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

考量誘捕系統下攻擊者成功機率最小化之近似最佳化防禦策略

Near Optimal Defense Strategies to Minimize Attackers' Success Probabilities for Networks of Honeypots

王猷順

Yu-Shun, Wang

指導教授：林永松 博士

Advisor: Yeong-Sung Lin, Ph.D.

中華民國 98 年 7 月

July, 2009

謝誌

本論文得以完成，首先要感謝恩師林永松教授在研究過程中細心而精闢的指導，不論在研究方向或是細節部分，都提供許多獨到的見解。再者萬分感謝本校電機學系鐘嘉德教授、輔仁大學資訊工程學系呂俊賢教授、國立高雄第一科技大學行銷與流通管理學系傅新彬教授，以及國立交通大學資訊工程學系林盈達教授在百忙中惠予擔任口試委員，並悉心指導，給予寶貴的意見，方使本論文臻於完善。

接著要感謝實驗室的明宗學長，使研究在萌芽初期，能以更全面性的觀點詮釋。感謝霈語學姐的指引，使文獻探討的部分更加完整。更要感謝柏皓學長，不論在何時，都以專業的角度，提供許多實務的經驗，當我遇到瓶頸時，更適時的給予協助，對於本研究有莫大的助益。

此外，同學峻韋、宴毅、培維、友仁、睿斌以及冠瑋在研究所學習階段中，與我一起切磋學業互相扶持，是我學習路上的最佳良伴。以及學弟怡緯、耀元、世昌、永斌的協助，處理許多論文口試時的相關事宜，在此一併致謝。

最後要感謝我親愛的家人，父母親無微不至的關懷與支持、二位姊姊的肯定與陪伴，讓我能在親情的鼓舞與扶持下毫無後顧之憂的全力撰寫論文，與我共同成長。謹將此研究成果敬獻給各位協助我、關心我的親友、師長、同學們一起分享，並致上無限的感激之意！

王猷順　謹誌

于台灣大學資訊管理研究所

中華民國九十八年七月

論文摘要

論文題目：考量誘捕系統下攻擊者成功機率最小化之近似最佳化防禦策略

作者：王猷順 **九十八年七月**

指導教授：林永松 博士

由於攻擊者的手法與策略日新月異，對於防禦者而言，網路系統時常被不同類型的攻擊者同時攻擊，因此，如何衡量系統在此種情境下的存活度是防禦者的首要任務。除此之外，從攻擊者的角度而言，其對於欲攻擊的目標通常僅具部分資訊，即「不完美知識」。有鑒於此，發展出了一種欺騙攻擊者與消耗其資源的防禦機制，稱為誘捕系統。該系統除了具備上述的重要功能之外，還可用於學習攻擊者技巧並記錄其所使用之系統漏洞，以降低核心節點被攻克的機率，增進整體系統的存活度。

在本論文中，我們將一個攻防情境轉化成數學規劃問題，用以描述系統被攻擊者攻克的機率，並且透過「評估流程」找出能讓該機率最小化之防禦資源配置模式。該法是利用一連串的評估以及策略強化逐步地提升解的品質，並在每一次的循環中，藉由現有的資訊推導出最適當的修正方向，持續的強化現有的配置方法，以期求得最佳解；此外，該法能夠用於解決具備不完美資訊特質的問題，透過適當的情境描述，加入隨機的變異性情況，使問題更貼近於真實情況，有效地提升對防禦者的正面效益。

關鍵詞：網路攻防、網路存活度、最佳化、資源配置、數學規劃、誘捕系統、不完美知識

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT NATIONAL TAIWAN UNIVERSITY

NAME：YU-SHUN WANG MONTH/YEAR JUL/2009

ADVISER：YEONG-SUNG LIN

Near Optimal Network Defense Strategies to Minimize Attackers' Success Probabilities for Network of Honeypots

Since the attack level and tactics of network systems grow with each passing day. Network systems are usually simultaneously attacked by different types of attackers. Therefore, the most important issue for defenders is to evaluate the system survivability under this scenario. Besides, from the view point of attackers, they usually only have partial information of the targeted system. In other words, they only have "imperfect knowledge". As a result, a mechanism which is capable to distract attackers and waste their budget is emerged. This defense technique, called honeypot, can not only assist defender to learn attack strategy and record system vulnerabilities attackers used but also allows defender to understand system vulnerabilities. Therefore, whole system compromised probability is reduced. In other words, survivability is raised.

In this thesis, we model the attack defense scenario as a mathematical programming problem that describes attackers’ success probability. The optimal defense resource allocation is discovered by evaluation process. This approach applies a serious of evaluations and policy enhancements gradually improve the quality of solution. For each round, we derive the most appropriate direction to amend and continually enhance the allocation scheme to achieve optimal solution. Besides, this approach can be applied to solve problems with imperfect knowledge property. Through appropriate scenario description and randomness involved, the problem can be closer to realistic, thus enhance the positive benefits effectively for the defenders.

Keywords: Network Attack and Defense, Network Survivability, Optimization, Resource Allocation, Mathematical Programming, Honeypots, Imperfect Knowledge

Table of Contents

謝誌 I

論文摘要 II

THESIS ABSTRACT III

Table of Contents V

List of Tables VII

List of Figures VIII

Chapter 1 Introduction 1

1.1 Background 1

1.2 Motivation 5

1.3 Literature Survey 11

1.3.1 Survivability 11

1.3.2 Analytical Mechanisms toward Security Problems 15

1.3.2.1 Worst Case Analysis 15

1.3.2.2 Average Case Analysis 18

1.4 Proposed Approach 22

1.5 Thesis Organization 23

Chapter 2 Problem Formulation 24

2.1 Problem Description 24

2.2 Problem Formulation 26

2.3 Attacker Classification and a Possible Scenario 28

2.3.1 Attacker Classification 29

2.3.2 A Possible Scenario 35

Chapter 3 Solution Approach 44

3.1 Evaluation Process 44

3.2 Policy Enhancement 49

3.3 Initial Allocation Scheme 59

Chapter 4 Computational Experiments 63

4.1 Experiment Environment 63

4.2 Experiment Result 65

4.3 Scenario Analysis 83

Chapter 5 Conclusion and Future Work 93

5.1 Conclusion 93

5.2 Future Work 94

References: 98

List of Tables

Table 1-1: Percentages of Incidents [1] 5

Table 1-2: Survivability Definition Summary........ 13

Table 1-3: Honeypot Definition Summary 20

Table 2-1: Problem Description 25

Table 2-2: Given Parameters and Decision Variables 26

Table 2-3: Problem Assumptions 36

Table 2-4: Given Parameters and Decision Variable of a Possible Scenario 41

Table 3-1: Initial Resource Allocation Algorithm 61

Table4-1: Parameters for Defender 63

Table4-2: Parameters for Attacker 64

Table4-3: System Parameters 65

Table4-4: Detailed Data when Benchmark is 443 72

Table4-5: Detailed Data of Enhanced Rate when Benchmark is 443 73

Table4-6: Detailed Data when Benchmark is 480 75

Table4-7: Detailed Data of Enhanced Rate when Benchmark is 480 76

Table4-8: Detailed Data when Benchmark is 515 78

Table4-9: Detailed Data of Enhanced Rate when Benchmark is 515 79

List of Figures

Figure 1-1: Number of Targeted Attacks [1] 3

Figure 1-2: Percentages of Key Types of Incident [1] 3

Figure 1-3: Exploits for Client-Side Vulnerabilities [9] 4

Figure 2-1: The Attacker Classification 34

Figure 2-3: Defense Resource Allocation.. 38

Figure 2-2: Initial State. 38

Figure 2-4: Probing Nodes 38

Figure 2-5: Node Selecting... 38

Figure 2-6-1-a: Successfully Cheat Attacker.. 39

Figure 2-6: False Target Honeypots 39

Figure 2-6-2-b: Total Attack Path of Penetrating False Target Honeypot. 39

Figure 2-6-2-a: Attacker Penetrate False Target.. 39

Figure 2-7: Explaination of components 40

Figure 3-1: Process for Getting Optimal Solution 48

Figure 3-2: The Possible Decisions of Tuning Method.. 56

Figure 3-3: The Operation of Policy Enhancement... 57

Figure 3-4: Relation between Policy Enhancement and Whole Process.................. 58

Figure 4-1: Experiment Result on M with 10 Chunks 66

Figure 4-2: Experiment Result on M with 100 Chunks 67

Figure 4-3: Experiment Result on M with 1000 Chunks 67

Figure 4-4: Experiment Result on M with 10000 Chunks 68

Figure 4-5: Relationship of Minimum Attack Cost and Percent of Applying Number of Hops in Initial Allocation 69

Figure 4-6: Performance of Initial Allocation when Benchmark is 466 70

Figure 4-7: Performance Comparison when Benchmark is 443 71

Figure 4-8: Enhanced Rate when Benchmark is 443 73

Figure 4-9: Performance Comparison when Benchmark is 480 74

Figure 4-10: Enhanced Rate when Benchmark is 480 76

Figure 4-11: Performance Comparison when Benchmark is 515 77

Figure 4-12: Enhanced Rate when Benchmark is 515 79

Figure 4-13: Topology and node index 81

Figure 4-14: Example of resource allocation when honeypots are located on node 6 and 9.. 81

Figure 4-15: Performance Comparison when honeypots are located on node 6 and 9 81

Figure 4-16: Performance Comparison when honeypots are located on node 4 and 7 82

Figure 4-17: Example of resource allocation when honeypots are located on node 4 and 7.. 83

Figure 4-13: Basic Information of Network 85

Figure 4-14: Initial Allocation of Network 85

Figure 4-15: Next Hop Selection 85

Figure 4-16: Compromising a False Target Honeypot.. 85

Figure 4-17: Compromised the False Target Honeypot.. 86

Figure 4-18: Result of Attackers Apply Lowest Defense Level as Next Hop Selection Criteria 86

Figure 4-19: Next Hop Selection on Enhanced Allocation.. 87

Figure 4-20: Assume Attackers Choose the Middle One 87

Figure 4-21: Successfully Penetrate the False Target Honeypot................................ 87

Figure 4-22: Result of Attackers Apply Highest Defense Level as Next Hop Selection Criteria.. 87

Figure 4-23: Information of Enhanced Allocation 89

Figure 4-24: Trap to Lure Attacker to False Target Honeypot 89

Figure 4-25: Attackers Have Sufficient Budget to Compromise the Node.............. 89

Figure 4-26: No Neighbors for Attackers to Select.. 89

Figure 4-27: Trace Back to Former Node 90

Figure 4-28: Result of Attackers Apply Lowest Defense Level on Enhanced Allocation 90

Figure 4-29: Next Hop Selection on enhanced allocation 91

Figure 4-30: No Candidate to Choose for Next Hop 91

Figure 4-31: Trace Back for Proper Candidate... 91

Figure 4-32: Result of Attackers Apply Highest Defense Level on Enhanced Allocation 91

Figure 4-33: Another Attack Path 92

Figure 4-34: Result of the Other Attack Path on Enhanced Allocation 92

Chapter 1 Introduction

1.1 Background

Due to the Internet, there are wide varieties of different applications that make our life much more convenient, for example, electronic commerce. However, with the increasing dependence on the Internet, cyber criminals can target the system with the highest connectivity in order to denial its service. Moreover, by compromising the most important server, some malicious attackers can gather high sensitive information such as trading information of Electronic Commerce (EC) in a firm. Therefore, chief of security officer should develop more effective mechanisms to face this challenge.

According to 2008 CSI computer crime and security survey [1], Robert Richardson, the CSI director, finds out that there are still numerous attacks focusing on one subject. These kinds of attacking behaviors are called target attack. Here, Robert uses a fairly broad definition of "target attack", which it was defined as a malware attack aimed exclusively at the organization or at a small subset within the organization, such as departments within a specific domain or industry. Under this definition, the author constructs the following figure. In figure 1-1, red bar represents statistical data collected during 2008, while yellow bar means for data in 2007. We can see that 32 percent of respondents who replied to the question about targeted attacks expressed that at least some of those incidents involved targeted attacks. The statistics number is slightly decreased in 2008 which is 27 percent. However, it is clear that targeted attacks are a significant reality today.

Moreover, after attackers successfully compromised those targets, the attacked organization will get lots of lose no matter on monetary or reputation. On monetary view, in [1], the author states that "*the most expensive kind of incident on average was financial fraud, with an average reported cost of $463,100, followed by dealing with "bot" computers within the organization's network, reported to cost an average of $345,600 per respondent. As a point of interest, dealing with loss of either proprietary information or loss of customer and employee confidential data averaged at approximately $241,000 and $268,000, respectively.*" Further, this survey also contains the analysis of major threat types. In figure 1-2, the data are collected since 1999. We can see that the main incident type in 2008 is virus and followed by inside abuse, laptop theft/fraud and unauthorized access. Except DNS, all types are trending to decrease, but this figure only includes a subset of computer attacks. The full categories of incidents surveyed in [1] are listed in table 1-1. It shows that only four categories with increased percentages including unauthorized access, theft/loss of proprietary information, misuse of web application and DNS attacks.

After discussing the threats that an organization may encounter, we know a fact that it is almost impossible to prevent our system from all categories of malicious attacks. In such way, we need define a metric to evaluate the system performance under malicious attacks. Survivability is a typical one for measuring system status since this concept is rapidly appeared in many literatures [2] [3] [4] [5] [6] [7] [8].

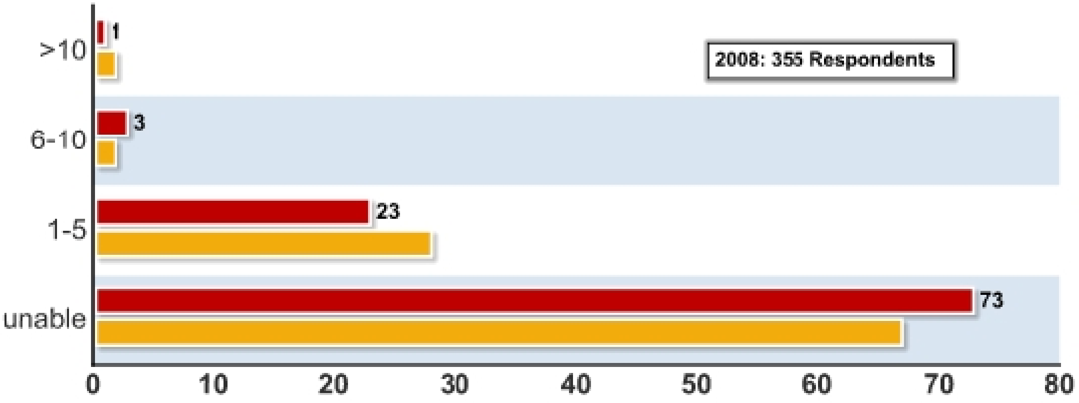


Figure 1-1: Number of Targeted Attacks [1]

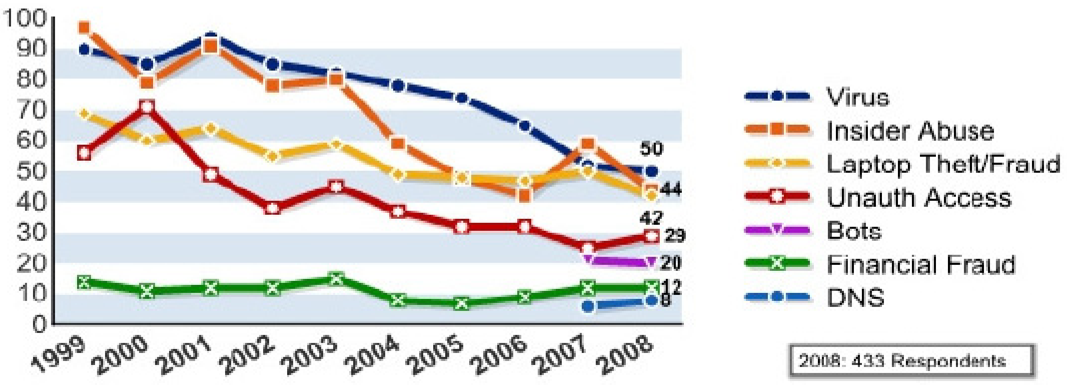


Figure 1-2: Percentages of Key Types of Incident [1]

Another difficulty for defenders is that they face attackers with distinct strategies. Consequently, the defender's main objective is to allocate finite resource to achieve the best system survivability. Besides, attackers usually use 0 day exploits to threaten the computer systems. Defenders' reacting time is almost compressed within one day

[9]. In figure 1-3, the 2008 data reflected situation during first half year, and we can discover that over 80 percent of these exploits are released on the same day or even before the official vulnerability disclosure. Therefore, it is a great challenge to detect, even prevent an attack before the patch of the vulnerability attacker exploited is released.

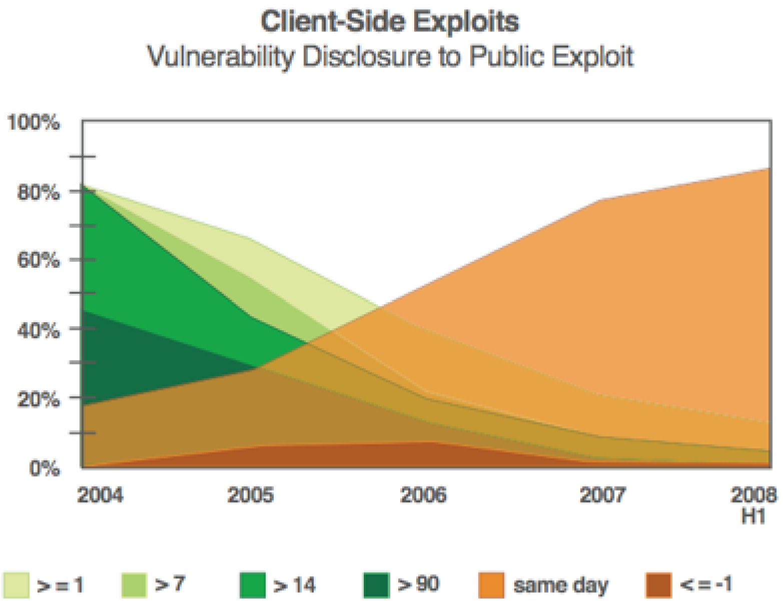
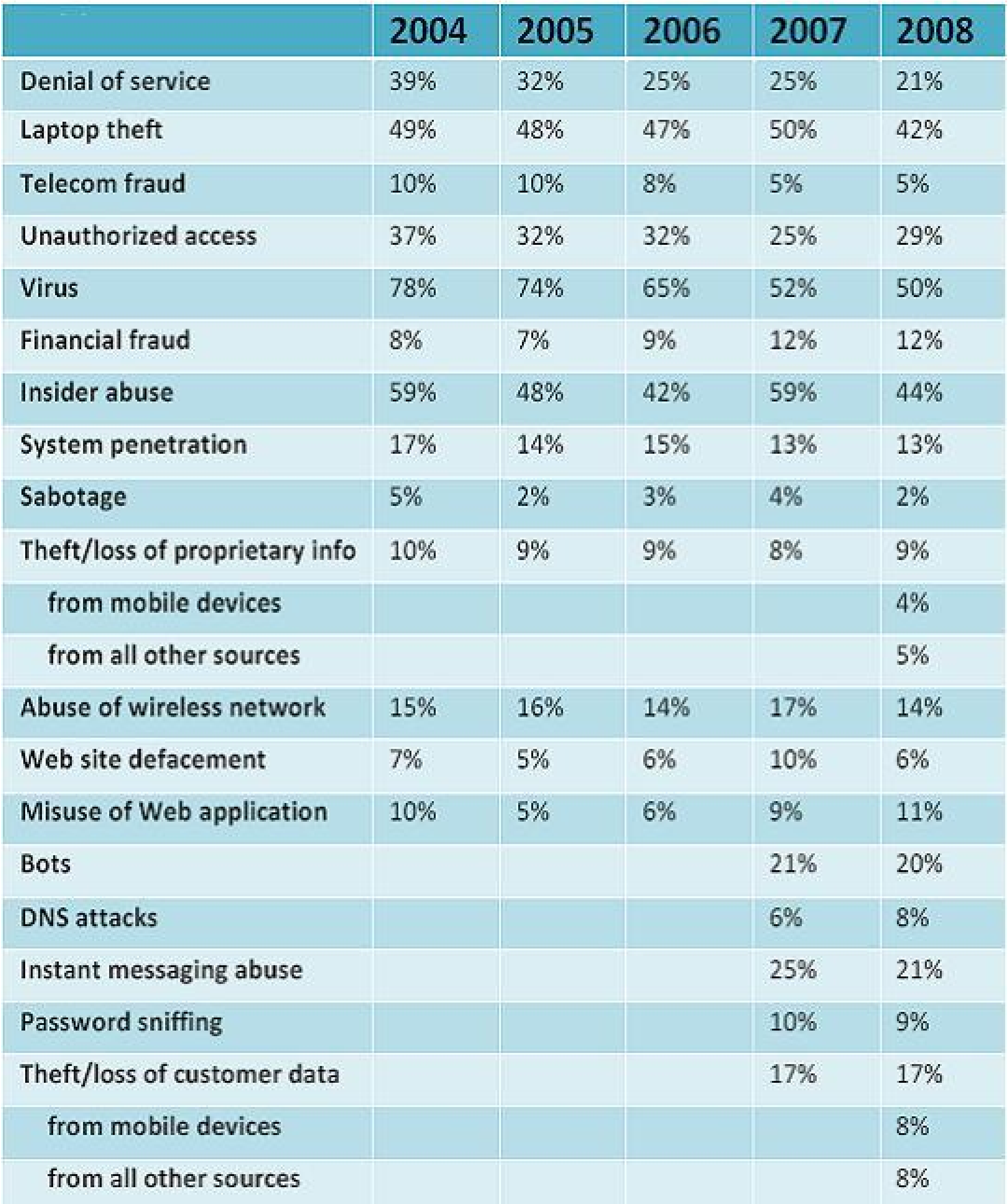


Figure 1-3: Exploits for Client-Side Vulnerabilities [9]

Table 1-1: Percentages of Incidents [1]



	2004	2005	2006	2007	2008
Denial of service	39%	32%	25%	25%	21%
Laptop theft	49%	48%	47%	50%	42%
Telecom fraud	10%	10%	8%	5%	5%
Unauthorized access	37%	32%	32%	25%	29%
Virus	78%	74%	65%	52%	50%
Financial fraud	8%	7%	9%	12%	12%
Insider abuse	59%	48%	42%	59%	44%
System penetration	17%	14%	15%	13%	13%
Sabotage	5%	2%	3%	4%	2%
Theft/loss of proprietary info	10%	9%	9%	8%	9%
from mobile devices					4%
from all other sources					5%
Abuse of wireless network	15%	16%	14%	17%	14%
Web site defacement	7%	5%	6%	10%	6%
Misuse of Web application	10%	5%	6%	9%	11%
Bots				21%	20%
DNS attacks				6%	8%
Instant messaging abuse				25%	21%
Password sniffing				10%	9%
Theft/loss of customer data				17%	17%
from mobile devices					8%
from all other sources					8%

1.2 Motivation

Science the complexity and attack level of network systems grow with each passing day, we need more solutions to deal with various threats from the present and future. Although there are already many different approaches to increase system security, defenders are still in a passive position. As a result, in this paper, we not only consider general defense resource (e.g. firewall, IDS, IPS, and so on) but also another kind of defensive technology, honeypot, as a deceptive tool to distract attackers. With this security tool, attackers may believe they are successes in compromising the server, even the core node, but in fact, they just only wasted their attack resource, and dropped into a trap set by defender.

Generally speaking, the honeypot is also called deception-based mechanism. It not only has been applied in real world for years but also has made into packages as commercial products in security domain [10]. In academic community, the first concept of honeypot was introduced in computing systems by Clifford Stoll in the late 80's [11]. In the "Cuckoo's Egg", he states the method about tracking and monitoring of an intruder [11]. After several years, the initial honeypot deployment on a simulating environment was done in 1991 by Cheswick in his account of tracking the Dutch hacker Berferd [12]. Since then, the concept of honeypot has been in continuous progressing, many different taxonomies and applications are gradually appeared [13] [14] [15].

In industry field, there is software that implements the idea of honeypot. For example, in [10], the design goal of the system is to record every activity done by attacker within it. By doing so, defender can learn more information about attacking strategy. This will help defender refine his policy to cope with malicious attackers. Moreover, it also provides a function for administrator to response immediately while system is probably under attack; that is, when a honeypot is successfully logged in, it will shut down itself instantly. The main purpose of this capability is to avoid the honeypot becoming attacker's springboard to compromise other systems in the network. However, from the learning view of honeypots, this setting will eliminate the chance to gather information about attacker's strategy. Therefore, defenders should give careful consideration on this decision.

Recent studies also agree the feasibility of honeynet which is composed by many honeypots applying in real systems. For example, in [16], Dimitriadis first analysis the vulnerabilities of 3G operator's architecture, then use the game theory to assay whether both an operator and his roaming partners will get benefit from deploying honeynet. The author defined a game called 3GHNET-G that is non-cooperative—the mobile operators don't have a common security infrastructure, and the game is static because players can make simultaneous moves. Also, this is a non-zero sum game, meaning that the total benefit to all players isn't zero because there's no relationship between one player's gain and another's loss. For the two players, one implements a honeynet architecture, and the other doesn't. Each player has two possible modes of behavior, which is normal or compromised. The result reveals that the implementation of a honeynet is useful to both players in accordance with each other. In other words, if the two players in this game both implemented honeynet architecture, they get the best player response, or namely, the Nash Equilibrium.

The corresponding literature survey is proposed in 1.3.2.2, here, we only introduce the basic concept of honeypots.

Although the use of honeypots can effectively improve whole system survivability, defenders usually have a budget limitation, it is hardly to find a situation that the defenders can arbitrarily deploy all kinds of resource without any restrictions. Furthermore, at the attackers' point of view, each attacker may have different budget level. Therefore, the damage caused by different attacker will be varied. The corresponding applicable defense strategy should also be different. Consequently, how to find a defense resource deployment that averagely has the best performance toward various categories of attackers is not only a practical but also an important issue.

The first thing defenders need to understand before applying any defense strategy is the environmental knowledge, for example, linking behavior between nodes in the network. Recent studies have demonstrated that the Internet and many other complex networks follow a power-law degree distribution, called scale-free networks [17]. Under this characteristic, there are some special nodes/servers which have a great number of connections to other nodes in the network. From the attackers' view, these nodes are ideal candidates to attack because once these servers are compromised; the whole network will get a huge influence on functional operation or information leakage. Therefore, how to properly deploy defense resource on “right” position to achieve the best performance?

To answer this question, we must transfer ourselves thinking into attackers’ way. In [18], Fred Cohen proposes an analysis based on game theory. The main purpose of this article is to find the best choice under different situations. Although the number in this analysis is made up, he claims these are revised from a real world event. However, the most interesting thing for us is not just the numeric result, but the attacker’s and defender’s strategies. For attacker, the author list seven different strategies containing speed, stealth, overwhelming force, indirection, random, least resistance and easiest to find strategy to achieve different goals. The detailed statements about these categories are as follows:

- ✓ **Speed:** These attackers’ main objective is to compromise the network as soon as possible. They choose the next hop only the fastest attacks available. This gives them the advantage that they can win before the defender detects or reacts to their presence.
- ✓ **Stealth:** Some attackers choose to conceal themselves to avoid detection. There are many mechanisms to apply this strategy, for example, one may choose the

node with least link utilization as next attack target since this node is seldom used. Therefore, attack on this node may also be ignored by defenders.

✓ **Overwhelming force:** Some attackers try to generate enough force - typically in the form of physical assault or sheer volume of resources - to overwhelm the defender.

✓ **Indirection (a.k.a. reflexive control):** Some attackers use deceptive techniques to cause the defender to spend resources on the wrong defenses or to cause the defender to act in ways that provide openings to attack.

✓ **Random:** Some attackers just try whatever they happen to come across as an idea on any given day.

✓ **Least Resistance:** Some attackers try to do things they think are the least likely to be defended against and which are the easiest for them to do. In other words, these attackers find next candidate by its defense level. The lower the defense level on one node, the higher possibility it will become next victim.

✓ **Easiest to find:** Some attackers just obtain software from the Internet and try it against many systems.

The above attacking strategies give us some hints in developing the scenario. We will apply some of these in our model.

To summarize the above, we understand that it is hard to discover a defense resource allocation which averagely has an acceptable defense level. Moreover, the average case should not only consider multiple types of defense resources but also multiple types of attackers. Therefore, we want to solve the problem both consider compound attacker types and defense resources. Then, evaluate the performance in an average way.

1.3 Literature Survey

In this section, we review previous works on the survivability and solution approaches toward security problems.

1.3.1 Survivability

Survivability is a metric that describes the performance of a system under malicious attack or other failure. In [8], the authors state that “*We define survivability as the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. We use the term system in the broadest possible sense, including networks and large-scale systems of systems.*” In other words, this measure is mainly describing the ability of the system to perform its task under abnormal situation.

Another concept in this domain that characterizes the hardness of the system is security. These two are very alike at first sight. However, there are some important differences between them. Survivability is focus on service availability and continuity but security is concentrate on system resistance. Besides, from the objective, security is concerning to protect information while survivability is trying to maintain its continuity. Further, the security concept considers systems as closed, bounded and under central administrative control. Nevertheless, survivability treats systems as open, unbounded, with distributed administrative control. From manager's point of view, security is usually considered as an overhead expense and solutions are usually technology-based while survivability is thought as an essential investment of the organization and the solutions are usually management based.

An extra managerial concept that related to survivability is risk management. In fact, there is a reversed relationship between survivability and risk. If a system is in a high risk status, its survivability is low. Meanwhile, if the survivability of the system is high, its risk level is low. From the resource perspective, survivability is a balancing act which is to find balance between the resources to investing and the level of survivability and evaluate the tradeoffs between the budgets of defense mechanisms and the resulting expected survivability after an attack. Other definitions we surveyed are summarized in table 1-2.

After the above depiction, we know that survivability is a measure that wildly used. Therefore, we also take this metric as main performance metric in this paper.

Table 1-2: Survivability Definition Summary

No.	Definition	Researcher(s)	Year	Origin
1	We define survivability as the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. We use the term system in the broadest possible sense, including networks and large-scale systems of systems.	D.A. Fisher, H.F. Lipson, N.R. Mead, R.C. Linger, R.J. Ellison, and T. Longstaff	1997	**[8]**
2	a. Service survivability (i.e., continued service provision in the event of network facility failures) b. Service survivability is defined as the capability to provide un-interrupted services in the event of failures.	D.-P. Hsing, H. Kim, L. Kant, and T.-H. Wu	1999	**[5]**
3	Survivability is a network's ability to perform its designated set of functions given network infrastructure component failures, resulting in a service outage, which can be described by the number of services affected, the number of	A.D. Malloy, A.P. Snow, and U. Varshney	2000	**[4]**

	subscribers affected, and the duration of the outage.			
4	Survivability, the ability of a network to withstand and recover from failures, is one of the most important requirements of networks.	D. Zhou and S. Subramaniam	2000	**[3]**
5	Survivability is used to describe the available performance of a network after a failure.	C. Charnsripinyo, D. Tipper, H. Shin, and T. Dahlberg	2002	**[6]**
6	The survivability of system services is defined as the capability of system services to fulfill the mission objectives in the presence of malicious attacks and/or partial system failures.	C. Fung, J. Lee, M. Anderson, R. Linger, R. Tarquini, X. Wang, and Y.-L. Chen	2005	**[2]**
7	Network survivability is the ability for network to recover traffic that is affected by failures at the node or the link.	D. Botvich, N. Agoulmine, S. Balasubramaniam, and W. Donnelly	2007	**[7]**

1.3.2 Analytical Mechanisms toward Security Problems

To solve security problems, there are two kinds of analysis approaches, worst case analysis and average case analysis. We will discuss their differences in this section.

1.3.2.1 Worst Case Analysis

For this approach, defenders are at the engineering perspective; always think the opponent will choose the best strategy to against the systems. Further, the optimal solution is discovered by mathematical programming. Therefore, in solving these problems, the author(s) usually propose a min-max or max-min mathematical model to describe the defense strategy and the attack strategy precisely. By solving the two-level model optimally, we first know the solution of the inner problem, and then take this result as a feedback of the outer problem. Through this process, defenders can obtain the optimal defense strategy to cope with the attackers applying optimal attack strategy. However, this approach has to base on an important assumption, that is, both defenders and attackers have complete information about the network.

When we claim the attackers have complete information of the network means that they know everything about the network. In other words, even though there are honeypots in the network, attackers will never attack them since it creates no benefit but only raise the total cost of each attack. Therefore, this analytical approach can not apply in the scenario with honeypots because the honeypot itself is an instance that violates the complete information assumption. The detailed reason why honeypots does not fit the complete information assumption will discuss in next subsection.

Further, in general situation, incomplete information is more frequently happened. Although attackers do gather intelligence before they launch an attack, they still rarely get the “complete” information. Generally, partial information is more suitable to describe what attackers have after intelligence gathering. However, assuming attackers hold complete information remains a reasonable viewpoint since it is an engineering aspect which considering the worst case scenario.

For instance, in [19], the authors model the problem about how network operators allocate resource effectively to maximize the survival time of core nodes under attack as a nonlinear, integer programming optimization problem. Then, they propose an effective solution approach based on Lagrangean relaxation and the subgradient method. The objective is to minimize the maximized end to end compromise probability. At attackers’ point of view, they try to maximize the compromise probability by selecting the most vulnerable nodes to attack. As regards defenders, they make an effort to minimize the compromise probability by allocating defense resource to each node.

The evaluating topology of [19] is based on two popular network topologies, grid network and random network. The authors compare the compromise probability of two simple algorithms with the proposed heuristic. The first one is a popularity-based budget allocation strategy that dispenses the budget according to the accumulated compromised frequency of each node on the candidate path. Another one is a greed based budget allocation strategy that first allocates a budget to the node with the smallest compromise probability between the source node and the core node. The result of the experiment shows that the more budget defenders allocate to a grid network, the lower will be the compromise probability of nodes. Unlike grid networks, the compromise probability of random networks cannot be reduced by allocating more budgets. The reason is that, in random networks, there exists a shortest path from the source node to the core node. Even if nodes on this critical path are allocated the maximum budget, an attacker will still choose it as an attack path because the compromise probability of random networks cannot be reduced by simply allocating extra budget. Furthermore, comparing grid networks with random networks under different total budget scenarios, we can see that the compromise probability of random networks is higher than grid networks. This is because grid networks have larger diameters than random networks, so attackers need to go through more hub sites to compromise the core nodes.

The key contribution of [19] is that the authors successfully model the security problem, including concepts like the core node, compromise probability, and survival time, as a well-formulated mathematical problem, which is then solved by the proposed heuristic. Further, the model proposed by them can be extended to different attack-defense scenarios in the context of survivability. For example, it may be stretched into the situation where attackers can devise new attack methods based on previous attack experience so that they can compromise other nodes more easily. Specifically, it is assumed that, for each node compromised, the attacker would obtain a discount coupon, which could be used to increase the compromise probability of nodes subsequently targeted for attack.

1.3.2.2 Average Case Analysis

Instead of assuming attackers have complete information of the network; this approach assumes attackers only have incomplete information of the network. Therefore, we can evaluate the performance of honeypots applied in the network. Before describing the detail, we first introduce the definition of honeypot. There are various editions of it. For instance, in [16], the honeypots are considered as information systems whose value lies in its unauthorized or illicit use. It helps security engineers learn from attacking entities and thus improve existing security architectures and systems. Table 1-2 lists several different definitions proposed by researchers. Although the definitions are diverse, their underlying concepts can be presented as two basic purposes:

- ✓ **False target:**

  This is mainly for distract attackers. We can implement this concept based on legacy systems and put some false data which looks like real sensitive information in them. Sometimes, for enticing attackers, defenders may decrease, even remove, defense resource allocated on it. While attacking, attackers may choose this node to compromise, but even he is successful at this attack, all he can get is nothing but false data [20] [21].

- ✓ **Learning attack tactic and wasting attack resource:**

  The major goal of this purpose is to record every "possible attack behavior" for defenders to analysis whether there are a new attacking tactics or malicious activities on compromising their systems. To achieve this objective, we can simply install the network sniffer on a computer system. However, for more detailed information, we usually implemented with other monitoring tools. After finding out the possible new attacking strategy, defenders can use this information to refine the existing security architecture and increase the survivability of their systems [16] [15] [22].

We can discover that only when attackers do not have complete information, the above two basic purposes can be achieved, since once attackers find a node is a honeypot, they will not choose it as an attacking target anymore. That is the reason why it is hard to handle problems concerning honeypots in worst case analytical way.

Table 1-3: Honeypot Definition Summary

No.	Definition	Researcher	Year	Origin
1	A honeypot is a computer system designed to capture all traffic and activity directed to the system. Most honeypots are designed strictly as a "lure" for would-be attackers. Honeypots differ from regular network systems in that considerably greater emphasis is placed on logging all activity to the site, either by the honeypot itself or through the use of a network/packet sniffer. A honeypot is designed to look like something an intruder can attack to gain access to a given system.	Michael Sink	2001	[22]
2	An Internet-attached server that acts as a decoy, luring in potential hackers in order to study their activities and monitor how they are able to break into a system. Honeypots are designed to mimic systems that an intruder	Barnett	2002	[20]

	would like to break into but limit the intruder from having access to an entire network. If a honeypot is successful, the intruder will have no idea that s/he is being tricked and monitored.			
3	A honeypot is security resource whose value lies in being probed, attacked or compromised.	Lance Spitzner	2002	[15]
4	The value of a honeynet which composed by multiple honeypots lies in its unauthorized or illicit use, which helps security engineers learn from attacking entities and thus improve existing security architectures and systems.	Dimitriadis, C.K.	2007	[16]
5	A honeypot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network but which is actually isolated, (un)protected, and monitored, and which seems to contain information or a resource that would be of value to attackers.	Wikipedia	2008	[21]

After understanding the concept of honeypot, we start to explain what the process of average case analysis is. To reach the average status, we use evaluation as a way toward it. By evaluation, we can detailed describe a scenario that close to reality. For example, when an attacker compromise a node which actually is a honeypot, if he is professional enough, he should aware that it is not a general node. This property is difficult to describe completely in mathematical form, but it is easy to implement in evaluation way. We can just simply add a “detect probability” at each honeypot, while attacking, we use this probability to determine whether the attacker will be deceived or not.

Besides, in real world, defenders usually face more than one type of attackers. However, it is hard to well formulate one mathematical expression that detailed describing every kind of attackers. To solve this problem, we adopt evaluation as a method to deal with this situation.

1.4 Proposed Approach

In this paper, we propose a minimization mathematical model to describe the defense resource allocation problem. By solving this problem, we can discover how to allocate finite defense resource to decrease the compromise probability in average situation.

However, our solution approach is not an optimization based mathematical programming. The reason is, in this paper, we assume attackers only have incomplete information. This characteristic violates the basic assumption of complete information in Lagrangean relaxation. Therefore, to achieve the incomplete assumption, we choose evaluation as our research method. By this method, we can detailed depict a scenario and make it as realistic as possible.

After modeling the problem into an evaluation scenario, the next step is to enhance the performance of defense resource allocation strategy. The key concept of this policy enhancement is based on derivative and attack cost wasted on each node. By executing this procedure iteratively, we can gradually enhance the effectiveness of the resource allocation strategy. Detailed implementation and steps of the procedure is proposed in Chapter 3.

1.5 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, we describe the problem and give a generic model in mathematical expression. Further, the attacker classification and a possible scenario are also given in this section. In Chapter 3, solution approaches to the problem is presented; in Section 3.1, we introduce the way to translate the mathematical model into evaluation; in Section 3.2, a policy enhancement based on evaluation is proposed.

Chapter 2 Problem Formulation

2.1 Problem Description

For improving system security, defenders deploy various defensive resources on different nodes according to their requirements. In general, there are lots of different types of resource to choose, but how to find the allocation with the most effectiveness is still a critical problem.

For attackers, they will adopt different strategies to each specific situation. For instance, if they are hired to steal highly sensitive information of an enterprise, they choose candidate node which is the most likely store these data. If they want to minimize attack cost at each node compromising, they may choose candidate node which lowest defense level. Besides these two strategies, there are still lots of other attack tactics. Therefore, we can see the diversity is wide in node selecting strategy.

To reflect this characteristic, we make the problem generic. In other words, we only obtain some basic information as given parameter and use the information effectively to minimize the compromise probability of the core node in the network under budget constraint. The detailed descriptions are shown in table 2-1.

Table 2-1: Problem Description

Given:

1. The total evaluation time for all attacker categories

2. The ratio of each attacker categories

3. The strategy of an attacker, including his budget, capabilities, and next hop selecting criteria.

Objective:

To minimize the compromised probability of the core node.

Subject to:

Budget constraint both for defenders and attackers.

To determine:

The strategy of defender to allocate defense resources on each node in the network.

The thing deserves to be mentioned is that the attacker categories, K, discussed in this problem is a given parameter. Defenders can set different value under distinct situation. In other words, attackers profile can be very detailed and very realistic. That is why we claim this is a generic model. Further, the ratio of each attacker type is also decided by defender. This value can be set one by one or randomly assigned. All of these features give defenders sufficient flexibility to discover the best resource allocation strategy.

2.2 Problem Formulation

We model the problem of minimizing the core node compromise probability as a mathematical formulation. The given parameter and decision variables are shown in table 2-2.

Table 2-2: Given Parameters and Decision Variables

Control parameter	
Notation	Description
M	The total evaluation frequency for all attacker categories
Given parameter	
Notation	Description
K	The total attacker categories
R_k	Rounded evaluation frequency of each attacker type
P_k	The portion of attacker type k in total attackers (where $k\in K$)
D	All possible defense strategies
$\overrightarrow{A_k}$	The strategy of an attacker, comprising his budget, capabilities, and next hop selecting criteria.
$S_{kj}(\overrightarrow{D}, \overrightarrow{A_k})$	1 if the attacker j of the k^{th} attacker category can compromise the core node under $\overrightarrow{D}$ defense strategy, and 0 otherwise (where $k\in K$)
Decision Variable	
Notation	Description
$\overrightarrow{D}$	The strategy of defender to allocate defense resources on each node in the network.

The control parameter, M, plays an important role in our model since this value will dramatically influence the quality of solution. The detailed explanation and corresponding determining method will be discussed in section 3. 1.

In given parameter, R_k represents a rounded number. This value is originally calculated from M multiple P_k which means the expected value of each attacker type. But this value may not be an integer because the value of P_k is between 0 and 1. Therefore, if we do not perform rounding process, there may be some computational errors in summing operation. To avoid this fault, while computing M multiple P_k, we only take the closest integer which the value is no more than it as the evaluation frequency of each attacker type. That is to say, we set M multiple P_k equals to $\lfloor M \times P_k \rfloor$. Then, the residual frequency, which is M minus $\sum_{k=1}^{K} \lfloor M \times P_k \rfloor$, is distributed to every attacker type by their original decimal value. The larger the original decimal value of one attacker type is; the more residual frequency it will be allocated.

The decision variable $\overrightarrow{D}$ is the defense resource allocation strategy. It includes the configuration of dispensing resource on each node. The category of defense resource can be various. Therefore, $\overrightarrow{D}$ is only a symbol represents the whole allocation scheme of every possible resource category.

The problem is modeled into a mathematical expression as follow.

Objective function:

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

Subject to

$$\overrightarrow{D} \in D \qquad \text{(IP 1.1)}$$

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

Explanation of the mathematical formulation:

- ✓ Objective function: the object is to minimize the system compromise probability. This probability is modeled as system compromise frequency divided by total attack time which is M. The system compromise frequency is govern by $S_k(\overrightarrow{D}, \overrightarrow{A_k})$.
- ✓ Constraint (IP 1.1) requires the defense resource allocation should belong to the feasible strategy which means each allocation needs to fulfill the budget constraint.
- ✓ Constraint (IP 1.2) limits the summation of rounded frequency of each attacker type should equal to M. Otherwise, it will cause inconsistent which may affect accuracy of our model.

2.3 Attacker Classification and a Possible Scenario

In this section, we first introduce the attacker categories considered in the following argument and give a specific scenario as the main problem to solve in this thesis.

2.3.1 Attacker Classification

There are many distinct classification methods for defenders to identify their enemy. The most important thing is the classification can reflect the real world situation. Therefore, there is no uniform classifying standard. Defenders can design his own principle depend on the environment or the defense technology he use. In this thesis, because we consider the deception-based defense technique, honeypot, we then contain the corresponding effect that honeypots caused to attacker in discussing the classification. The following is our classifying process and corresponding result. We apply this outcome to form the attacker set in our problem.

The classification measures are budget, capability, and next hop selecting criteria. As we mentioned before, we included the ideal in [18] to form our next hop selecting criteria. Also, we assume attackers probe the neighbor nodes and gather sufficient information before they launch an attack. The followings are the three measures and corresponding descriptions:

✓ **Budget**

For this criterion, we divide three intervals from the whole possible range to describe an attacker's budget. Here, we use "minimum attack cost" as a baseline to complete our budget classification. The "minimum attack cost" is calculated from defenders' view. After deploying every defense resource, defenders can always calculate one or more path(s) with the lowest cost for attackers to compromise. Thus, the multiple of the "minimum attack cost" becomes our distinction standard. The three intervals are high, medium, and low, described below:

- *High*

  We set high budget level as five times of minimum attack cost or more since attackers may detour in the network by applying different next hop selecting criterion or distracted by honeypots.

- *Medium*

  For the medium budget level, if attackers' budget falls in three times to five times of minimum attack cost, they will be classified into this level.

- *Low*

  At this level, we think the attack budget lower than three times of minimum attack cost should be the low level.

✓ **Capability**

This measure is mainly to describe attackers' professional degree. Followed by above classifying concept, we also divide three intervals to characterize different attackers. But what is the difference between the three levels from defenders' perspective? To answer this question, we first recall a concept about false target honeypot. While attacking the false target honeypot, attackers will terminate this attack with a certain probability since they are cheated by the honeypot. Back to here, we believe attackers with higher professional degree have a lower probability cheated by this kind of honeypot. That is to say, the higher an attacker's professional degree is, the higher probability he can penetrate this defense resource. The three degrees are discussed below:

- *High*

  Attackers at this degree are more skillful than other two categories, so we set the probability attackers be deceived by false target honeypot 30%.

- *Medium*

  The medium degree is the most general case. Most of attackers fall in this level. The probability that attackers be cheated by false target honeypot is 50%.

- *Low*

  These attackers are the least skillful. In fact, they do not master in intruding systems. They can just use some hacking packages to attack other computers. Therefore, we set the probability that they are cheated by false target honeypot is 70%.

- ✓ **Next hop selecting criteria**

This measure is focus on attackers' decision making process about determining next attacking candidate. According to [18], we choose four different criteria from it and add one new strategy, for valuable information strategy, to describe attackers who want to increase the chance of obtaining high value information at each hop compromising. The followings are description about these criteria:

- ➢ *The neighbor which has the highest defense level:*

  This tacit is mainly to get valuable information. Since the defense level is high, attackers may consider this node is an important node. In other words, there may store some sensitive data, for example, customers' privacy data. Therefore, attackers who want critical information prefer to choose this node as next attack target.

- ➢ *The neighbor which has the lowest defense level:*

  While attacking, some attackers are carefully not to be identified. Although this may take longer time to compromise the core node, their objective is to achieve the goal in a silent method. Therefore, we call this strategy as stealth strategy which is also originated from [18]. Because the defense level of one node can reflect its situation, the lower the defense level is, the higher possibility this node is unimportant. Consequently, for attackers who want to hide their tracks, the neighbor with the lowest defense level may be a good candidate to compromise.

- *Randomly choose next hop:*

  This criterion is a possible one in real world. For attackers who only get incomplete information, they may just randomly take a neighbor node as the next target to attack. This tacit is also mentioned in [18].

It is worth to mention that the total number of attacker category can not only be a specific value but also be extended to infinite. If the value of the attacker subclass is described by a probability distribution rather than discrete intervals, we can determine an attacker type by using a random number pointed to the distribution. The corresponding value pointed by this random number becomes the attribute of this attacker type. This feature makes our model more flexible.

The above classification is summarized into a hierarchical structure and illustrated in figure 2-1.

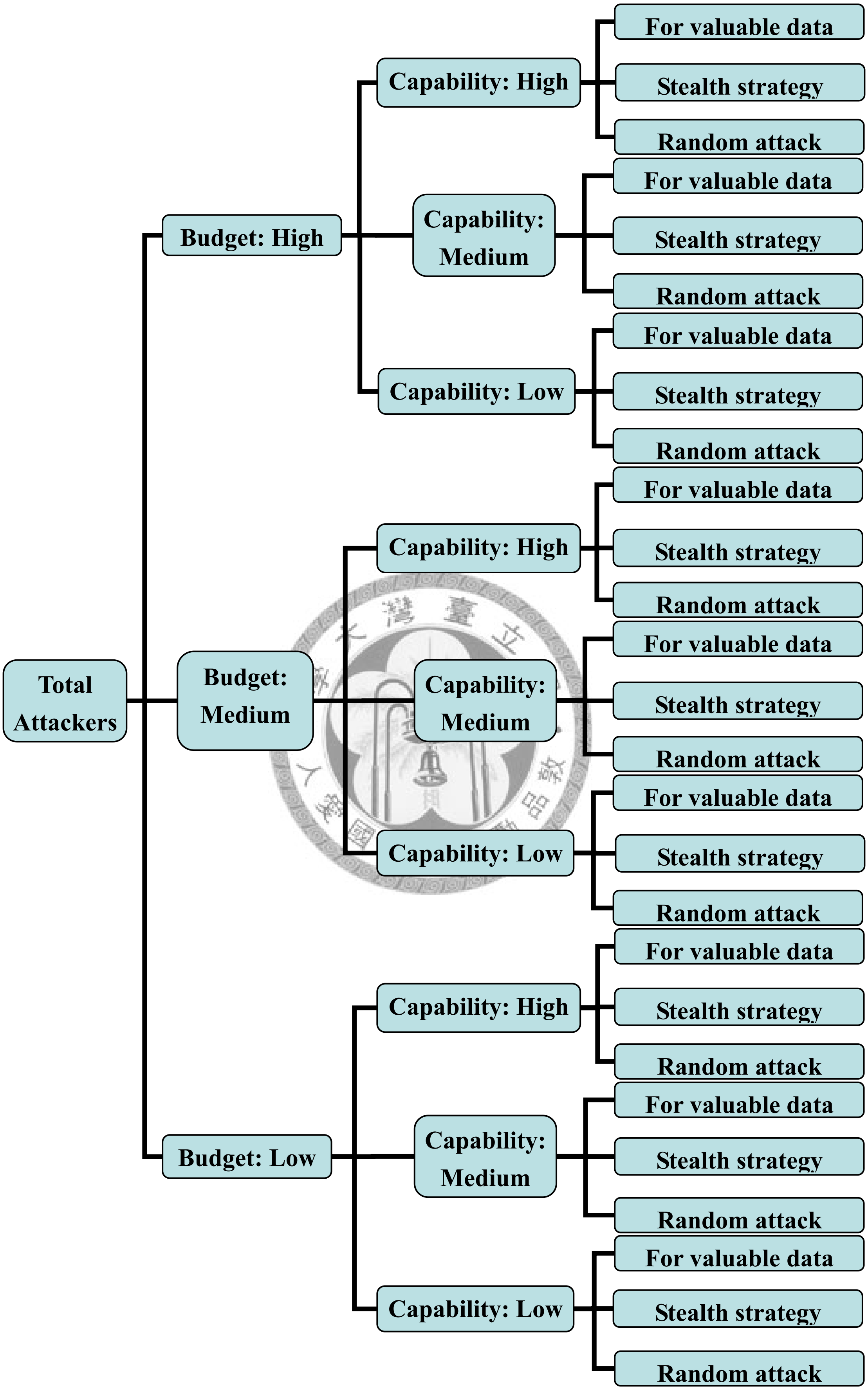


Figure 2-1: The Attacker Classification

2.3.2 A Possible Scenario

The following is a possible scenario. Assuming there is one node with high sensitive information in the network. For improving system security, defenders deploy various defensive resources on different nodes according to their requirements. Here, we consider multiple types of resource, including honeypot with “fake information” to lure attackers to spend attack resource on it, and another kind of honeypot to learn attackers’ behavior and waste their resource.

For attackers, the main goal is to compromise the core node. We assume attackers are not aware the existence of honeypots in the network. In other words, they only have imperfect knowledge. Therefore, while attacking, they may believe they have successfully achieved their objective, but in fact, they just simply attacked a false target. Another possible situation is they are distracted by the honeypot which set up for learning attack strategies and wasting attack resource.

For defenders, in order to increase attack cost and decrease the core node compromised probability, we deploy defense resource including honeypots on each possible attack path. However, there is always an irresistible constraint, the budget constraint. Defenders should allocate defense resource on each node under this restriction. All above decisions are important variables that intensely influence the whole system defense capability. The detailed assumptions are described in table 2-2.

Table 2-3: Problem Assumptions

Assumptions:

1. There is only one single core node in the network.
2. The defender has the perfect knowledge of network that is attacked by several attackers with different budget, capabilities, and next hop selecting criteria.
3. The attackers are not aware that there are honeypots deployed by the defender in the network, i.e., the attackers only have imperfect knowledge of network.
4. There are two types of defense resources, the honeypot and non-honeypot. Further, honeypots can be divided into two categories, one is used for wasting attackers' resources and learning their tactics, and the other is used to play the role of fake core node to distract the attackers.
5. A node is only subject to attack if a path exists from the attacker's position to that node, and all the intermediate nodes on the path have been compromised.
6. A node is compromised when attack resources allocated to it is no less than the defense force incurred by defense resources.
7. Only malicious attack is considered.
8. Only nodal attack is considered.
9. The network is viewed at the AS level.

As mentioned before, honeypots are divided into two categories. One contains fake information which seems to be real sensitive data [15]. After compromised this kind of honeypot, attackers may believe they have got what they want and terminate this attack. Another is deployed for wasting attackers' resource and learning their attacking strategies [20]. Although this type of honeypot may become a spring board for attackers to perform further malicious activity, they can never get any critical information from it. Besides, defenders only lost a "shell system", but get lots of valuable records about attacking behavior. Moreover, we also weaken the attack power of the malicious user.

To describe the attack procedures specifically, we adopt the following concept. First, the attacker occupies an initial node, s (Figure 2-2). Due to different budget allocation, each node has distinct defense capability (Figure 2-3). Next, he chooses a target from the candidate set and compromises it if he can apply enough attack power to it. The compromised node is used as a hop-site and its uncompromised neighbors are added to the set of victim candidates for the next stage of the attack (Figures 2-4 and 2-5). When attackers compromise a false target honeypot, there are probabilities attackers will believe they achieve their goal and halt this attacking act. (Figure 2-6 and 2-6-1-1). If attackers penetrate false target honeypot, they will repeat the selecting and compromising process until exhaust their resource or have already reach their target (Figure 2-6-2-1). Finally, if attackers penetrate false target honeypot and compromise the core node successfully, the attack path is illustrated in Figure 2-6-2-2. Diagrams of the attack behavior are presented below.

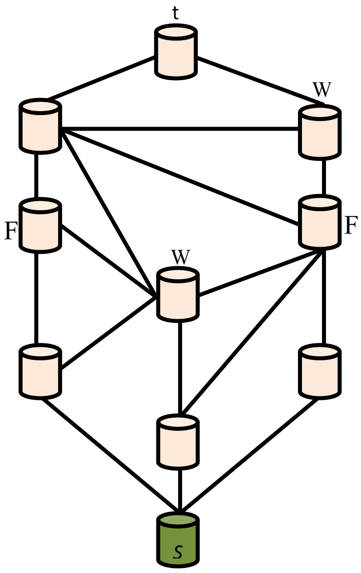


Figure 2-2: Initial State.

Initially, the attacker is on node s.

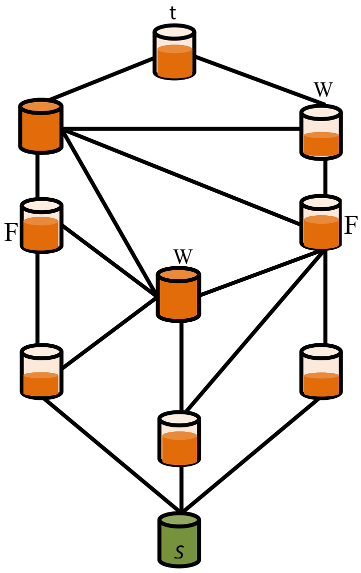


Figure 2-3: Defense Resource Allocation

Due to different budget allocation, each node has distinct defense capability.

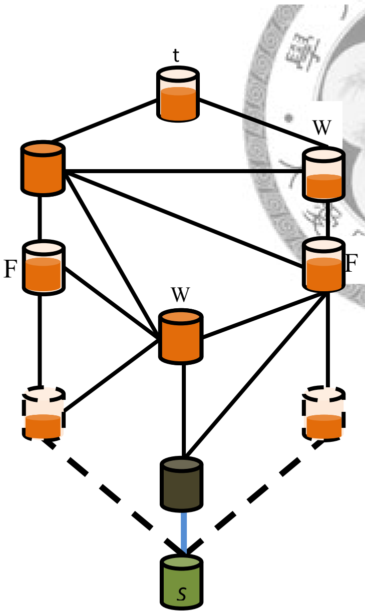


Figure 2-4: Probing Nodes

Attackers choose next hop from node s according to their selecting criteria.

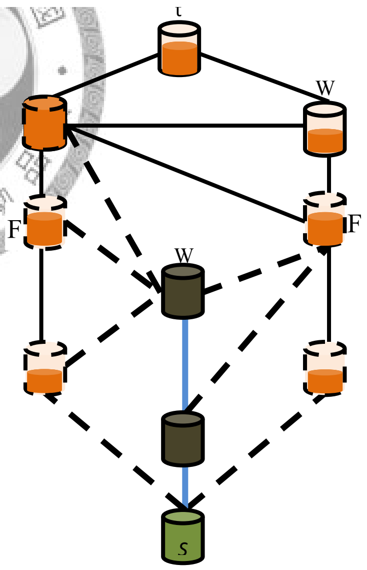


Figure 2-5: Node Selecting

Attackers continue selecting next hop from node s or from node which can be accessed in the previous attacking step.

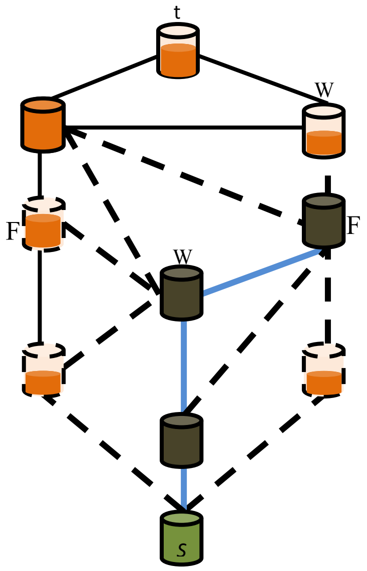


Figure 2-6: False Target Honeypots

When attackers compromise a false target honeypot, there are probabilities attackers will believe they achieve their goal and halt this attacking act.

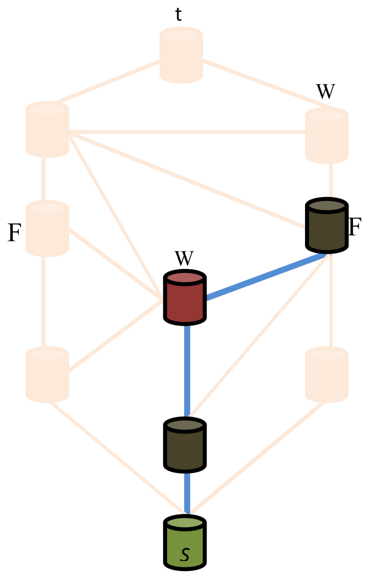


Figure 2-6-1-a: Successfully Cheat Attacker

If attackers are cheated by false target honeypot, this is their attack path, which ignores links and nodes are not chosen during attack process.

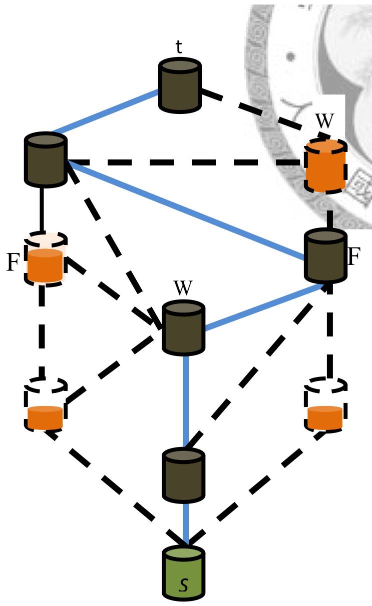


Figure 2-6-2-a: Attacker Penetrate False Target

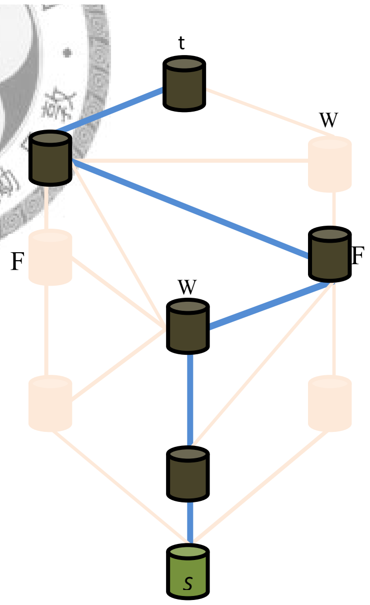


Figure 2-6-2-b: Total Attack Path of Penetrating False Target Honeypot.

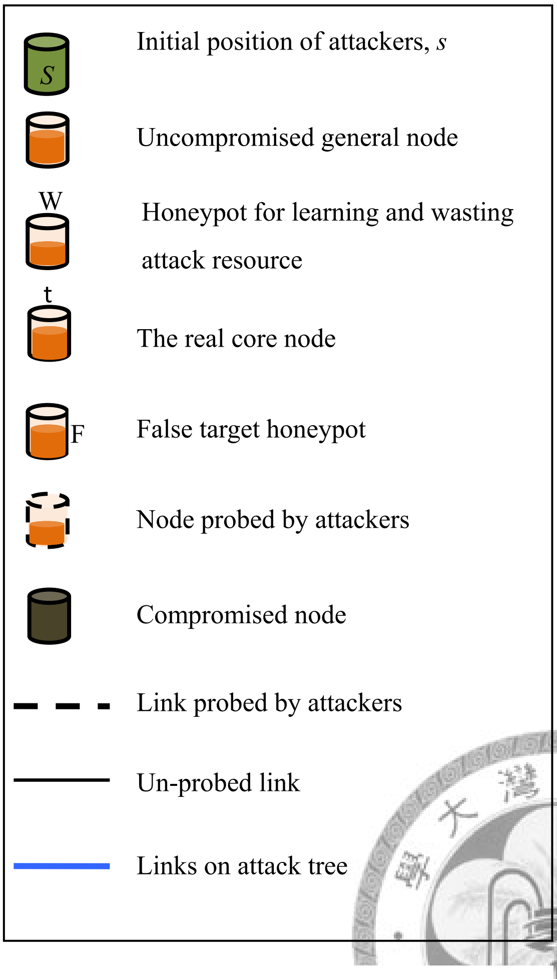


Figure 2-7: Explaination of components

For defend strategies, there are various possible situations. Our model just represents a generic concept, $\vec{D}$. It may contain different factors in each application. The following is an example that we let $\vec{D}$ includes honeypots (both wasting attack resource and distraction) and other defense resource that raise the attack cost. Besides, there is budget constraint both for attackers and defenders. The corresponding settings are in table 2-4.

Table 2-4: Given Parameters and Decision Variable of a Possible Scenario

Control parameter	
Notation	Description
M	The total evaluation frequency for all attacker categories
Given parameters	
Notation	Description
K	The total attacker categories
R_k	Rounded evaluation frequency of each attacker type (where $k \in K$)
P_k	The portion of attacker type k in total attackers (where $k \in K$)
D	All possible defense strategies
$\overrightarrow{A_k}$	The strategy of an attacker, comprising his budget, capabilities, and next hop selecting criteria (where $k \in K$).
$S_{kj}(\overrightarrow{D}, \overrightarrow{A_k})$	1 if the attacker j of the k^{th} attacker category can compromise the core node under $\overrightarrow{D}$ defense strategy, and 0 otherwise (where $k \in K$)
B	The total budget of defender
B_k	The total budget of the k^{th} type of attacker, where $k \in K$
N	The index set of honeypots for wasting attackers' resources and learning their tactics
F	The index set of honeypots to play the role of fake core nodes
I	The index set of all general nodes in the network
Decision variables	
Notation	Description
b_i	The defense resource allocated to protect a node i, where $i \in I$
h_n	The defense resource allocated to honeypot n in the network, where $n \in N$
h_f	The defense resource allocated to honeypot f as the fake core node in the network, where $f \in F$
$a(b_i)$	The cost of compromising a general node i in the network, where $i \in I$
$a(h_n)$	The cost of compromising a honeypot n in the network, where $n \in N$
$a(h_f)$	The cost of compromising a honeypot f in the network, where $f \in F$

As mentioned before, the detailed description and corresponding determining method of the control parameter, M, will be discussed in section 3. 1.

The corresponding mathematical formulation for this possible scenario is modeled in the following. The thing deserves to be mentioned is that the objective function seems the same as we proposed in section 2.2. This is because the previous model is a generic one and the scenario is a specific subset belongs to the generic model. Therefore, the expression looks the same. However, there are still different points. For instance, the constraint is distinct than the previous one. Because this is a specific scenario, we can model it detailed in mathematical. The meaning of Constraint (IP 1.1) and Constraint (IP 1.2) are the same with those in section 2.2. Constraints (IP 1.3) to Constraint (1.6) jointly represent the defender's budget constraint. Constraints (IP 1.7) to Constraint (1.10) indicate the attacker's budget limitation.

Objective Function:

$$\min_{\overrightarrow{D}} \frac{\sum_{k=1}^{K}\sum_{j=1}^{R_k} S_{kj}(\overrightarrow{D},\overrightarrow{A}_k)}{M} \qquad \text{(IP 1)}$$

Constraint:

$$\overrightarrow{D} \in D \qquad \text{(IP 1.1)}$$

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

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

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

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

$$0 \le h_n \le B \qquad \forall n \in N \qquad \text{(IP 1.6)}$$

$$\sum_{i\in I} a_i(b_i) + \sum_{n\in N} a_i(h_n) + \sum_{f\in F} a_i(h_f) \le B_k \qquad \forall k \in K \qquad \text{(IP 1.7)}$$

$$0 \le \sum_{i\in I} a_i(b_i) \le B_k \qquad \forall k \in K \qquad \text{(IP 1.8)}$$

$$0 \le \sum_{n\in N} a_i(h_n) \le B_k \qquad \forall k \in K \qquad \text{(IP 1.9)}$$

$$0 \le \sum_{f\in F} a_i(h_f) \le B_k . \qquad \forall k \in K \qquad \text{(IP 1.10)}$$

Chapter 3 Solution Approach

3.1 Evaluation Process

To measure the effectiveness of a defense resource allocation, we adopt evaluation as an approach. Since our scenario and environment are very dynamic, it is hard to solve the problem purely by mathematical programming. Through evaluation, we can well describe the behavior of distinct attackers. For instance, every attacker category discussed in 2.3.1 is integrated into the evaluation. While canvassing one defense strategy, the defense resource allocation configuration is attacked by all categories of attackers. The attacker number of each category is determined by the product of M (Total attacker/evaluation number) and P_k (the portion of attacker type k in total attackers).

For each attacker category, although attackers belong to the same type, there is still some randomness between each other. This is caused by honeypots. Recall that in attacker classification, if an attacker compromises a false target honeypot, there is a probability that he will believe the core node is compromised and terminate this attack. Therefore, we can never guarantee the result of an attack is successful or failed until the end of the evaluation.

The total evaluation frequency is set to be M and this value is determined by experiment. First, we make an initial value, for example, 10 million. Then, we let 10 thousands as a chunk to summary the result and draw a diagram depicting the relationship between compromised frequency and number of chunks. If the diagram shows a converging trend, it implies the value of M is an ideal one since there is no obvious difference of compromised frequency between chunks. On the other hand, if the result shows a vibrating result, we think M is set too small and we will let it be a larger number to run this testing experiment.

After deciding the value of M, we can start our process of discovering the optimal solution. At the beginning, we have an initial resource allocation configuration. Based on this, we run evaluation with the whole 27 distinct categories of attackers for M times and get the core node compromise frequency. Then, we use this frequency divided by M to gather average core node compromised probability. This result becomes a benchmark to evaluate the performance of the following consequence.

The next step is to enhance the quality of solution. We select nodes need to be adjusted by heuristic and change the defense resource allocated on them. After that, we run another M times evaluation using adjusted defense parameters and get the core node compromised frequency. Again, let the frequency divided by M to gather average core node compromised probability. Finally, we check whether total number of policy enhancement has reached N times or not. If it has, then we terminate the enhancing process and compare the final result with initial state. Otherwise, we continue next round until the terminate condition happened.

As mentioned before, we know that N plays an important role in our process of discovering optimal solution. There are various methods to determine this value, but we choose it from the basic of whole problem. The problem we want to solve is a defense resource allocation strategy for defenders. Therefore, the outcome of the problem should generate as soon as possible. Unlike other planning problems, for example, network planning, once the conclusion is made, administrator can apply it for a couple of years or even more. However, in security domain, the attacker tactics evolve almost every day, defenders should adjust their strategy corresponding. Further, defense technology is also progressing. Security officer may need to adopt novel defense technique. Therefore, the lifetime of one defense resource allocation is not as long as those general planning problems. Defenders may need the solution within one day and only use this result for one year.

Therefore, the value of N is decided by a resource constrained approach. We first set the whole period defenders willing to give this process, for instance, 8 hours. Then, let the whole period divided by the time required for one round policy enhancement and this quotient is N.

Besides N, if we cannot find any other defense resource allocation scheme which achieves better performance than current one, we also stop policy enhancement. Therefore, at each round we check both conditions to make sure whether enhancement should be terminated or not.

The above is the explanation of whole process, we summary these into a flow chart and present it in Figure 3-1.

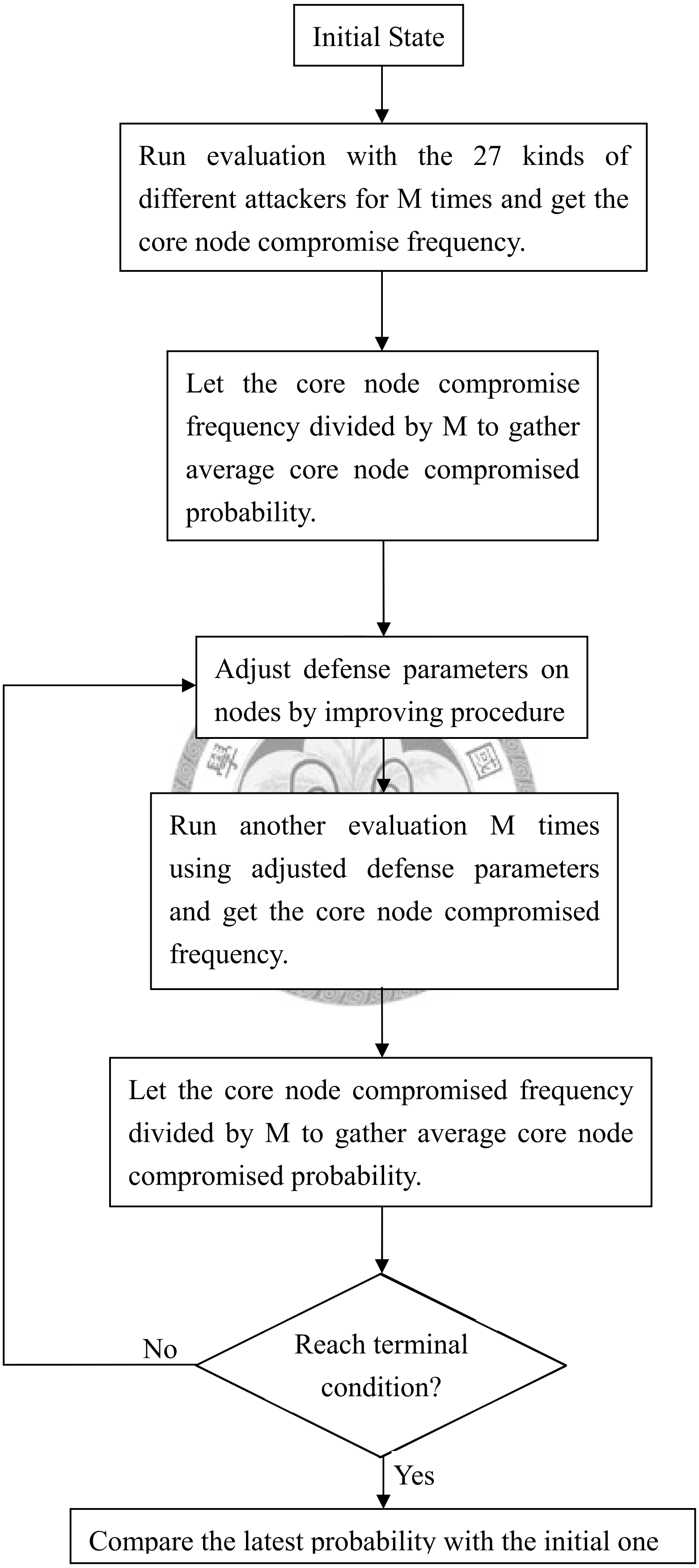

Initial State
Run evaluation with the 27 kinds of different attackers for M times and get the core node compromise frequency.
Let the core node compromise frequency divided by M to gather average core node compromised probability.
Adjust defense parameters on nodes by improving procedure
Run another evaluation M times using adjusted defense parameters and get the core node compromised frequency.
Let the core node compromised frequency divided by M to gather average core node compromised probability.
Reach terminal condition?
No
Yes
Compare the latest probability with the initial one

Figure 3-1: Process for Getting Optimal Solution

The enhancing method of the solution approach is proposed in the next subsection.

3.2 Policy Enhancement

The methodology we used to enhance the resource allocation performance can be summarized into two main concepts: derivative and popularity based strategy. Detailed descriptions are in the following statement.

- **Derivative**

  This concept is using to measure the marginal effectiveness of defense resource allocation. Like the word "marginal" means, derivative is the difference between the present allocation and the allocation with more unit resource deployed. Defenders can make decision by evaluating the derivative value. The higher the derivative of one node, the worthier defender allocates defense resource on it. This approach is not only applied here but also used in a lot of different domains. For example, when finding the minimum (or maximum) value of a simple linear equation, we may calculate its derivative and discover the extreme value of this equation within a short time.

  However, the problem we faced is not a simple linear equation. If we still want to deploy this mechanism, some adaptations need to be made. Therefore, the method used to calculate derivative is not algebra. We derive it through experiments. While finding the derivative of one allocation, we reallocate defense resource, then evaluate the performance (in our problem, the performance is represented by core node compromised probability). The difference of the probability between the previous state and the enhanced state divided by total amount of resource unit forms our derivative.

The main concept of enhancement is to reallocate resource in the network. We first try to take certain amount of resource from every node in the network. If all nodes afford the resource removing, we can get (Number of node in the network × Amount of resource we took from each node) reallocating resource.

And then choose node(s) to reallocate resource by popularity based strategy. After finishing this procedure, we let the posterior core node compromised probability minus the prior one and divided by the amount of reallocating resource to get the derivative. Notice that, since the numerator of derivative is the core node compromised probability, the ideal difference of posterior and prior state should be a negative value. Therefore, we choose the scheme which has the lowest derivative to replace current one.

- **Popularity Based Strategy**

This strategy is focuses on those nodes are frequently attacked. The basic view point is that if we can discover what kind of nodes are "popular" for attacker, the corresponding defense strategy will easy to be determined. Therefore, the first step is to reveal a proper metric which can reflect the node's popularity for attacker.

There are many different criteria to establish this metric. However, once defender chooses a metric which can not exactly express the habits of attackers, the whole enhancing process will lead to a wrong resource allocation configuration. Therefore, we decide to let the total cost attackers spent on one node divided by accumulated attack cost spent on every node in the network as the metric in policy enhancement since larger this ratio is, more confidence we have to conclude this is a main target of attackers. By deploying this concept, we avoid the situation of misunderstanding attacker's preference.

After understanding the concept of derivative and popularity based strategy, we combine both principles to form the policy enhancement. The following is our integrating method.

First, at each evaluation, we remove defense resource from nodes in the network.

However, the number of nodes we take resource is not exactly the same in each enhancement. It is because defense resource is not evenly distributed through entire network, there may be some nodes only with little defense resource initially. These nodes are not proper candidates to remove resource. Therefore, in each enhancement, the number of nodes which we take resource is not our decision metric. Instead, we use total quantity of resource to reallocate as the pointer. If the quantity of resource we can use in reallocation is larger than a predefined threshold, then we continue the follow-up procedure, otherwise, we back to start and change the value for another trial.

Besides, in each enhancement, we examine a wide variety of quantities of resource we take from each node by experiments and figure out the optimal one for following procedure. In other words, we discover the best amount of resource to reallocate at current status. Further, the quantity of resource we take from nodes is adjusted by concept of harmonic series. For the first iteration, we test two different quantities. One is initial value plus gain, which is the initial value divided by one plus the number of iteration. The other is initial value minus the gain. Using these two values, we run individual experiments to calculate the core node compromised probability. The value contribute lower compromised probability will become the initial value of next iteration. In other words, we not only examine different value but also determine which direction we should adjust to find the optimal quantity. The possible decisions of this tuning method are shown in figure 3-2.

We can discover that this method is gradually narrow down the possible area. By harmonic series, gain value is not remaining the same at different stages. In figure 3-2, there is an initial value at the beginning. For each decision, there are two options, which are current value plus gain and current value minus gain. For example, if current value is V_3, then the next step is to evaluate the core node compromised probability using V_7 and V_8. If V_8 perform better than V_7, we will take V_8 as the quantity we take resource from nodes in the network. Further, gain value becomes smaller when search round time grows. This is because denominator of gain becomes bigger and this feature helps us to find optimal value since more times we searched, much closer we are toward the optimal one. Therefore, descending gain can prevent over shooting situation.

The reason why we apply different value on gathering resource is to avoid the situation of “over shooting”. While adjusting, there may not be directly proportional to performance of defense resource allocation and quantity of defense resource allocated on it. This is because though we raise the defense level of one node to an extremely high level, at the same time we sacrifice other nodes’ defense force. It probability make whole network more vulnerable than before and this situation is called “over shooting”.

After gathering proper resource to reallocate, the next step is to find out how to distribute the resource. By calculating the quotient of attack cost spent on each node divided by total attack cost spent in the entire network, we can generate a list, following the order from high to low, and allot resource. However, the way we reallocate resource is also an extremely important process in our enhancement. This can direct influence the whole network security. This is because in some status, concentrating resource makes greater performance while in some situations, distributed allocation contributes better consequence. Therefore, the number of nodes needs to add defense resource is determined through experiments. We test different methods of reallocation, for example, add all resource upon one single node or separate resource to many nodes in the network. However, the relationship of resource and defense level forms a concave function, it is difficult to judge whether concentrating or distributing strategy is better without experiment. Therefore, we calculate derivative for each reallocation scheme. Since the metric is core node compromised probability, the smaller the derivative is the greater performance this scheme provides. Then, we take the scheme with lowest derivative to replace current resource allocation scheme. The operation of this process is illustrated in Figure 3-3 and the relationship between the policy enhancement and the whole process is shown in Figure 3-4.

It may be confused that in each enhancement, if we cannot get enough defense resource, we go back to change a smaller test value and execute again. This is because the amount of nodes which can be removed larger test value may be fewer than those can be removed smaller one.

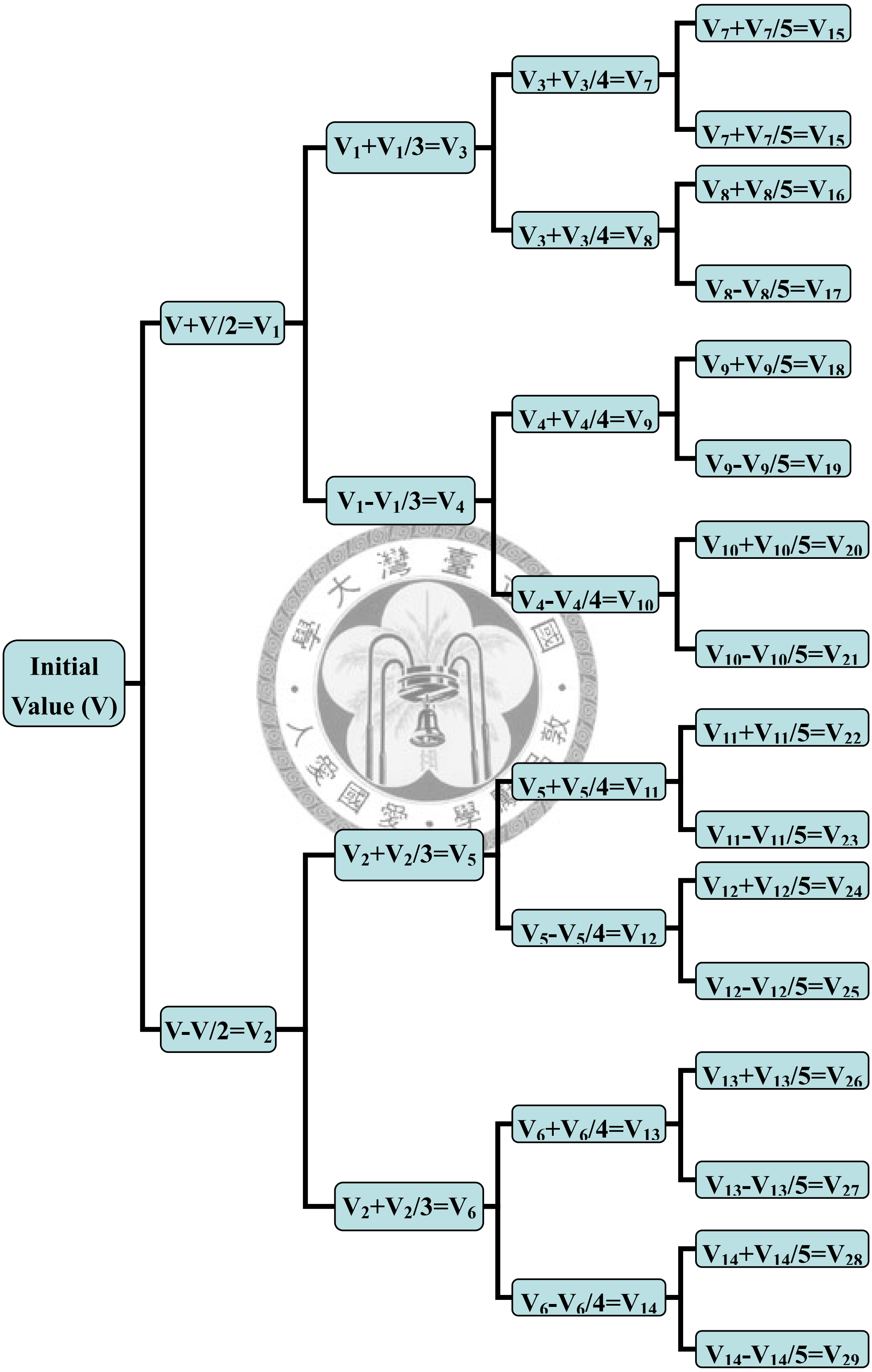
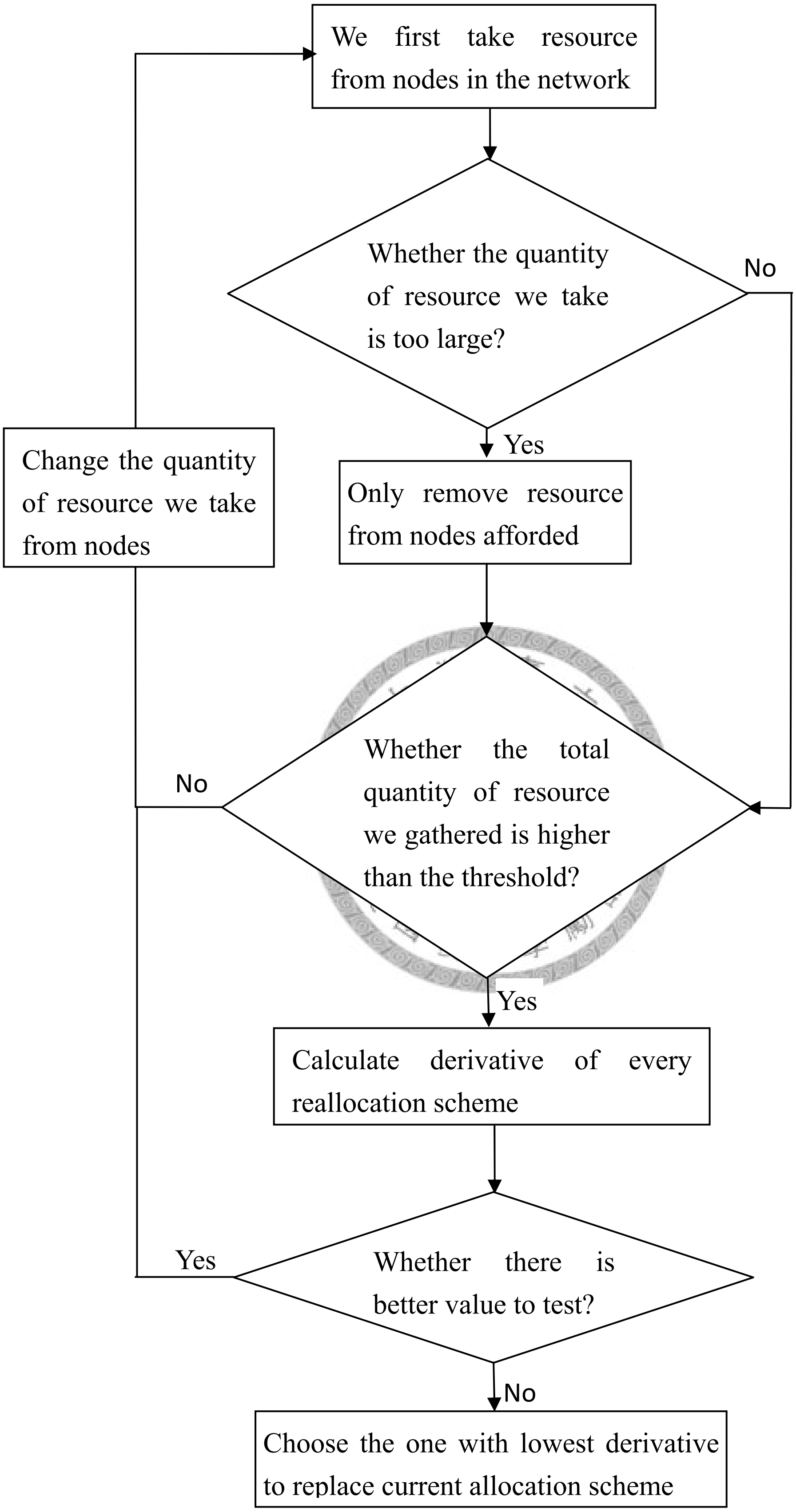


Figure 3-2: The Possible Decisions of Tuning Method



We first take resource from nodes in the network
Whether the quantity of resource we take is too large?
No
Yes
Only remove resource from nodes afforded
Change the quantity of resource we take from nodes
Whether the total quantity of resource we gathered is higher than the threshold?
No
Yes
Calculate derivative of every reallocation scheme
Whether there is better value to test?
Yes
No
Choose the one with lowest derivative to replace current allocation scheme

Figure 3-3: The Operation of Policy Enhancement

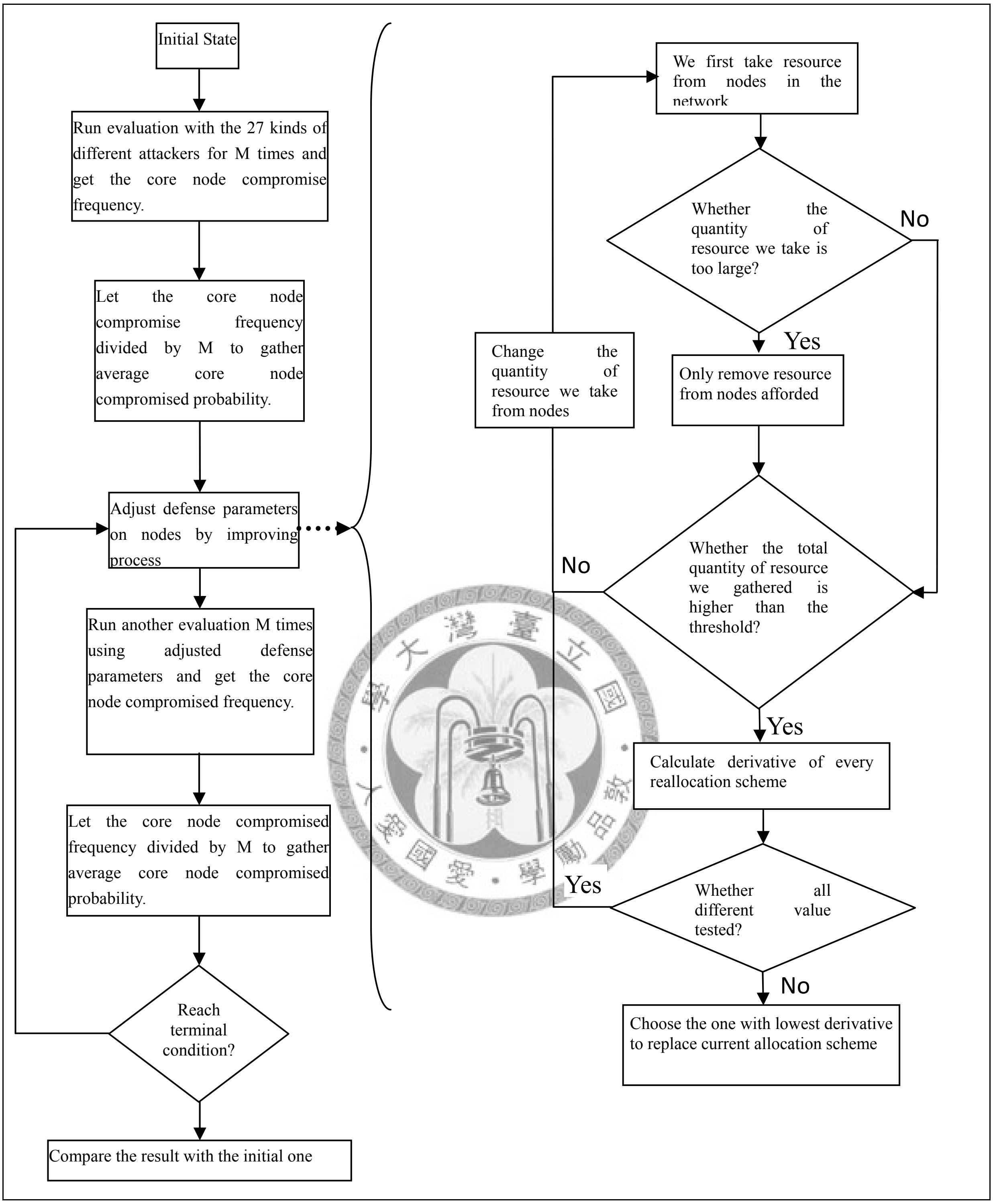

Initial State
Run evaluation with the 27 kinds of different attackers for M times and get the core node compromise frequency.
Let the core node compromise frequency divided by M to gather average core node compromised probability.
Adjust defense parameters on nodes by improving process
Run another evaluation M times using adjusted defense parameters and get the core node compromised frequency.
Let the core node compromised frequency divided by M to gather average core node compromised probability.
Reach terminal condition?
Compare the result with the initial one
We first take resource from nodes in the network
Whether the quantity of resource we take is too large?
No
Yes
Only remove resource from nodes afforded
Change the quantity of resource we take from nodes
Whether the total quantity of resource we gathered is higher than the threshold?
No
Yes
Calculate derivative of every reallocation scheme
Whether all different value tested?
Yes
No
Choose the one with lowest derivative to replace current allocation scheme

Figure 3-4: Relation between Policy Enhancement and Whole Process

3.3 Initial Allocation Scheme

In this section, we will introduce our initial defense resource allocation algorithm. This initial scheme is also the start point of our evaluation. Since the quality of initial scheme will influence the process of evaluation, we must choose an effective method to construct it. Therefore, rather than considering only one metric, in our algorithm, we take two important pointers of topology, which are number of hops from the core node and link degree of each node. The detailed description is as follows:

- **Number of hops from the core node**

  While evaluating the importance of nodes in the network, this metric is an ideal one since more closer to the core node, more strategic value it has. Once attackers have compromised the node which is only one hop from the core node, the opportunity of attackers to successfully compromise the core node is higher than attackers are far away from the core node. Therefore, we apply this concept on getting our initial allocation scheme.

  The principle of allocating resource by number of hops from the core node is that defender will put more resource on nodes that close to the core node while those are distant from the core node, less resource will be allocated to them.

- **Link degree of each node**

  For this guidance, we treat a node as a critical one when it has higher link degree than other nodes in the network. This concept is from the attackers' view. This is probability that attackers' judge a node by its number of links since higher connectivity of one node is, more important role it plays in the network. Therefore, attackers may prefer to compromise those nodes with high link degree no matter their purpose is to steal critical information or destroy the entire system. In other words, this point of view can be adopted on diversiform of attackers.

Besides, the way we evaluate link degree is the number of links on each node divide by total number of links within the network. The higher this value is, the more critical this node is. It is obvious that we apply a fractional method to determine the weight which makes this perspective external.

After explained the ideals of number of hops from the core node and link degree, the following is the combination method of these two concepts. We combine these ideals by proportional method. Moreover, for discovering which strategy is more important for defenders, we apply eleven different ratio configurations to get various initial allocation schemes. That is to say, the weight of number of hops and link degree are diverse in each scheme. The first allocation will take 0% link degree and 100% hop numbers as the allocating guidance. The next allocation will consider 10%

link degree and 90% hop numbers and so on so forth. Similarly, the final scheme takes 100% link degree as resource allocating guideline. Then, let these results become the input of our evaluation process.

Additionally, through these experiments, we can also find out which proportional is the best one of these combinations. The corresponding algorithm is described in table 3-1.

Table 3-1: Initial Resource Allocation Algorithm

Step 1: Calculate the number of hops to the core node for each node in the network. This will be the raw data to form our metric.

Step 2: Sort the number of hops by descending order and calculate the ratio of hop number divided by summation of hop numbers for each node.

Step 3: Because we believe nodes which closer to the core node, it play more important role. Therefore, we need to reverse the order we get from step 2 since the data computed by above step has higher ratio when nodes are far away from the core node.

Step 4: Compute number of links on each node and store this information for further use. This is another metric to evaluate the importance of one node.

Step 5: Let data collected from step 4 divide by the summation of link numbers in the network individually.

Step 6: Apply different weight, for example, 40% by number of hops and 60% by link degree, to the two metric and allocate defense resource.

Chapter 4 Computational Experiments

In this section, we illustrate the detailed description about computational experiments, which includes the environment, and experiment result and scenario analysis.

4.1 Experiment Environment

All algorithms we proposed are written in C and executed on a laptop. The CPU is Intel Core 2 Duo T7300 2.00GHz. Main memory size is 2GB. For defender, corresponding parameters are described in table 4-1.Parameters regarding to attackers are listed in table 4-2. Other system experiment parameters are shown in table 4-3.

Table4-1: Parameters for Defender

Parameter	Value
Total budget	1,000
Number of nodes	10
Number of honeypots for wasting attackers' budget	2
Number of honeypots as false target	2
Number of links in the network	17

Table4-2: Parameters for Attacker

Parameter	Value
Total number of attacker profiles	27
Budget levels	3
Capability levels	3
Types of next hop selecting criteria	3

The total number of attackers' profile is composed by budget level, capability level and next hop selecting criteria. Therefore, the number 27 is formed by 3×3×3. All three budget levels take minimum attack cost as the benchmark. For low level attackers, we set their budget is minimum attack cost. Then, medium level is 1.5 times of minimum attack cost. High budget level attackers have 2 times of minimum attack cost as their budget.

Capability also plays an important role in attackers' profile. Since it influence the probability whether attackers will be distract by false target honeypot or not. Therefore, we set attackers with low level capability have 70% probability deceived by false target and medium level attackers have 50%. For high level capability attackers, there is only 30% chance defender can distract them by honeypots.

Table4-3: System Parameters

Parameter	Value
CPU	Intel Core 2 Duo T7300 2.00GHz
Main memory	2GB
Operating system	Microsoft Windows XP
Total evaluation frequency for one round	10,000,000

In system parameters, total evaluation frequency refers to the parameter M in evaluation process. This value is determined by experiments and we will have further discussion in next subsection.

4.2 Experiment Result

The first thing we need to determine before experiment is the value of M. Therefore, we run four different experiments and then figure out the proper value for our scenario. As mentioned before, ten thousand attacks are aggregated into a chunk and become one data point in our figures. The four distinct experiments are different from number of chunks we executed. Figure 4-1 shows the result of 10 chunks. It is obvious that core node compromised frequency is very dynamic; we cannot see any trend on this metric. For result of 100 chunks in figure 4-2, the vibration phenomenon is getting alleviative. However, it is still not reach our desired level. When we increase hunk number to 1,000, in figure 4-3, there is a stable trend. However, for precise consideration, we still run experiment with 10,000 chunks. In figure 4-4, it shows stable condition after chunk number 1,000. Therefore, we set M as 1,000 chunks, which is 10,000,000.

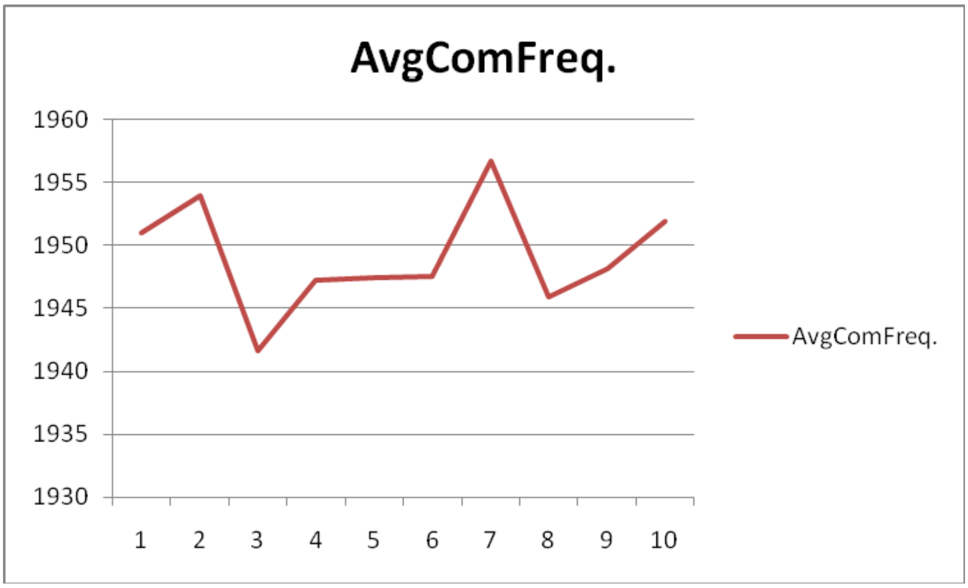


Figure 4-1: Experiment Result on M with 10 Chunks

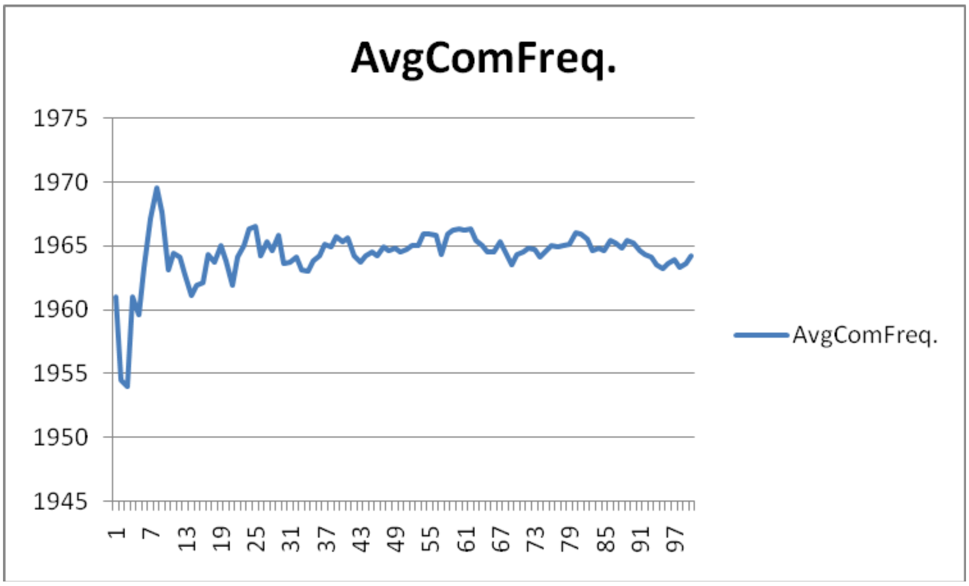


Figure 4-2: Experiment Result on M with 100 Chunks

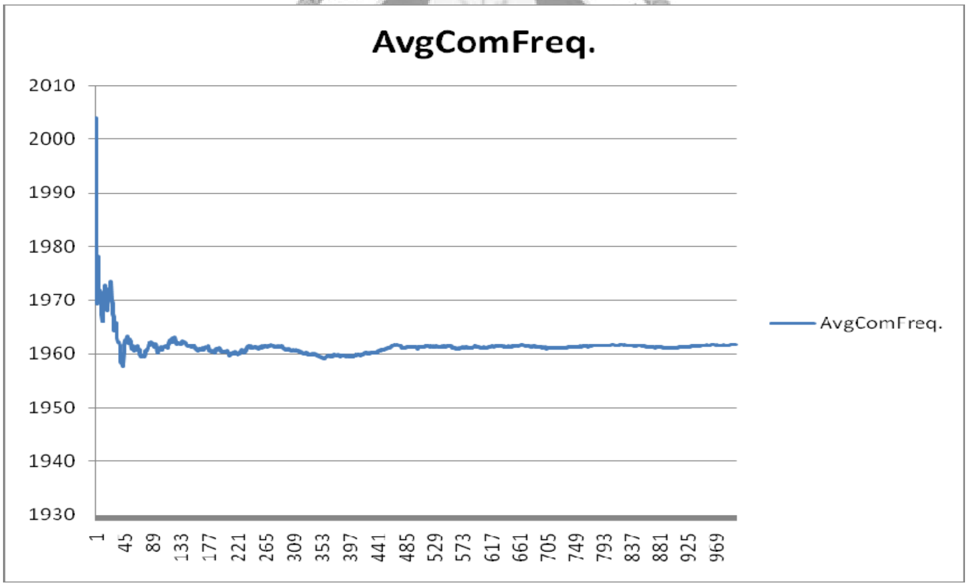


Figure 4-3: Experiment Result on M with 1000 Chunks

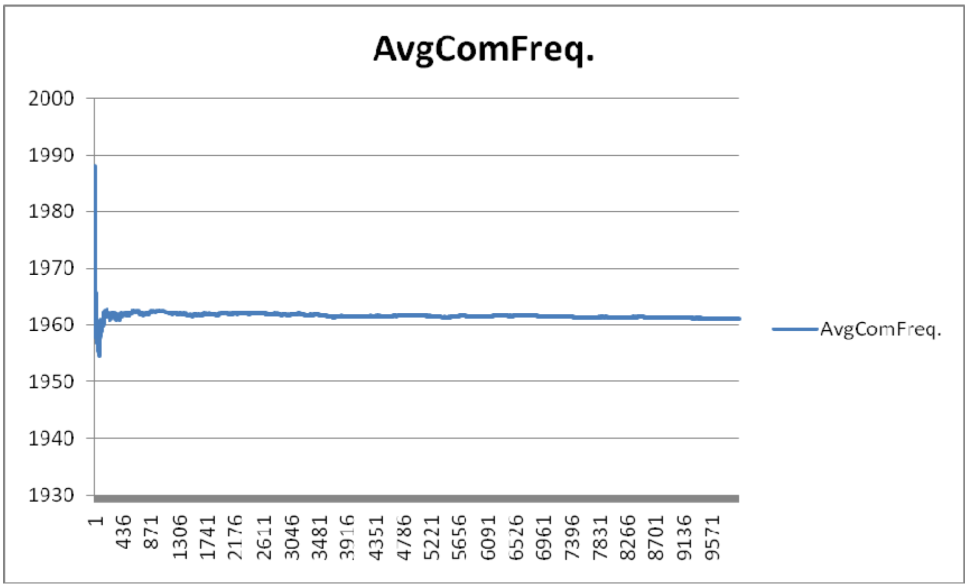


Figure 4-4: Experiment Result on M with 10000 Chunks

After deciding M, we are going to illustrate performance of initial allocation and corresponding scheme refined by policy enhancement. As mentioned before, we tried different percent combination of number of hops and link degree. However, with distinct percent combination, the minimum attack cost of each topology also changes. Recall the method we determine attackers' budget. We use multiple of minimum attack cost to generate them. Therefore, once this cost changes, attackers' budget are also influenced. The relationship of minimum attack cost and percent of applying number of hops are shown in figure 4-5.

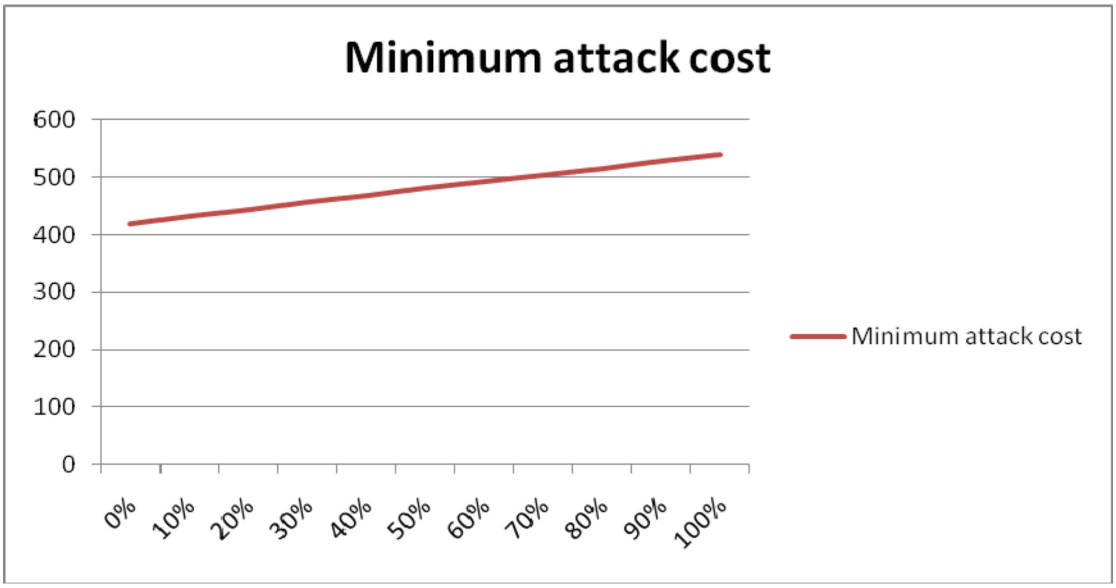


Figure 4-5: Relationship of Minimum Attack Cost and Percent of Applying Number of Hops in Initial Allocation

From figure 4-5, we can obviously discover that the minimum attack cost of each topology is increasing with the percent of applying number of hops in initial allocation. Therefore, it is unfair if we execute these distinct percent of initial allocation directly and compare the performance between initial and enhanced one. Since we should examine the result under coincidental condition rather than attackers' budget is not the same. Moreover, in figure 4-5, we also can find out that when defender applies 70% number of hops and 30% link degree, minimum attack cost of this topology is over 500. Therefore, high budget level attackers have more than 1000 budget where defender only has 1000. In other words, once these attackers are not deceived by false target honeypots, they eventually can compromise the core node. To avoid this unfair situation, the benchmark of deciding attackers' budget in figure 4-6 is fixed at 443, which is the minimum attack cost of 20% number of hops and 80% link degree initial allocation. The vertical axis refers to the core node compromised probability and the cross axix means the percentage of defender applying hop count to form initial allocation. From this figure, we can discover applying 40% hop count and 60% link degree perform the best defense result in initial allocation. Besides, performance of those apply more portion of hop count is less effective than applying less portion of hop count. However, it also shows that if defender do not apply any strategy of hop count, it also lead to a poor performance.

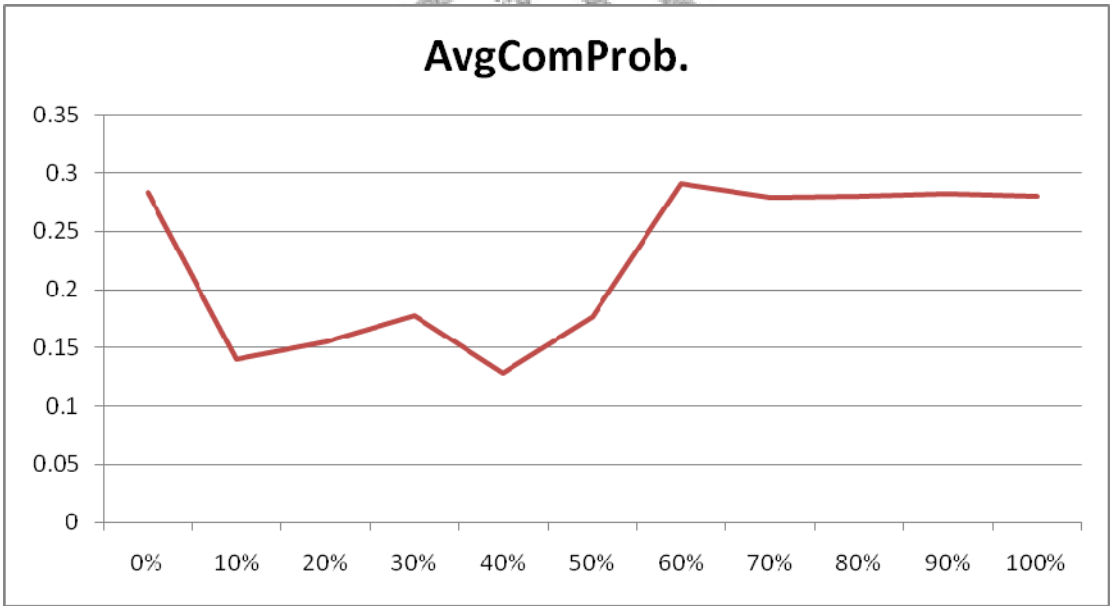


Figure 4-6: Performance of Initial Allocation when Benchmark is 466

For figure 4-7, it shows the comparison between initial allocation and enhancement allocation when defender apply 20% hop and 80% link degree. It is obvious that enhanced allocations perform much better than initial ones and the result also disclose the method we proposed is a stable one which can reach almost the same consequence even applies different initial allocation schemes. The core node compromised probability of initial allocation is between 0.1289 and 2.906 while the corresponding value of enhanced allocation is located from 0.0302 to 0.0448. The detailed data are described in table 4-4.

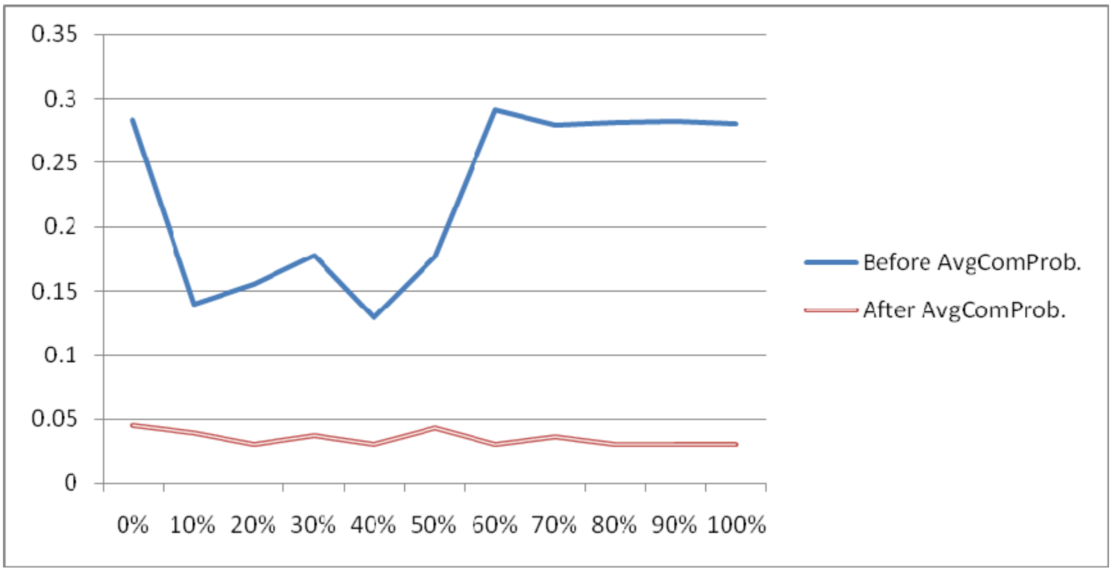


Figure 4-7: Performance Comparison when Benchmark is 443

Table4-4: Detailed Data when Benchmark is 443

% of applying hop count	Average core node compromised probability of initial allocation	Average core node compromised probability of enhanced allocation
0%	0.2832824	0.0448108
10%	0.1399964	0.0396111
20%	0.155181	0.0302591
30%	0.1775212	0.0377209
40%	0.128916	0.0302321
50%	0.1765675	0.0435941
60%	0.2906145	0.030272
70%	0.2792023	0.0362835
80%	0.280492	0.0303068
90%	0.2819864	0.0302178
100%	0.2803243	0.0303173

Figure 4-8 illustrates the enhanced rate for distinct initial allocations. The formulation of enhanced rate is $\frac{Before - After}{Before}$ where before means the core node compromised probability of initial allocation and after means the core node compromised probability of enhanced allocation. We can discover our policy enhancement can improve at least 71.71% enhancement and at most 89.58% enhancing.

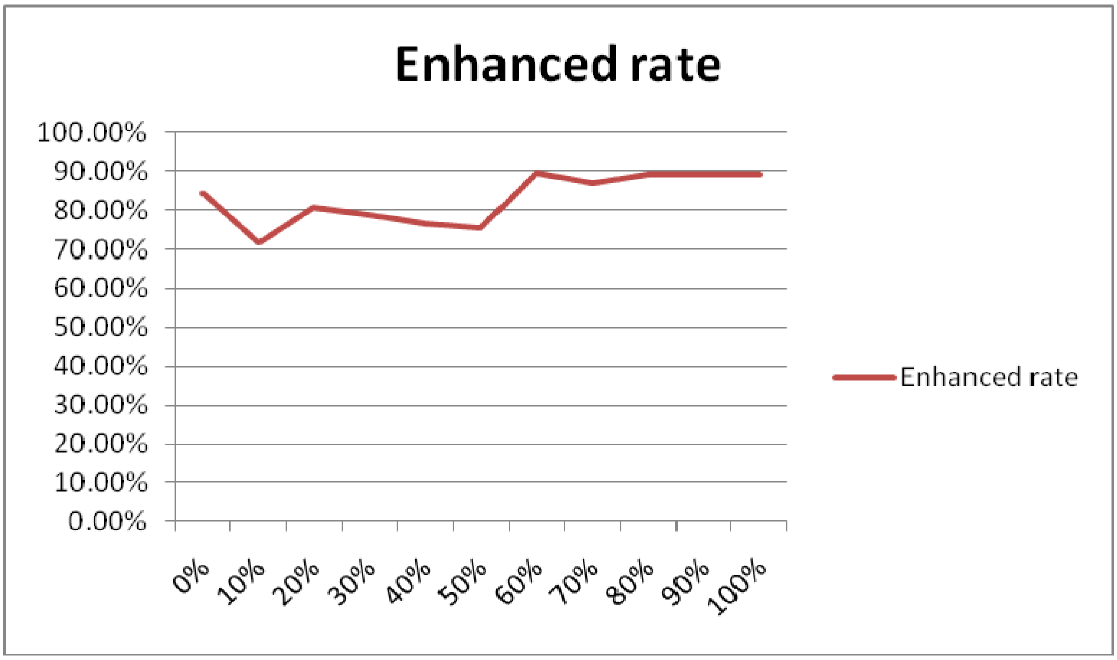


Figure 4-8: Enhanced Rate when Benchmark is 443

Table4-5: Detailed Data of Enhanced Rate when Benchmark is 443

% of applying hop count	Enhanced rate
0%	84.18%
10%	71.71%
20%	80.50%
30%	78.75%
40%	76.55%
50%	75.31%
60%	89.58%
70%	87.00%
80%	89.20%
90%	89.28%
100%	89.18%

Apropos of figure 4-9, attackers' budget is set as multiple of benchmark at 50% number of hops and 50% link degree. Similar to previous case, our policy enhancement can improve the performance of defense resource allocation scheme and reduce the core node compromised probability to 5%. The detailed data are described in table 4-6.

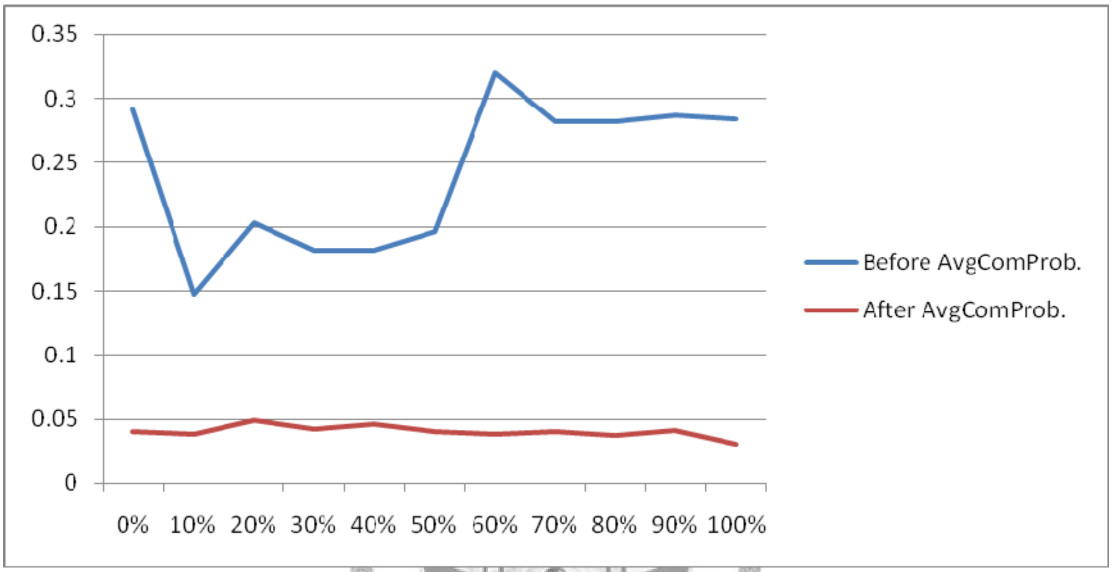


Figure 4-9: Performance Comparison when Benchmark is 480

Table4-6: Detailed Data when Benchmark is 480

% of applying hop count	Average core node compromised probability of initial allocation	Average core node compromised probability of enhanced allocation
0%	0.2915965	0.0401307
10%	0.1471459	0.0379525
20%	0.2025409	0.0493132
30%	0.1818899	0.041737
40%	0.1819564	0.0457036
50%	0.1960634	0.0405727
60%	0.3194912	0.0386533
70%	0.2818197	0.0399901
80%	0.2821127	0.037613
90%	0.2867689	0.0416979
100%	0.2839798	0.0302985

The enhanced rate of this scenario is shown in figure 4-10. We can see that the rate is still over 70% no matter on which case. The range of enhance rate is between 71.71% and 89.58%. Detailed data is listed in table 4-7.

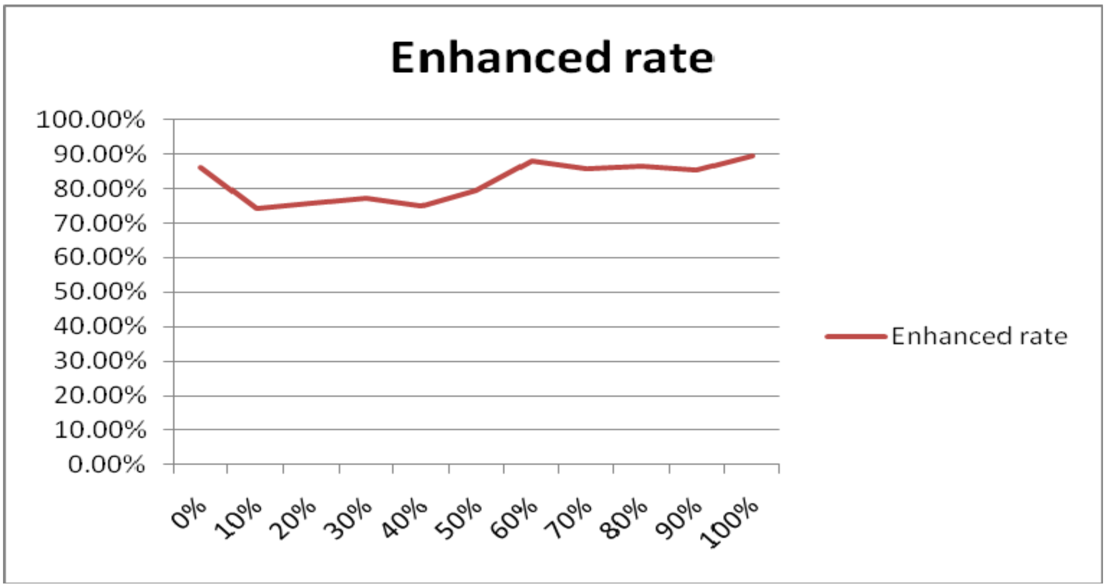


Figure 4-10: Enhanced Rate when Benchmark is 480

Table4-7: Detailed Data of Enhanced Rate when Benchmark is 480

% of applying hop count	**Enhanced rate**
0%	86.24%
10%	74.21%
20%	75.65%
30%	77.05%
40%	74.88%
50%	79.31%
60%	87.90%
70%	85.81%
80%	86.67%
90%	85.46%
100%	89.33%

In figure 4-11, minimum attack cost of 80% number of hops and 20% link degree topology becomes benchmark to form attackers' budget. In this case, attackers in high budget level have more attacking power than defender's total defense force. Therefore, enhanced allocation in this scenario does not perform as well as before. There are some vibrations in the figure since at least 33% attackers can compromise the core node if they are not distracted by false target honeypots. However, our approach still can improve the performance even though attackers' budget is more than defender's. Table 4-8 shows the corresponding data of figure 4-11.

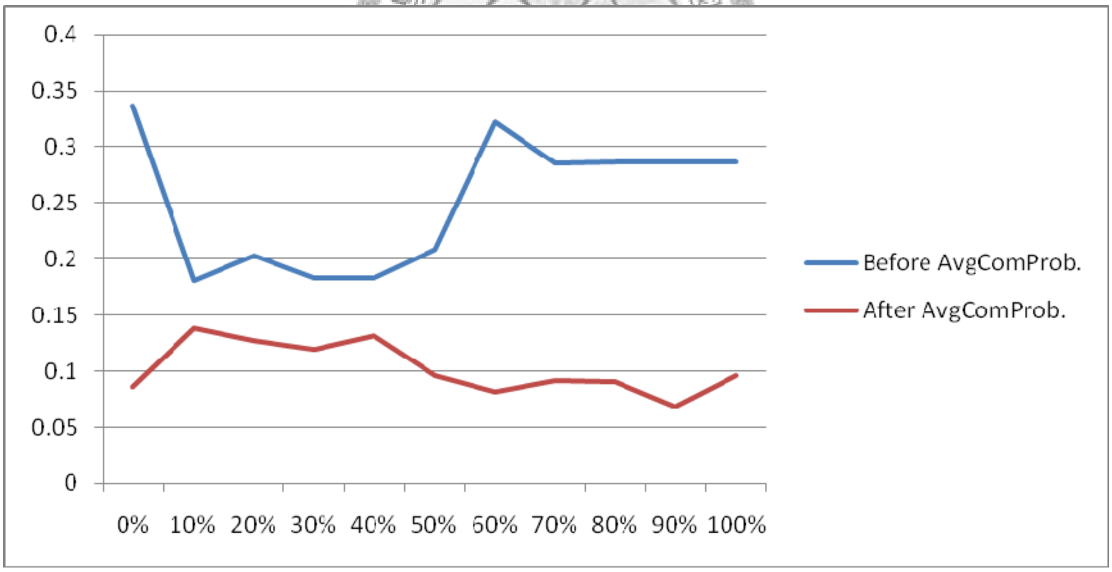


Figure 4-11: Performance Comparison when Benchmark is 515

Table4-8: Detailed Data when Benchmark is 515

% of applying hop count	Average core node compromised probability of initial allocation	Average core node compromised probability of enhanced allocation
0%	0.3363215	0.0856754
10%	0.180994	0.1380733
20%	0.2029839	0.1276159
30%	0.1828713	0.1191142
40%	0.1829314	0.1315601
50%	0.2087678	0.096209
60%	0.3223794	0.081588
70%	0.2852464	0.0918126
80%	0.2865961	0.0903938
90%	0.2869989	0.0684021
100%	0.2866791	0.0962543

For enhanced rate, we can also discover in some cases our policy enhancement can only improve near 24% but in most cases we can still reach over 66% enhancement. Figure 4-12 illustrate the enhanced rate under different initial allocation scheme and detailed data is listed in table 4-9.

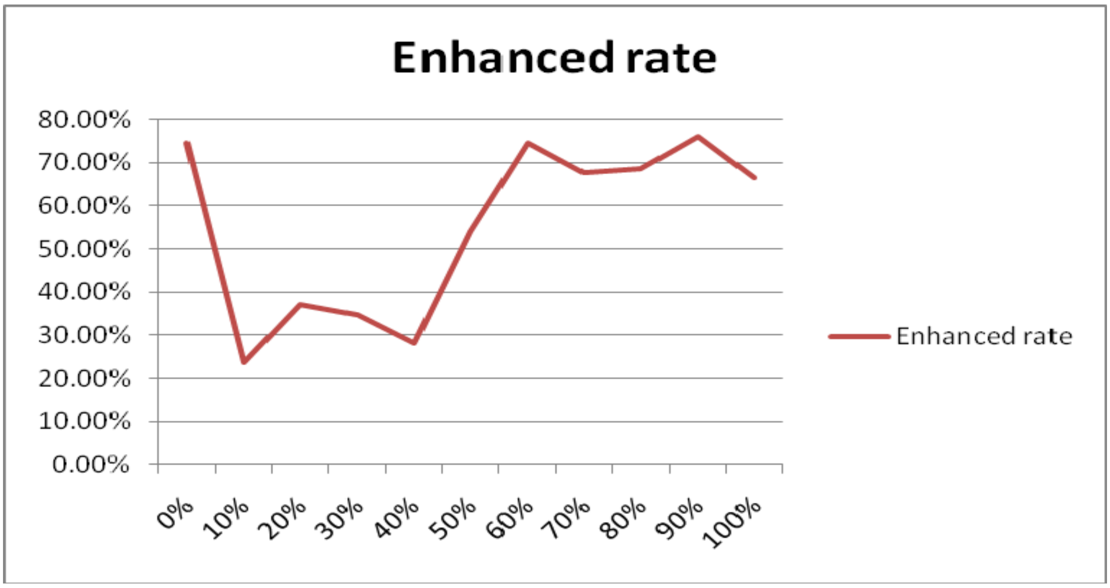


Figure 4-12: Enhanced Rate when Benchmark is 515

Table4-9: Detailed Data of Enhanced Rate when Benchmark is 515

% of applying hop count	Enhanced rate
0%	74.53%
10%	23.71%
20%	37.13%
30%	34.86%
40%	28.08%
50%	53.92%
60%	74.69%
70%	67.81%
80%	68.46%
90%	76.17%
100%	66.42%

Furthermore, we construct other experiments to verify the impact of the location of honeypots on our algorithm. The topology and corresponding node index is shown in figure 4-13. Location of honeypots on the first experiment is changed to node 6 and 9. The benchmark to decide attackers' budget is set at 515. Experiment result is shown in figure 4-15. Although the curve is similar to previous results, the resource allocation scheme is quite different. In previous scheme, defense resource is concentrated on node 6. However, if we move the honeypot from node7 to node 9, our algorithm will no longer allocate large amount of resource on a single node. Instead, in this scenario, our algorithm prefers to concentrate defense resource on multiple nodes which located on important site. For example, in figure 4-14, it illustrates enhanced scheme when defender applies 30% hop and 70% link degree as initial allocation. The number represents defense budget allocated on it. We can discover resource is concentrated on node 2, 5 and 6. This is because we remove honeypot on node 4 and translate it to node 9. The left side of entire network is more vulnerable than previous scenario. Therefore, our algorithm prefers to allocate resource on node 2 and 5 to raise system survivability.

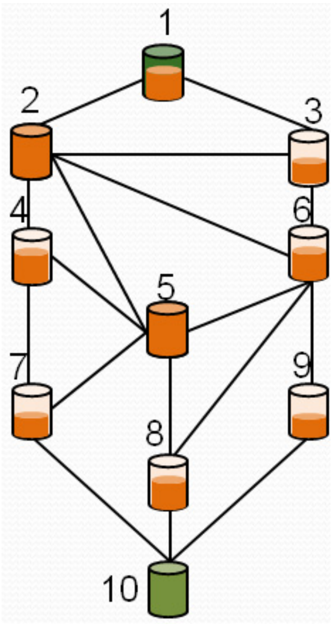


Figure 4-13: Topology and node index

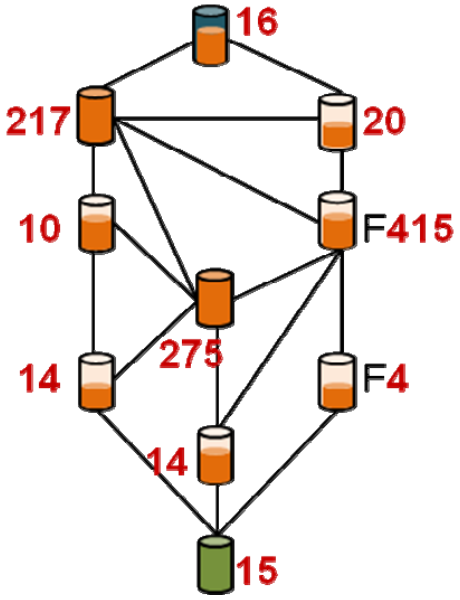


Figure 4-14: Example of resource allocation when honeypots are located on node 6 and 9

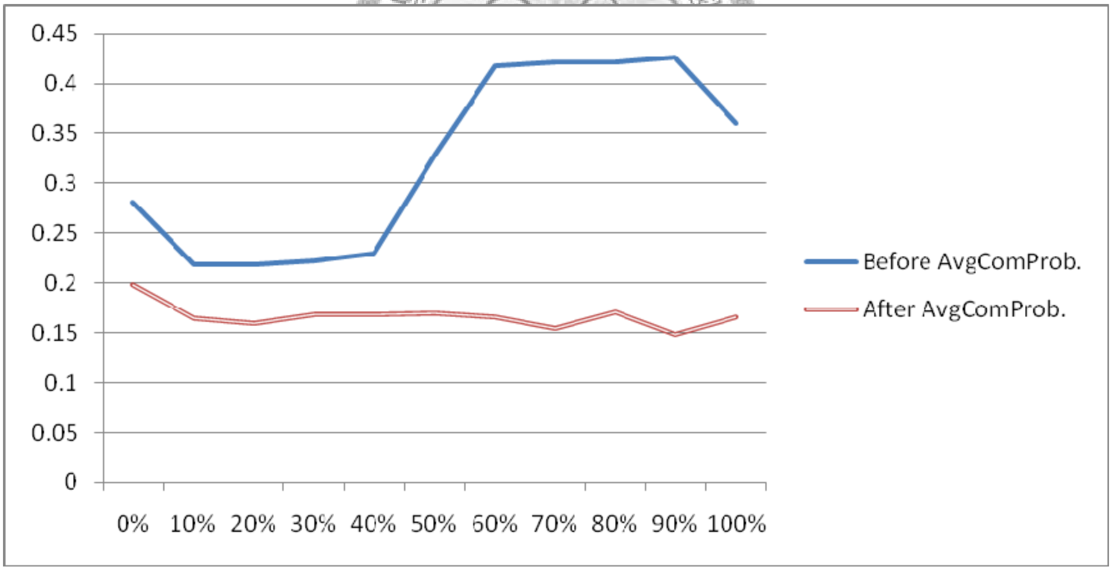


Figure 4-15: Performance Comparison when honeypots are located on node 6 and 9

Besides, we also verify the consequence of honeypots located on node 4 and 7. The benchmark of deciding attackers' budget is also 515. Figure 4-16 shows the performance comparison between initial and enhanced allocation. Our algorithm still performs well on this scenario. However, the resource allocation scheme is different. In this scenario, our algorithm concentrates resource to node 6 again. This is because there are two honeypots on the left side to defend attackers. Similar with previous case, we use topology features to force attackers compromise a node with high defense level. For attackers apply random strategy, we take advantage of honeypot characteristic to distract them. Therefore, most defense resource is allocated to node 6. Figure 4-17 illustrates the enhanced scheme when defender applies 10% number of hops and 90% link degree as initial allocation.

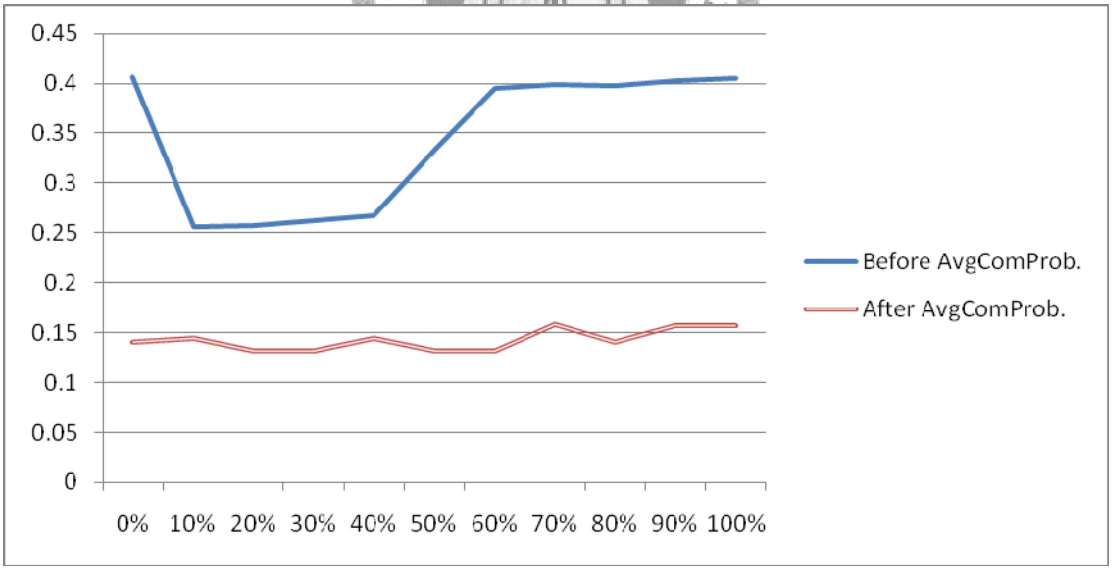


Figure 4-16: Performance Comparison when honeypots are located on node 4 and 7

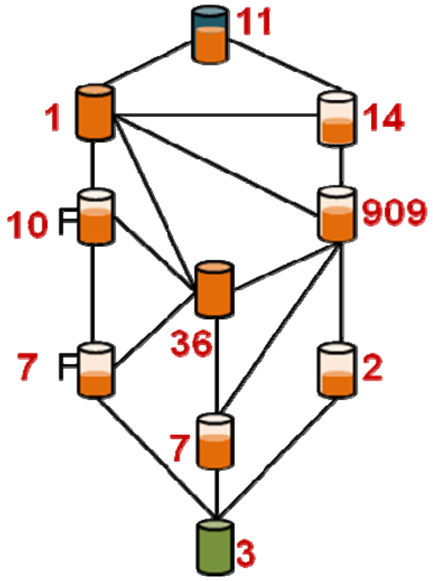


Figure 4-17: Example of resource allocation when honeypots are located on node 4 and 7

4.3 Scenario Analysis

In this part, we explain how our algorithm decrease the core node compromised probability. Also, by scenario analysis, we clearly demonstrate how our algorithm deals with attackers apply different next hop selecting criteria with distinct budget level and capability.

The scenario we are going to analyze is that defender applies 70% number of hops and 30% link degree to deploy defense resource as initial allocation. Besides, the minimum attack cost is fixed at 486 which is the one in applying 60% number of hops and 40% link degree initial allocation. Therefore, the high budget level attackers have 978 attacking power. For medium budget level attackers, they get 726 attacking budget and low level attackers only have 484 budget to attack. Then, we can start to demonstrate what the differences are between initial allocation and the enhanced one.

Figure 4-13 shows the basic information of the network. F is the site of false target honeypot and W is the location of honeypot wasting attackers' budget. S is the start point of attackers and t is the site of the core node. Figure 4-14 describes the initial allocation of this scenario. Black number refers to the defense level of this node and the red number means the budget defender allocated on it. We first demonstrate attackers apply lowest defense level as next hop election criteria.

Figure 4-15 shows attackers compromise the first node and select next hop from neighbors by lowest defense level. The defense level of candidate nodes are: 74, 74, and 65. Therefore, the node with 65 defense level becomes next hop of attackers to compromise. The corresponding situation is described in figure 4-16. After that, attackers start to compromise a false target honeypot shown in figure 4-17. Following the same criteria, figure 4-18 illustrates result of attack if attackers successfully penetrate the false target honeypot. We can discover that attackers will run out of budget before they compromise the core node even their budget level is high. In other words, this initial allocation has great performance on defending attackers apply lowest defense level as next hop selection criteria.

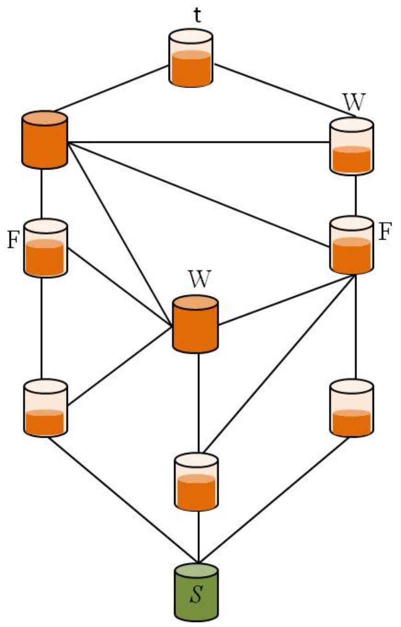


Figure 4-13: Basic Information of Network

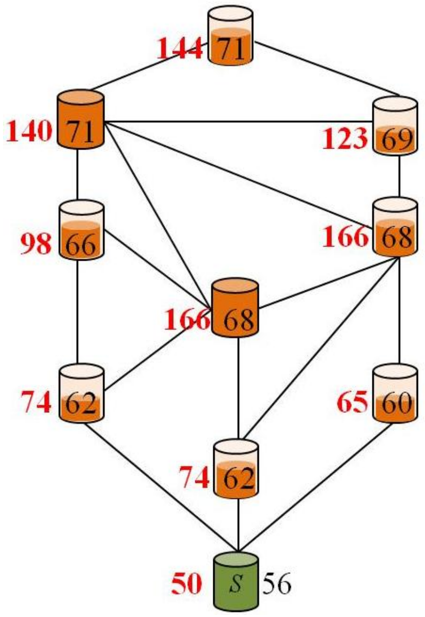


Figure 4-14: Initial Allocation of Network

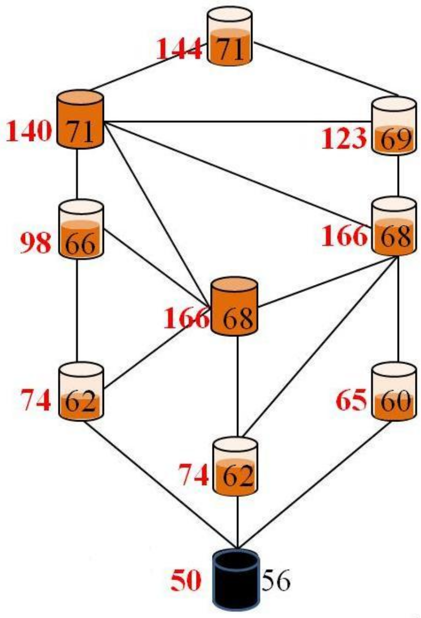


Figure 4-15: Next Hop Selection

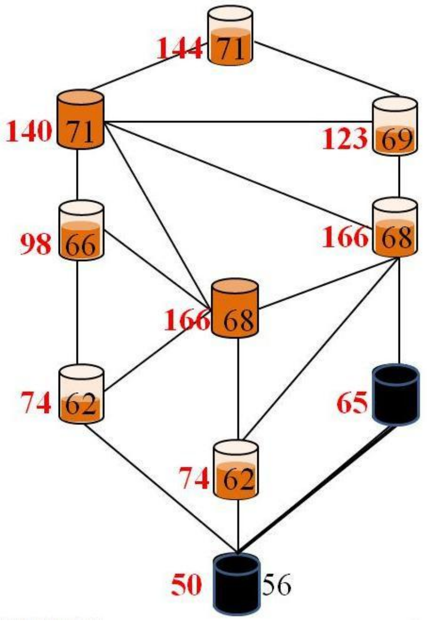


Figure 4-16: Compromising a False Target Honeypot

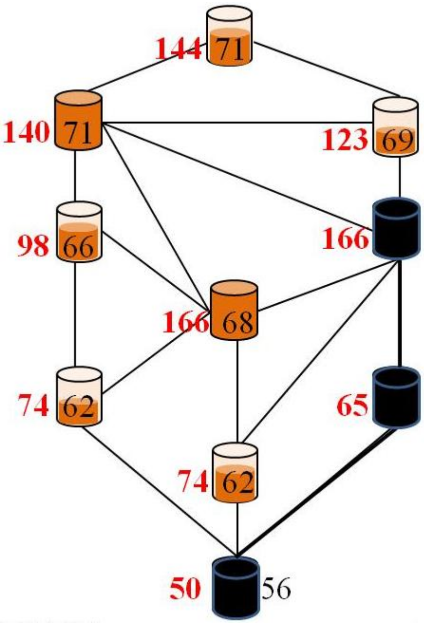


Figure 4-17: Compromised the False Target Honeypot

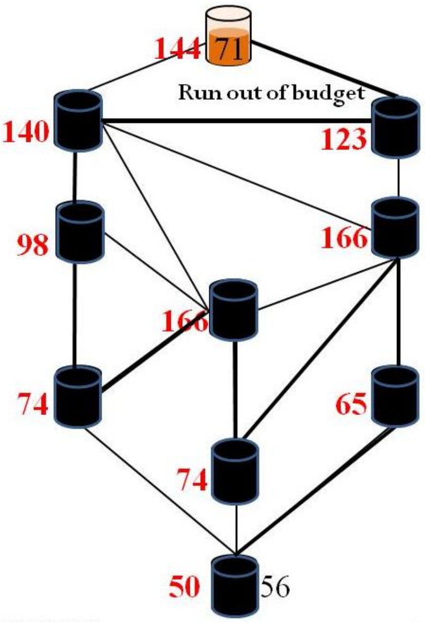


Figure 4-18: Result of Attackers Apply Lowest Defense Level as Next Hop Selection Criteria

Continually, we start to evaluate the performance of attackers applying highest defense level. Figure 4-19 shows attackers selecting next hop among nodes with 74, 74, and 65 respectively. Notice that if attackers get two candidates with the same defense level, they will randomly choose one from them. Assuming attackers select the one shown in figure 4-20 and continually compromise the rest node. In figure 4-21, we can find out attackers choose a false target honeypot as next hop to compromise. If they are not distracted by the honeypot, the result of attackers who apply highest defense level as next hop selection criteria is shown in figure 4-22.

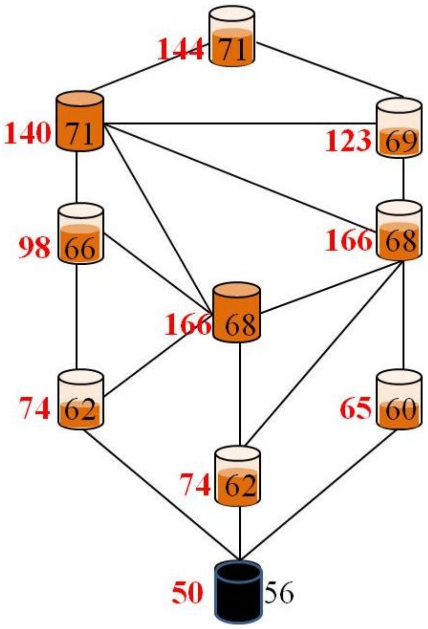


Figure 4-19: Next Hop Selection on Enhanced Allocation

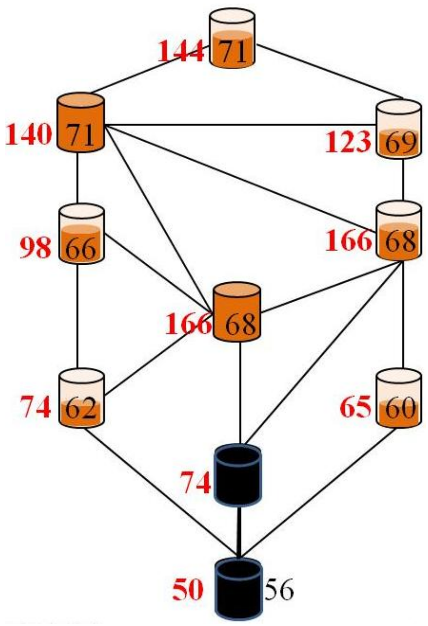


Figure 4-20: Assume Attackers Choose the Middle One

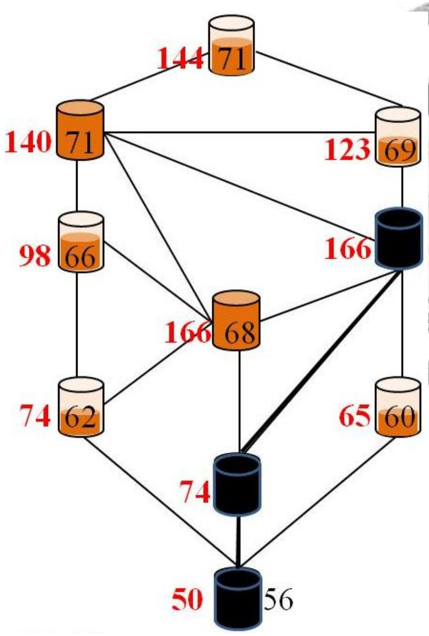


Figure 4-21: Successfully Penetrate the False Target Honeypot

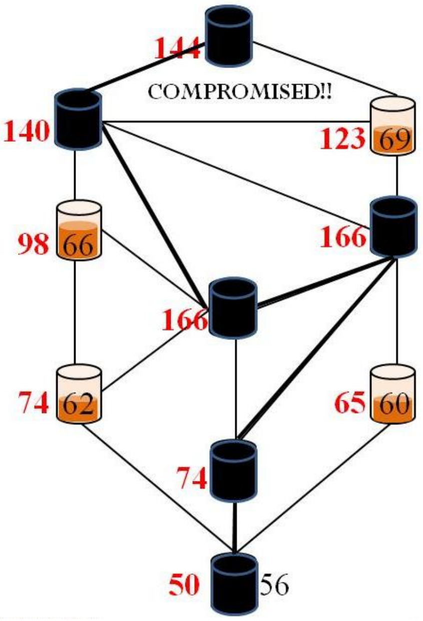


Figure 4-22: Result of Attackers Apply Highest Defense Level as Next Hop Selection Criteria

After understanding the scenario of initial allocation, the next step is to illustrate the corresponding situations of enhanced allocation. We also start with attackers apply lowest defense level as their next hop selection criteria. Figure 4-23 describes resource allocation of enhanced allocation. We can figure out enhanced allocation concentrate most defense resource on a false target honeypot. Furthermore, we also setup a trap which is the node with 0 defense level in figure 4-24 to lure attackers apply lowest defense level criteria to compromise. Since attackers apply lowest defense level criteria originally evade nodes with high defense level, they only intend to attack nodes with low defense level. However, our policy enhancement first setups a trap for attackers and coerces them to compromise a false target honeypot with high defense level. Even though attackers successfully penetrate the false target in figure 4-25, their attack budget has already been dramatically weakened. Following the same rule, attackers continually compromise other nodes.

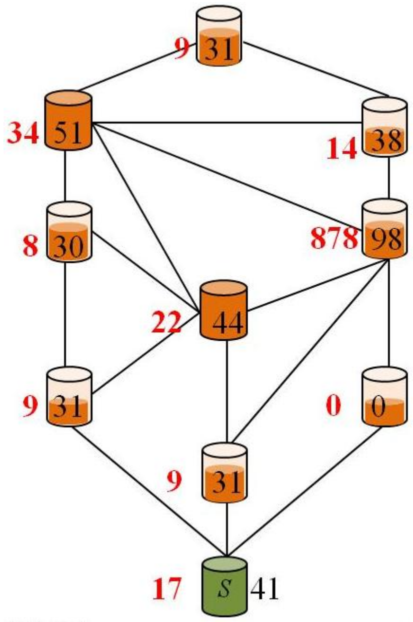


Figure 4-23: Information of Enhanced Allocation

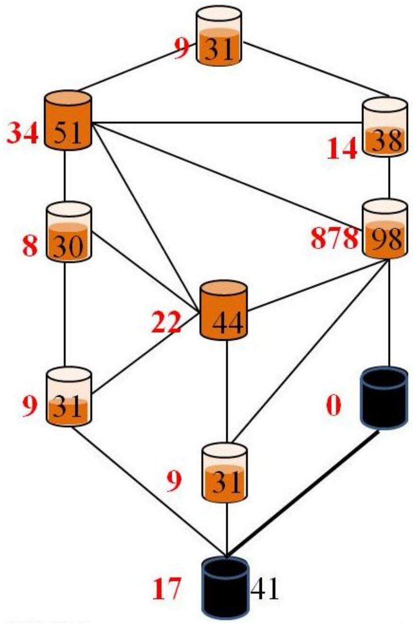


Figure 4-24: Trap to Lure Attacker to False Target Honeypot

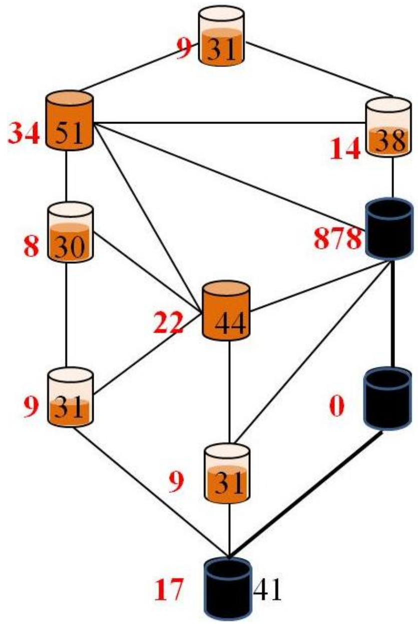


Figure 4-25: Attackers Have Sufficient Budget to Compromise the Node

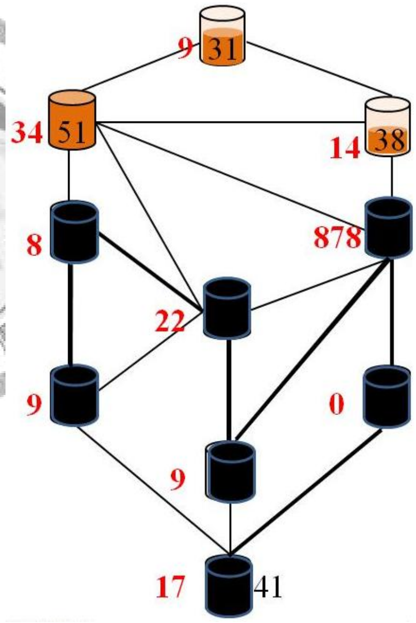


Figure 4-26: No Neighbors for Attackers to Select

However, in figure 4-26, attackers find out all neighbors have been compromised. Therefore, they start to trace back. Figure 4-27 shows they get back for one hop and find a candidate to compromise. The result of attackers apply lowest defense level on enhanced allocation is illustrated in figure 4-28. Similar consequence with initial allocation, enhanced allocation also has excellent performance on dealing with attackers apply lowest defense level as next hop selection criteria.

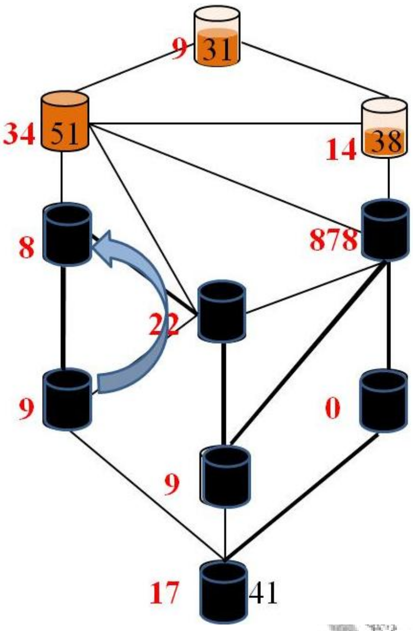


Figure 4-27: Trace Back to Former Node

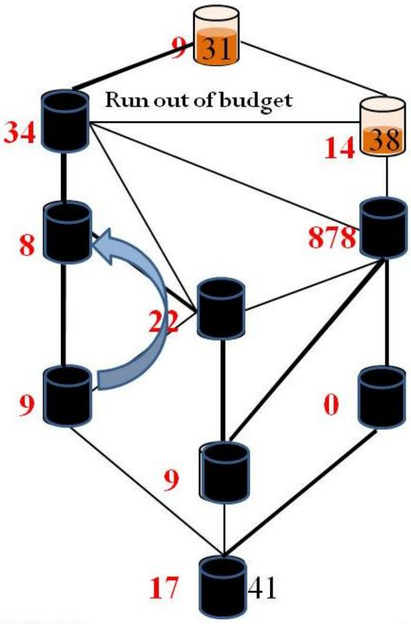


Figure 4-28: Result of Attackers Apply Lowest Defense Level on Enhanced Allocation

Finally, we illustrate the scenario when attackers apply highest defense level as next hop selection criteria. Similar situation with initial allocation, attackers randomly select next hop in figure 4-29. We first demonstrate one possible scenario that described in figure 4-30. With the same criteria, attackers find out they need to trace back again since there are no proper candidates to select. Figure 4-31 depicts they trace three hops to discover an ideal next hop. The result of attackers apply highest defense level on enhanced allocation is described in figure 4-32. It shows enhanced allocation still performs well on this scenario.

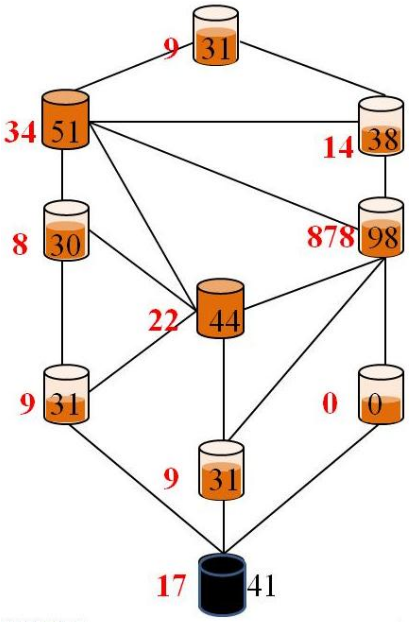


Figure 4-29: Next Hop Selection on enhanced allocation

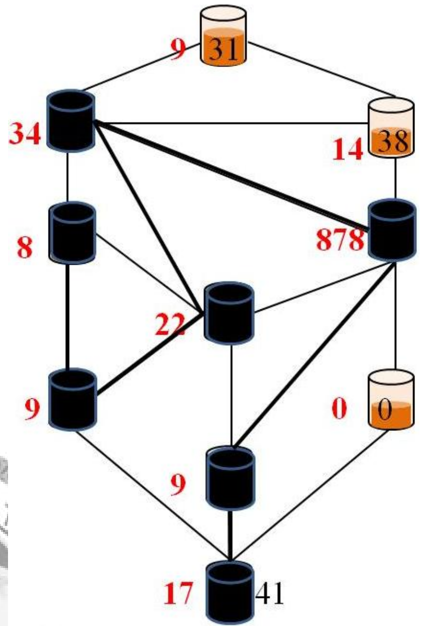


Figure 4-30: No Candidate to Choose for Next Hop

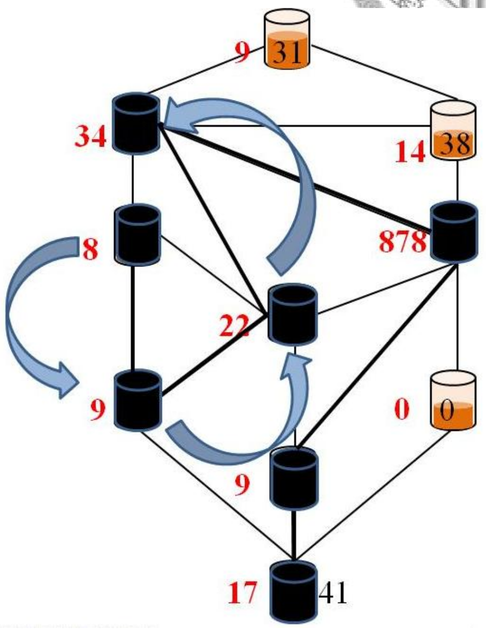


Figure 4-31: Trace Back for Proper Candidate

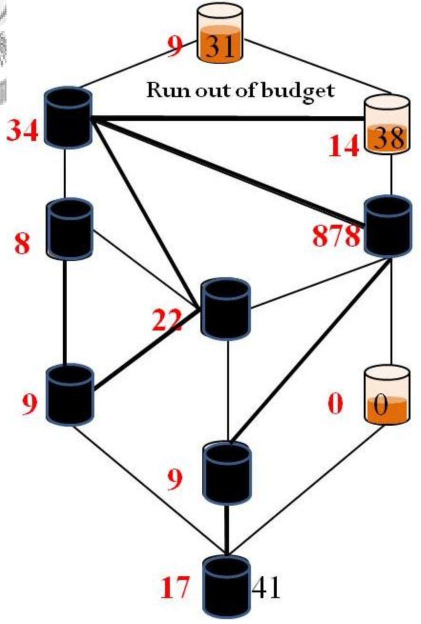


Figure 4-32: Result of Attackers Apply Highest Defense Level on Enhanced Allocation

However, attackers may choose the other path to attack. Therefore, the following demonstrates the scenario attackers take the other option to attack. The other possible scenario is shown in figure 4-33. Corresponding result of attackers apply highest defense level on enhanced allocation is illustrated in figure 4-34. Attackers are also failed to compromise the core node even they are not distracted by false target honeypot. In other words enhanced allocation has great performance no matter attackers apply highest defense level or lowest defense level criteria.

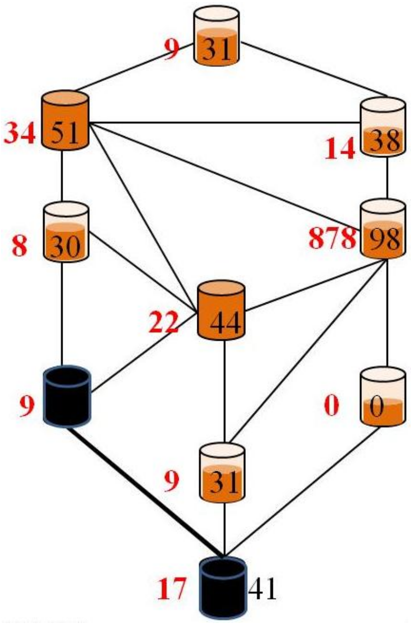


Figure 4-33: Another Attack Path

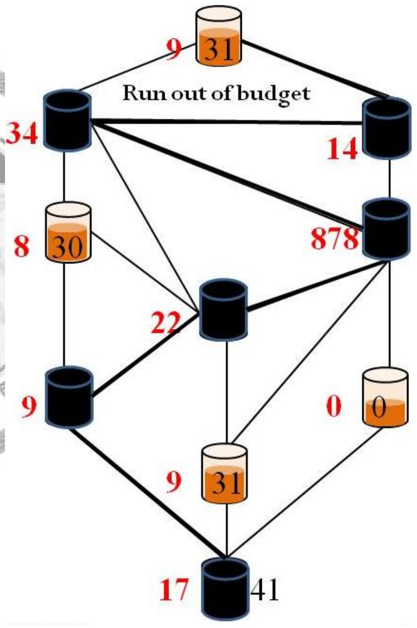


Figure 4-34: Result of the Other Attack Path on Enhanced Allocation

From above discussion, we can understand why the enhanced allocation performs much better than ignition one. Our policy enhancement not only allocates defense resource intelligently but also makes a trap to lure attackers to compromise the node with high defense level.

Chapter 5 Conclusion and Future Work

5.1 Conclusion

In this thesis, we first address the importance of imperfect knowledge and propose a generic model to evaluate network survivability. The metric we used in this model is the core node compromised probability. Furthermore, we also propose another specific model which has more detailed description on constraints. Another feature of this work is the classification of attackers. We evaluate survivability under average case which means the network is simultaneously attacked by different types of attackers.

The main contribution of this work is that we combine mathematical programming with simulation and develop a novel approach to solve problems with imperfect knowledge property. This mechanism helps us widely extend the scope of problems we can solve. Besides, this approach works steady at most cases even defender applies distinct initial allocation schemes. Even if there are some attackers' total budget is higher than defender's, our method still can reallocate defense resource effectively and reduce the core node compromised probability.

In addition, we can also setup a trap to lure attackers and force them to compromise nodes they originally are not going to attack. Moreover, we concentrate defense resource on a false target honeypot. Even though attackers successfully penetrate it, their attacking power still has been dramatically weakened. Therefore, the threat caused by attackers is decreased and system survivability is increased.

5.2 Future Work

We slightly point out some issues which can extend from this thesis and explain the corresponding concepts.

- **More concepts of mathematical programming**

In this thesis, we combine mathematical programming with simulation by derivative and popularity based strategy. However, we can still apply more concepts of mathematical programming into our approach or other techniques to rich this solution approach.

For example, concept of relaxation may be a choice. While solving the objective function, we can first relax some constraints. In other words, we may ignore some limitations and find the optimal solution of this problem. Then, use this solution as a hint to help us find the optimal solution of the primal problem.

- **Classification of attackers**

For attacker classification, we apply three categories to differentiate between attackers. Each category can be further divided into three levels. However, we can apply a statistical distribution to describe each category rather than discrete levels. While deciding attackers’ categories, we can randomly assign values on them.

This can help us extend the diversity of attackers. By random number, the total type of attackers is infinite. Therefore, we can evaluate network survivability in a generic way.

- **Secret sharing concept**

In our scenario, once attackers have compromised the core node, they can get all the valuable information. However, the concept of secret sharing is wildly used in practice. Confidential information is usually encrypted and separated store on different nodes. Therefore, attackers have to compromise multiple nodes to gain the important information.

For this scenario, we can build more next hop selection criteria for attackers to choose. This makes our scenario more reality.

- **Location of honeypots**

In this thesis, location of honeypots is fixed, our algorithm raises system survivability by reallocating defense resource on nodes. However, we know the location plays an important role in our scenario. Therefore, our next step is to extend our solution approach to take location of honeypots into consideration. In this way, we can not only find out optimal defense resource allocation but also discover optimal honeypot allocations.

- **Combine with network planning**

While evaluating system survivability, we can even contemplate network planning. For example, we may construct a long path for attackers but this path leads to nowhere. In other words, we can extremely consume attackers' budget. However, this method may result in poor reliability since nodes on this path only connect with the one right before and after. Once a link on this path has crashed, entire network is separated into two parts.

Therefore, how to balancing system survivability and reliability is the most important issue in this scenario.

In this research, we have addressed a real world attack defense scenario of information security. Nevertheless, the future research issues mentioned above can further enhance and rich our model and scenario. Future researches can follow in these directions.

References:

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

[2] C. Fung, Y.-L. Chen, X. Wang, J. Lee, M. Anderson, R. Tarquini, Richard L., “Survivability Analysis of Distributed Systems Using Attack Tree Methodology,” *IEEE Military Communications Conference*, Volume 1, pp.583–589, 2005.

[3] D. Zhou, S. Subramaniam, “Survivability in Optical Networks,” *IEEE Network*, Volume 14, Issue 6, pp.16–23, 2000.

[4] A. P. Snow, U. Varshney, A. D. Malloy, “Reliability and Survivability of Wireless and Mobile Networks,” *IEEE Computer Society*, Volume 33, Issue 7, pp.49–55, 2000.

[5] L. Kant, H. Kim, ,D.-P. Hsing, T.-H. Wu, “Modeling and Simulation Study of Survivability Mechanisms in WDM-Based High-Speed Networks,” *Global Telecommunications Conference*, Volume 1B, pp.1028–1034, 1999.

[6] D. Tipper, T. Dahlberg, H. Shin, C. Charnsripinyo, “Providing Fault Tolerance in Wireless Access Networks,” *IEEE Communications Magazine*, Volume 40, Issue 1, pp.58–64, 2002.

[7] S. Balasubramaniam, D. Botvich, W. Donnelly, N. Agoulmine, “A Multi-Layered

Approach towards Achieving Survivability in Autonomic Network," *IEEE International Conference on Telecommunications and Malaysia International Conference on Communications*, pp.360–365, 2007.

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

[9] http://www.net-security.org/secworld.php?id=6363

[10] http://www.symantec.com/region/tw/enterprise/article/mantrap.html

[11] C. Stoll, "Stalking the Wily Hacker," *Communications of the ACM*, Volume 31, No. 5, 1988.

[12] B. Cheswick, "An Evening with Berferd in which a Cracker is Lured, Endured, and Studied," *USENIX Conference*, pp. 163–174, 1922.

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

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

[15] H. Debar, F. Pouget, and M. Dacier, "White Paper: "Honeypot, Honeynet, Honeytoken: Terminological issues"," *Institut Eurécom Research Report*

RR-03-081, 2003.

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

[17] A. Réka, H. Jeong, A.-L. Barabási, "Error and Attack Tolerance of Complex Networks," *Nature*, Volume 406, pp. 378–382, 2000.

[18] http://www.blacksheepnetworks.com/security/info/misc/9907.html

[19] P.-H. Tsang, F.Y.-S. Lin, C.-W. Chen, "Maximization of Network Survival Time in the Event of Intelligent and Malicious Attacks," *IEEE International Conference on Communications*, pp. 1722–1726, 2008.

[20] http://honeypots.sourceforge.net/

[21] http://en.wikipedia.org/wiki/Honeypot_(computing)

[22] http://www.lib.iup.edu/comscisec/SANSpapers/msink.htm