-based (R_Se)
Defense Capability $\hat{a}_i(b_i)$	$\hat{a}_i(b_i, V_i) = \ln(\frac{b_i \times M}{V_i} + 1)$, b_i is the budget allocated to node i and V_i is the vulnerabilities on node i, $\forall\, i \in N$
Total Escalation Levels on Each Node	3

4.2.2 Experiment Results

In this experiment, we use *Initial Attack Cost* value as the total attack cost under the degree-based budget allocation strategy, and the value *Opt. Attack Cost* is the total attack cost after the defender adjusting the defense budget allocation strategy. The improvement ratio of *Opt. Attack Cost* to *Initial Attack Cost* is calculated by $\frac{Opt.\ Attack\ Cost - Initial\ Attack\ Cost}{Initial\ Attack\ Cost} \times 100\%$.

Table 4-12 Experiment Results of Networks with 25 Nodes

Network Topology	Initial Attack Cost	Budget Reallocation	Opt. Attack Cost	Improvement Ratio of Opt. Attack Cost
Grid Networks	15.7374	R_Uni	18.5868	18.11%
		R_Deg	18.5078	17.60%
		R_Se	18.4212	17.05%
Random Networks	14.6404	R_Uni	18.1024	23.65%
		R_Deg	20.8437	42.37%
		R_Se	20.1834	37.86%
Scale-free Networks	14.0495	R_Uni	17.3452	23.46%
		R_Deg	19.9577	42.05%
		R_Se	19.5507	39.16%

Table 4-13 Experiment Results of Networks with 49 Nodes

Network Topology	Initial Attack Cost	Budget Reallocation	Opt. Attack Cost	Improvement Ratio of Opt. Attack Cost
Grid Networks	18.2946	R_Uni	22.4249	22.58%
		R_Deg	21.109	15.38%
		R_Se	21.612	18.13%
Random Networks	18.2307	R_Uni	23.3908	28.30%
		R_Deg	25.0837	37.59%
		R_Se	25.4136	39.40%
Scale-free Networks	17.9993	R_Uni	22.057	22.54%
		R_Deg	23.5961	31.09%
		R_Se	24.7593	37.56%

Table 4-14 Experiment Results of Networks with 81 Nodes

Network Topology	Initial Attack Cost	Budget Reallocation	Opt. Attack Cost	Improvement Ratio of Opt. Attack Cost
Grid Networks	22.9025	R_Uni	28.8376	25.91%
		R_Deg	26.107	13.99%
		R_Se	25.6143	11.84%
Random Networks	22.3458	R_Uni	28.7168	28.51%
		R_Deg	31.1373	39.34%
		R_Se	30.2038	35.17%
Scale-free Networks	22.2694	R_Uni	27.5676	23.79%
		R_Deg	28.865	29.62%
		R_Se	29.6259	33.03%

Table 4-15 Experiment Results of Networks with 100 Nodes

Network Topology	Initial Attack Cost	Budget Reallocation	Opt. Attack Cost	Improvement Ratio of Opt. Attack Cost
Grid Networks	29.3864	R_Uni	34.9833	19.05%
		R_Deg	35.8021	21.83%
		R_Se	34.2001	16.38%
Random Networks	29.594	R_Uni	36.0661	21.87%
		R_Deg	38.4503	29.93%
		R_Se	39.5487	33.64%
Scale-free Networks	28.3074	R_Uni	34.5632	22.10%
		R_Deg	36.7125	29.69%
		R_Se	37.8339	33.65%

Table 4-16 Experiment Results of Networks with 144 Nodes

Network Topology	Initial Attack Cost	Budget Reallocation	Opt. Attack Cost	Improvement Ratio of Opt. Attack Cost
Grid Networks	39.0366	R_Uni	46.2397	18.45%
		R_Deg	44.1372	13.07%
		R_Se	43.9814	12.67%
Random Networks	37.128	R_Uni	43.113	16.12%
		R_Deg	47.939	29.12%
		R_Se	47.336	27.49%
Scale-free Networks	37.1385	R_Uni	43.6026	17.41%
		R_Deg	47.7711	28.63%
		R_Se	47.5039	27.91%

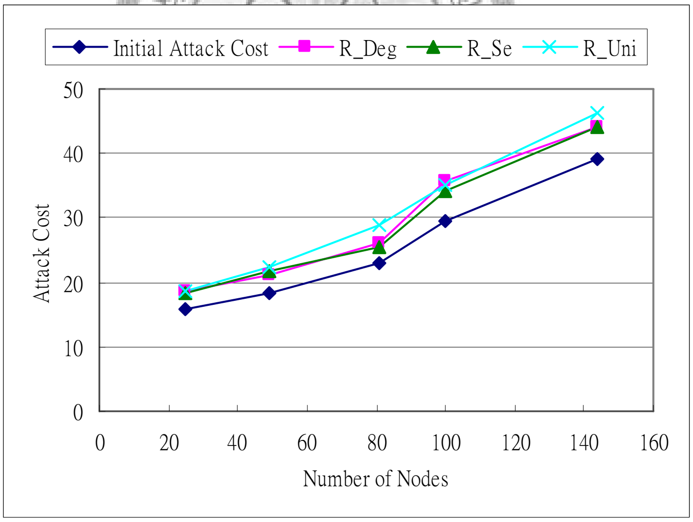

Figure 4-17 Total Attack Costs of Grid Networks under Different Defense Budget Reallocation Strategies

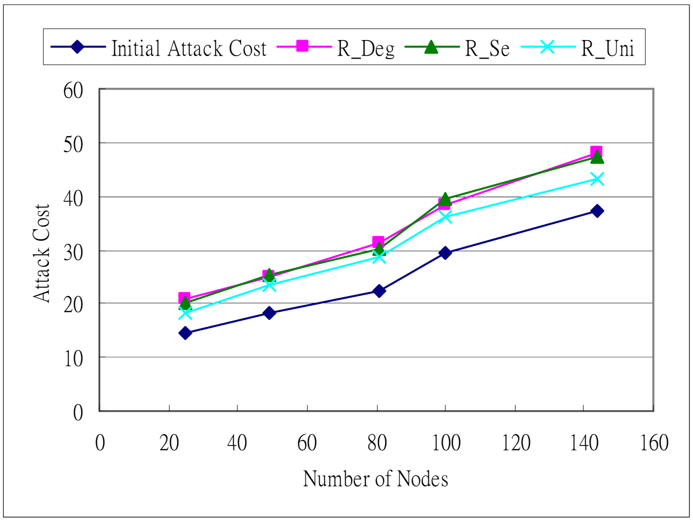

Figure 4-18 Total Attack Costs of Random Networks under Different Defense Budget Reallocation Strategies

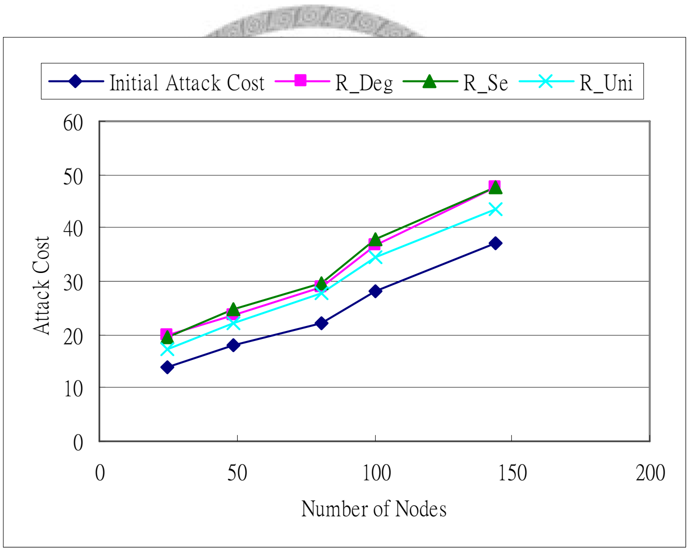

Figure 4-19 Total Attack Costs of Scale-free Networks under Different Defense Budget Reallocation Strategies

4.2.3 Discussion of Results

Figures 4-17 to 4-19 show the cost that the attacker needs to pay after the defender reallocating defense budget under different topologies. From the figures, we can observe:

- The degree-based and sequence-based budget reallocation strategies may be the better choices for the defender to protect the networks because under these two strategies, the attacker may need to pay more attack costs to reach his goal. The sequence-based reallocation strategy would allocate the budget according to the sequence of the attacks, so that the nodes which would be compromised in the beginning are reallocated budget first. In addition, due to the effect of the accumulated experience we evaluated here, the reduced cost of each attack would increase while the accumulated experience becomes more and more. Thus, allocating budget to the nodes near the attack source would be reasonable in this problem. Besides, degree-based reallocation strategy is also a good strategy here. The reason is that the importance of nodes depends on their degree and this is similar to the above discussions. Therefore, the rich get richer, and the poor get poorer may be also a good way to reallocate defense budget.
- Another finding is that the improvement ratio in the grid networks is the smallest of the three network types. This is also due to the characteristics of the grid networks. The attacker can not easily find a shortcut to reach his goal but it also implies that once a path is allocated much budget, the attacker can easily find another path as the substitute path.

- The random and scale-free networks can be more robust than the grid networks by applying our proposed defense resource reallocation strategy. It also indicates that the random and scale-free networks can be very robust if the network defender uses the appropriate budget allocation strategy.
- The uniform reallocation strategy improves less under the random and scale-free networks, but it performs as well as other two strategies under the grid networks. As noted earlier, it is because the grid networks are some regular patterns. Thus, the quality of the three reallocation strategies under grid networks is close.

Chapter 5 Conclusion and Future Work

5.1 Conclusions

The widespread use of the Internet and computer networks brings not only some convenience but also opportunities for network criminals to easily reach their targets. Thus, understanding the attacker behavior is helpful to minimize the damage from network attacks.

In this research, we have addressed the issues of attacker behavior under network defense-attack scenario. We focus on the learning skill of intelligent attackers and how it could help the attackers to reduce their costs in the future. This concept is generalized as a term, experience, in this thesis. Besides, we also modeled the escalation of attackers and evaluated the impact incurred by probing information at different escalation levels. As a result, the attacker would try to minimize the total attack cost under these issues. In response, the network defender would try to maximize the total attack cost by a proper defense budget allocation strategy.

The key contribution of this thesis is the development of a max-min mathematical model which well formulated the interaction between attackers and defenders in the real world and the concept of escalation and the experience of attackers. We have also solved this model by several proposed heuristics. To the best of our knowledge, very little research is done to model the real-world attack behavior in the offense-defense sceneries by this approach.

Another contribution is that we have evaluated the robustness of different network topologies by the minimized total attack cost. In addition, to the measure the robustness on different defense resource allocation strategies, we used uniform, degree-based, and vulnerability-based defense strategies to observe the value of the minimized total attack cost.

Moreover, we have developed an engineering guideline for the network defender. It provides the defender with the information that the best defense budget reallocation strategy should be based on the nodal degree and the attack sequence of nodes on the attack tree. Besides, we also provide the network operator with different approaches for difference considerations. While the operator needs a higher quality solution, he may try to use our TSA approaches. If the operator does not need that kind of solution, he may apply our SA approaches, because the quality of these solutions is also good enough.

5.2 Future Work

In the following, we summarize several issues which could be further researched.

Information issues

We assumed that information probed from each level on a compromised node would not be duplicated. Thus, the experience and the impact of information would be accumulated continuously. By this assumption, an attacker is very skillful and intelligent that he would not pay any unless fee to gain duplicated information. Therefore, we could further discuss the duplicated information issues in our research. Since the information could be measured in a more complex way, impact incurred by information leakage might also be evaluated in some more practical methods. That is we could also consider the effect of duplicated information while measuring the impact of probing information by the attacker.

Experience of attackers and defenders

We considered the experience of attackers by a value between o and 1 which is similar to a discount on each attack cost. This attribute is set in accordance with attack skill which the attackers could learn from each node. However, we should further measure the attack skill the attacker has already accumulated before launching each attack. That is, if the attacker compromises a node in the beginning, he could gain an experience value about 0.5, but in the same situation, if he compromises it after attacking several nodes, he might only gain an experience value about 0.9. What the attacker can learn in the second case is less than that in the first case. That is because the attack skill he could learn from the node might be already learned from other nodes.

In addition, not only attackers but also defenders could learn something from their pervious efforts. In other words, the defenders can gain some experience from protecting nodes. According to these two reasons, we could study the effect of the experience learned by the attackers in different time and the experience of defenders in the future.

References

[1] R. Richardson, "2007 CSI/FBI Computer Crime and Security Survey", *Computer Security Institute*, 2007, http://GoCSI.com.

[2] R.J. Ellison, D.A. Fisher, R.C. Linger, H.F. Lipson, T.A. Longstaff, and N.R. Mead, "Survivable Network Systems: An Emerging Discipline," *Technical Report CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University*, November 1997 (Revised: May 1999).

[3] P. Tarvainen, "Survey of the Survivability of IT Systems," *The 9th Nordic Workshop on Secure IT-systems*, November 2004.

[4] J.C. Knight and K.J. Sullivan, "On the Definition of Survivability," *Technical Report CS-TR-33-00, Department of Computer Science, University of Virginia*, December 2000.

[5] Y. Liu and K.S. Trivedi, "A General Framework for Network Survivability Quantification," *Proceedings of the 12th GI/ITG Conference on Measuring, Modeling and Evaluation of Computer and Communication Systems*, September 2004.

[6] J.C. Knight, E.A. Strunk, and K.J. Sullivan, "Towards a Rigorous Definition of Information System Survivability," *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2003)*, Volume 1, pp.78-89,

April 2003.

[7] R.J. Ellison, D.A. Fisher, R.C. Linger, H.F. Lipson, T.A. Longstaff, and N.R. Mead, “An Approach to Survivable Systems,” *Technical Report CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University*, 1999.

[8] http://technet.microsoft.com/en-us/windowsserver/default.aspx

[9] M.N. Azaiez, V.M. Bier, “Optimal Resource Allocation for Security in Reliability systems” *European Journal of Operational Research*, 181 pp. 773-786, 2007.

[10] Z. Yongzheng and Y. Xiaochun, “A New Vulnerability Taxonomy Based on Privilege Escalation,” *Proceedings of the 6th International Conference on Enterprise Information Systems*, 2004.

[11] M.A. McQueen, W.F. Boyer, M.A. Flynn, G..A. Beitel, “Quantitative Cyber Risk Reduction Estimation Methodology for a Small SCADA Control System” *Proceeding of IEEE Internation Conference on System Sciences*, 2006.

[12] O.M. Alhazmi, Y.K. Malaiya, “Quantitative Vulnerability Assessment of Systems Software” *Proceeding of IEEE Reliability and Maintainability Symposium*, pp. 615-620, Jan 2005.

[13] B. Brykczynski, R.A. Small, “Reducing Internet-Based Intrusions: Effective Security Path Management,” *IEEE Software,* Volume 20, No. 1, pp. 50-57, January

2003.

[14] S. Bistarelli, F. Fioravanti, P. Peretti, "Defense Tree for Economic Evaluation of Security Investments," *Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*, 2006.

[15] C. Iheagwara "The effect of intrusion detection management methods on the return on investment," *Computers & Security*, Volume 24, Number 3, pp. 231-228, 2004.

[16] F. Harmantzis and M. Malek, "Security Risk Analysis and Evaluation," *Proceedings of IEEE International Conference on Communications ,*Volume 4, pp. 1897-1901, June 2004.

[17] V.R. Westmark, "A Definition for Information System Survivability," *Proceedings of the 37th IEEE Hawaii International Conference on System Sciences*, Volume 9, p. 90303.1, 2004.

[18] S.C. Liew and K.W. Lu, "A Framework for Network Survivability Characterization," *IEEE Journal on Selected Areas in Communications*, Volume 12, Number. 1, pp. 52-58, January 1994 (ICC, 1992).

[19] E.Jonsson and T. Olovsson, "A Quantitative Model of the Security Intrusion Process Based on Attacker Behavior", *IEEE Transactions of Software Engineering*, Volume 23, Number 4, pp. 235-245, April 1997.

[20] J. McDermott, “Attack-Potential-Based Survivability Modeling for High-Consequence System,” *Proceedings of the Third IEEE International Workshop on Information Assurance (IWIA’05)*.

[21] S. McClure, J. Scambray, G. Kurtz, Hacking Exposed Network Security Secrets and Solutions, *ISBN:9780072260816*.

[22] X. Song, M. Stinson, R. Lee, P. Albee, “An Approach to Analyzing the Windows and Linux Security Models” *Proceeding of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Architecture and Resuse (ICIS-COMSAR’06)*, pp. 56-62, 2006.

[23] S. Kirkpatrick, C.D. Gelatt, Jr., M.P. Vecchi, “Optimization by Simulated Annealing”, *Science*, Volume 220, Number 4598, pp. 671-680, May 1983.

簡歷

姓名：陳奐廷

出生地：台灣 高雄市

生日：中華民國七十二年十月八日

學歷：九十一年九月至九十五年六月

國立中央大學資訊管理學系學士

九十五年九月至九十七年七月

台灣大學資訊管理研究所碩士