

國立交通大學

資訊學院 資訊學程

碩士論文

虛擬化技術架構下的虛擬誘捕網路安全平台之  
設計與實作



Design and Implementation of Virtual Honeynet Security  
Platform based on Virtualization Technology

研究生：張志鴻

指導教授：蕭子健 博士

中華民國九十八年十一月

虛擬化技術架構下的虛擬誘捕網路安全平台之設計與實作

Design and Implementation of Virtual HoneyNet Security  
Platform based on Virtualization Technology

研究生：張志鴻

Student : Chih-Hung Chang

指導教授：蕭子健

Advisor : Tzu-Chien Hsiao



Submitted to College of Computer Science  
National Chiao Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of  
Master of Science  
in  
Computer Science  
November 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年十一月

# 虛擬化技術架構下的虛擬誘捕網路安全平台之設計與實作

研究生：張志鴻

指導教授：蕭子健 博士

國立交通大學 資訊學院 資訊學程 碩士班

## 中文摘要

傳統上誘捕網路(Honeynet)包括了多種類型的誘捕系統(Honeypot)，它在動態部署的彈性、時效、安全性與技術整合上有其管理及成本上的差異。在實務上軟硬體資源利用率也相較為低，同時缺乏有效的策略性整合應用，因此，仍然有許多可以改善與進步的空間。

本文主要目的在改善傳統 Honeynet 架構的設計方法與概念。以虛擬化技術(Virtualization Technology)為發展基礎，結合誘捕網路技術以及網路縱深防禦(Defense-in-Depth Network)的安全概念，設計出一個虛擬誘捕網路架構，並實作此虛擬誘捕網路安全平台(Virtual Honeynet Security Platform, VHSP)。另外，亦提出一種虛擬誘捕系統重導向機制(Virtual Honeypot Redirect Mechanism)，來解決誘捕系統模組因平行運作而導致繞過安全模組的問題。最後，採用 Nessus 弱點掃描工具來做網路模擬攻擊，並透過事件檢示器來獲得所有的模擬攻擊記錄。因此我們可以驗證：(1) VHSP 的虛擬網路及誘捕系統模組已重導向至安全模組。(2) VHSP 整體運作的可行性與實用性。

透過模組化的設計概念，可依需求來搭配 VHSP 內部的模組，如此靈活的應用將更優於傳統誘捕網路的設計。因此，VHSP 將具備 (1) 彈性應用-彈性且有效地運用與分配軟硬體資源；(2) 優質化管理- IT 管理與成本之優化考量；(3) 技術創新-整合綠色 IT 之節能應用策略。

關鍵詞：誘捕系統、誘捕網路、虛擬化技術、虛擬誘捕網路安全平台、虛擬誘捕系統重導向機制




# Design and Implementation of Virtual Honeynet Security Platform based on Virtualization Technology

Student : Chih-Hung Chang

Advisor : Dr. Tzu-Chien Hsiao

Degree Program of Computer Science  
National Chiao Tung University

## ABSTRACT



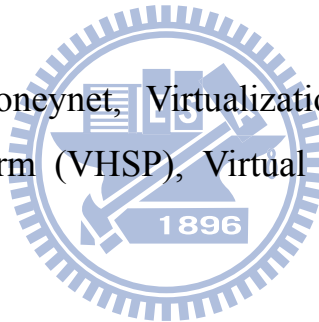
Conventional Honeynet includes various Honeypots; it has managerial limitation in the flexibility, time limit for dynamic deployment, technological integration and network security issues. In practice, software and hardware resources utilization is comparatively low, and meanwhile it also lacks of the effective application of strategic integration; therefore, there is still a lot of room for improvement and progress.

The main purpose of this thesis is to improve the designs and concepts of conventional Honeynet Architectures. Virtualization technologies are used as a platform for design and development, and combined with Honeynet technologies and the concept of Defense-in-Depth Network, a new Virtual Honeynet architecture is developed, and the Virtual Honeynet Security Platform (VHSP) is implemented. In addition, this work proposes and designs the Virtual Honeypot Redirect Mechanism (VHRM) for solving the problem of the Honeypot Module (HM) parallelism operation issue, and makes HM bypass the

Security Module (SM). Finally, using the Nessus vulnerability scanner for simulated network attacks, and next, through the event viewer can obtain all information about simulated attack records. Therefore we can verify : (1) The VHSP virtual networking and network transmission problem of HM bypass SM. (2) feasibility and practicality of the VHSP operation.

Through the design concept of modularization, each module could be customized according to different needs; VHSP could become more superior to the conventional Honeynet systems design. Therefore, the VHSP has: (1) application range - software and hardware resources could be used flexibly and efficiently; (2) Optimal Management - Optimization of IT management and cost considerations; (3) Technological Innovation - incorporating Green IT design strategies to save energy.

Keywords: Honeypot, Honeynet, Virtualization Technologies (VT), Virtual Honeynet Security Platform (VHSP), Virtual Honeypot Redirect Mechanism (VHRM).



# Acknowledgment

I have to express my sincere gratitude to the following person who made this thesis possible. I would like to specially thank my dear advisor, Prof. Tzu-Chien Hsiao for his guidance, advice and patience.

First of all, I would like to thank all of my lab mates at the Virtual Biomedical Management Laboratory (VBM Lab.) for providing valuable suggestions during the writing for this thesis. Besides, researching in the VBM Lab. has been a wonderful and substantial experience. Secondly, I would also like to thank my dear friends and colleagues, L.-N. Yen, Dr. Martin Peng, Yilang Tsai, Sam Chu, Sampo Tseng, and the IEEE PASSAT General Chair, Prof. Justin Zhan and the IEEE SMC Taipei Chair; Prof. Shun-Feng Su for their invaluable comments powerfully support and endless help. With their concern and efforts, the research really would have been possible. Thirdly, I would like to extend my heartfelt gratitude to all of my teachers and classmates at College of Computer Science in National Chiao Tung University. I also want to thank my committee members, my advisor, Prof. Tzu-Chien Hsiao, NCHC Director, Dr. Eugene Yeh, Prof. Yu-sung Wu, and Prof. Wen-Guey Tzeng, for all the comments they made on this thesis.

Finally, I would like to sincerely and gratefully thank my families for their continued and unlimited love is the best imputes and encouragement in my future work, study and research.

Chih-Hung (Jason) Chang  
November, 2009

# Contents

<b>Chinese Abstract</b>	<b>i</b>
<b>English Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Contents</b>	<b>vi</b>
<b>List of Tables</b>	<b>viii</b>
<b>List of Figures</b>	<b>ix</b>
<b>Chapter 1 Introduction</b> .....	<b>1</b>
<b>1.1 Motivation</b> .....	<b>1</b>
<b>1.2 Literature Study</b> .....	<b>1</b>
<b>1.3 Objective</b> .....	<b>3</b>
<b>1.4 Organization</b> .....	<b>4</b>
<b>Chapter 2 Background Information</b> .....	<b>5</b>
<b>2.1 Overview of Honeypot and HoneyNet</b> .....	<b>5</b>
2.1.1 Honeypot Architecture.....	5
2.1.2 HoneyNet Architecture.....	6
2.1.3 Conventional HoneyNet Composing Components.....	7
<b>2.2 Overview of Virtualization</b> .....	<b>8</b>
2.2.1 Full Virtualization.....	8
2.2.2 Hardware-assisted Virtualization.....	9
2.2.3 Xen Paravirtualization .....	9
<b>Chapter 3 Design and Implementation</b> .....	<b>11</b>
<b>3.1 Xen Virtual Machine Monitor Architecture</b> .....	<b>11</b>
3.1.1 Xen Virtual System Architecture.....	11
3.1.2 Xen Virtual Network Environment.....	13
3.1.3 Xen Virtual Network-Bridge Algorithm.....	14



<b>3.2 Conceptual Design of the Defense-in-Depth Network.....</b>	<b>15</b>
<b>3.3 VHSP based on Green IT Design Concept.....</b>	<b>16</b>
<b>3.4 Design of VHSP System and Network Architecture .....</b>	<b>17</b>
<b>3.5 Implementation of VHSP Modules .....</b>	<b>18</b>
3.5.1 Xen Module .....	19
3.5.2 Security Module.....	19
3.5.3 Honeypot Module .....	20
3.5.4 Database Module .....	20
3.5.5 Management Module .....	20
<b>3.6 Backup and Recovery Strategies for VHSP .....</b>	<b>20</b>
<b>3.7 Design Approach of VHSP Virtual Networking.....</b>	<b>21</b>
3.7.1 Design of VHRM based on HoneyNet Deploy Approach.....	21
3.7.2 Implementation of Isolated Virtual Network Module .....	22
<b>3.8 VHSP Virtual Networking.....</b>	<b>24</b>
<b>Chapter 4 Results and Discussions.....</b>	<b>26</b>
<b>4.1 Simulation Scenario.....</b>	<b>26</b>
<b>4.2 Simulation Validation .....</b>	<b>28</b>
<b>4.3 Comparison Results of HoneyNet Design Methodology .....</b>	<b>30</b>
4.3.1 Comparison of the Minimum HoneyNet Deployment Requirement.....	31
4.3.2 Comparison of the Minimums Number of Hardware Devices .....	32
4.3.3 Comparison of Different Feature of HoneyNet .....	33
<b>Chapter 5 Conclusion and Future Works .....</b>	<b>35</b>
<b>5.1 Conclusion .....</b>	<b>35</b>
<b>5.2 Future Works.....</b>	<b>35</b>
<b>References.....</b>	<b>37</b>
<b>Vita.....</b>	<b>39</b>

# List of Tables

Table 4.1: Minimums Honeynet deployment requirement..... 31

Table 4.2: Minimums number of hardware devices of Honeynet comparison results..... 32

Table 4.3: Comparison of different features of Honeynet..... 33



# List of Figures

Figure 2.1: The architecture of conventional Honeypot network environment. ....	5
Figure 2.2: Conventional Honeynet architecture.....	6
Figure 3.1: The architecture of Xen 3.x Hypervisor environment. ....	12
Figure 3.2: Xen Virtual Network-Bridge path and virtual ethernet interfaces.....	15
Figure 3.3: The concept of Defense in Depth Network.....	16
Figure 3.4: The virtual system and network architecture of VHSP operation based on XenoLinux environment.....	18
Figure 3.5: The virtual architecture of VHSP modules based on XenoLinux environment. ..	19
Figure 3.6: Design of Isolated Virtual Network module based on VHRM.....	22
Figure 3.7: Flow chart of VHSP design approach.....	23
Figure 3.8: The architecture of VHSP Module and Flow Design. ....	25
Figure 4.2: The Web Management Interface status monitor which is based on Virtual Machine Monitor environment.....	29
Figure 4.3: Virtual Machine Monitor based on VHSP Monitor interface.....	30
Figure 4.4: Comparison of the performance of Native Linux (L), XenoLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).....	34

# Chapter 1 Introduction

## 1.1 Motivation

Conventional Honeynet includes various Honeypots; it has managerial limitations in terms of flexibility, technological integration, and dynamic deployment. In addition, its software and hardware resource utilization is comparatively low, and it also lacks of the effective application of strategic integration and has relatively high equipment cost. Therefore, it has room for improvement. This study reports the integration of Virtualization Technology (VT) and Honeynet technologies with a Defense-in-Depth Network which is then properly deployed to a physical network environment and deploy in an Internet Gateway or other network security research and applications.



## 1.2 Literature Study

Founded in 1999, the Honeynet Project [1][2][3][4][5] is an international Non-Profit Research Group that focuses on strengthening research on network and information security technologies. This project is dedicated to improving the security of the internet at no cost to the public, and has formed several chapters around the world. This project has jointly developed many relevant software instruments and trapping technologies. [3][4][5][6]. For example, the Honeypot has been adopted by researchers mostly within a physical network environment.

Honeynet includes various Honeypots [1][3][5]. This approach allows researchers to make use of a great quantity of physical hardware equipment that would otherwise consume too many computer system and network resources. Thus, Honeynet enables technological

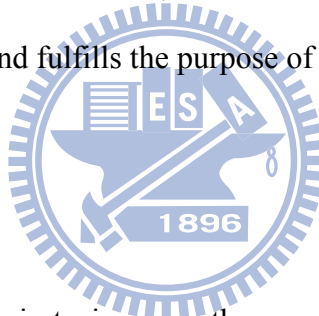
integration in the design, flexibility, and dynamic deployment of system architecture for various design issues.

Virtualization Technologies (VT) has been developed into a matured and important technology. [7][8][9] The earliest concept of virtualization was proposed by IBM and was implemented on the Mainframe of IBM system 360-67 in 1960 [5][10]. So far, VT has been definitely an important tool and technology for the design and evolutionary development of computer systems [8]. VT includes three types of technologies, including Full VT [5][10], Hardware-assisted VT [11], and Paravirtualization [12][13]. Full VT is operated directly in a physical operating system and is used as a Virtual Machine Monitor (VMM) [9][14] to control system resources without modifying Guest Operating System (Guest OS) or Application Programs (APs), contrast to paravirtualization where the guest kernel needs to be modified. Hardware-assisted VT was recently added to x86 processor from 2006 in the form of Intel ® VT-x [10][13] and AMD-v™ [11]. Both of these technologies have a new implementation mode in the CPU, can use the hardware-assisted VT function of the physical operating system, and also support many of the operating system platforms in Virtual Machine (VM) systems. Paravirtualization is acknowledged as the most rapid and safest software virtualization technology in the industry at present. This approach only requires about 10% of the system consumption and requirements of Full VT. For the basic testing of the Xen open source Virtual Machine (VM), it generally consumes less than 5% of system resources, and is regarded as a breakthrough for related technologies [12][15].

This thesis proposes a cost-effective design for information security research domain and a usable Virtual HoneyNet Security Platform (VHSP). The all-in-One architecture has greater flexibility and usability. This study also proposes a design approach for a Virtual HoneyPot Redirect Mechanism (VHRM). Compared with a conventional HoneyNet, deploying HoneyNet as a benchmark to quantify this thesis proposes VHSP that only needs about 33% of

hardware requirements, for a savings of approximately 66%, thus, this work has contributed a better operating strategy of software and hardware resources. The main contribution of this thesis is that it provides better strategies for applying resources and reducing costs, while providing new improved solutions for conventional Honeynet systems. Thus, in terms of applicability, this study provides researchers and an innovative usage model for the design and application of conventional Honeynet. Moreover, in terms of resource management and costs, the proposed model could reduce the complexity of deployment and excessive use of human resource, time, and costs incurred by conventional Honeynet, while reducing the space and removing restrictions. Finally, from the perspective of technological development and innovation, our proposed VHSP is not only a cost-effective design but also a more flexible security research platform, and meanwhile, it also follows the latest developing concept of Green IT [16][17][18] design and fulfills the purpose of eco-friendliness.

### **1.3 Objective**



The purpose of this work is to improve the concepts and strategies for the Honeynet architecture and Honeynet system design. The key objectives of this study are to strengthen network security and provide a usable, flexible platform for security researchers. The Green IT design concept is applied to the VHSP.

The project has come up papers published at the 2009 IEEE International Symposium on Secure Computing (SecureCom'09), Vancouver, Canada. [19], and the 2009 IEEE International Conference on Systems, Man, and Cybernetics (SMC'09), San Antonio, Texas, USA. [20]. The paper on IEEE SecureCom'09 was published on August 29-31, 2009, and the one on IEEE SMC'09 was published on October 11-14, 2009.

## 1.4 Organization

The work is organized into five parts; Chapter 2 introduces the basic concepts and background information of the Honey pot, Honey net, virtualization technologies, Defense-in-Depth Network [21], Green IT design concept, and related work. Chapter 3 shows how the Paravirtualization and Hardware-assisted VT support VHSP operations design and organize the concepts of DDN and Green IT design on the proposed platform. Next, a new Virtual Honey net Security Platform (VHSP) and a Virtual Honey pot Redirect Mechanism (VHRM) are developed. The experiments described in Chapter 4 are conducted on the basis of the feasibility and availability of this 5-in-1 new architecture. The simulation scenarios for VHSP validation use Nessus [22] vulnerability scanning tool to verify our platform, network path and VHSP modules, and also check and review the virtual networking and event logs from Web Management Interface of VHSP. This study compares the results of various Honey net design methodologies, including the minimum Honey net deployment requirement, the minimum number of H/W devices, and different Honey net features. The final chapter briefly summarizes this study and provides a discussion of future work.

# Chapter 2 Background Information

This chapter provides the necessary background required to appreciate the work presented in this thesis. All of Honeypot and HoneyNet technologies are presented in Section 2.1. Virtualization Technologies is given in Section 2.2.

## 2.1 Overview of Honeypot and HoneyNet

### 2.1.1 Honeypot Architecture

Figure 2.1 shows the conventional trapping system [5] or is also called Honeypot adopted by researchers that mostly within a physical network environment, and established in a public/opened network space in accordance with the concept of a distributed and independent single system.

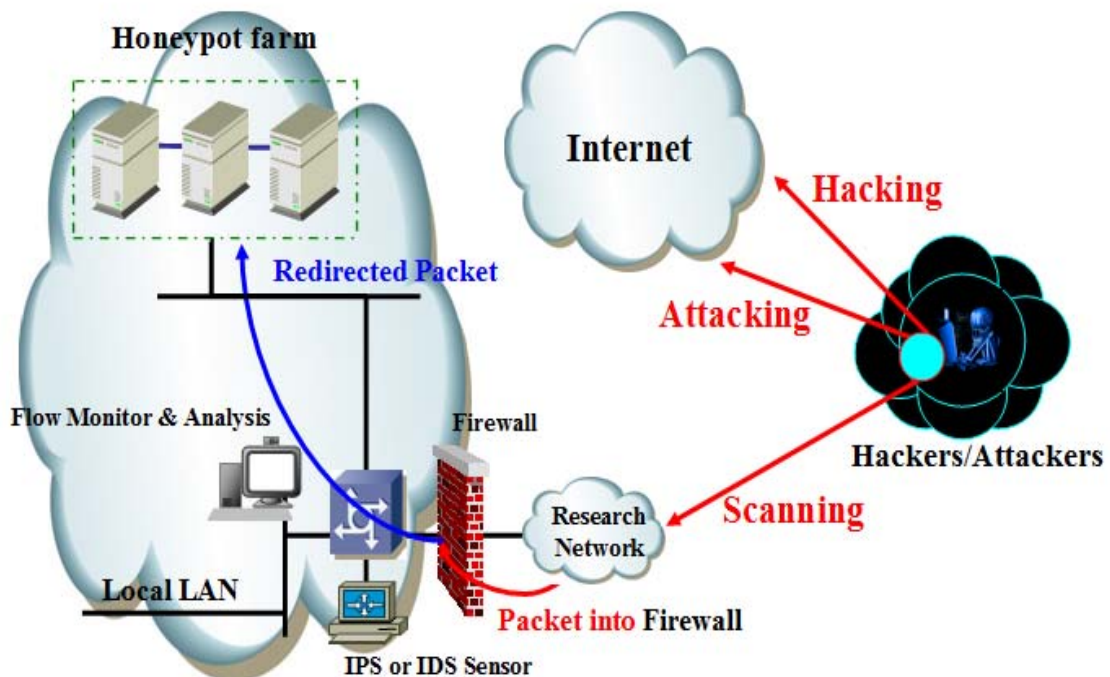


Figure 2.1: The architecture of conventional Honeypot network environment.



## 2.1.2 Honeynet Architecture

As for the main purpose of Honeynet, it is a research tool for the information and network security research with conducting analysis and studying on the attacking behavior and method of the intruders on the Internet. Most of the Honeynet architectures [3][5] are in the public or opened physical network environment, and are the network that composed of many Honeypot.

Honeynet is able to operate the physical operating system environment [1], and currently, most of them is designed in the High-interaction Architecture. In addition, it can be mixed to sue with Honeyd [14], which is used to simulate the operating system and service software and is categorized as the Low-interaction system architecture.

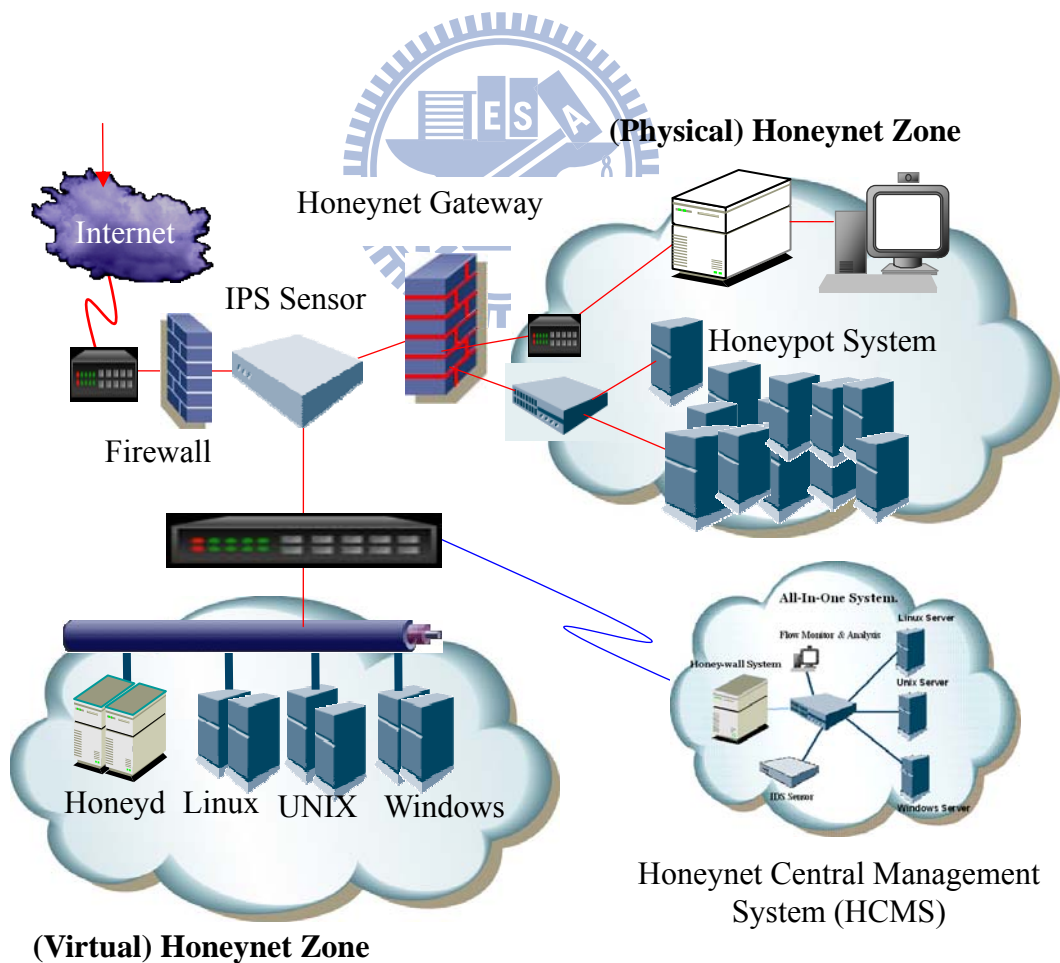


Figure 2.2: Conventional Honeynet architecture.

In figure 2.2, the conventional Honeynet environment or also called Honeynet System adopted by researchers that mostly within a physical network environment establishes in a public or opened network space in accordance with the concept of a distributed and independent single system. The Honeynet includes various Honeypots that include many components. In figure 2.2 shows two Honeynet environments that based on different architectures. The Honeynet architecture that includes two major design approaches: Honeynet environment based on virtual architecture and the other is based on physical devices and real network environment.

### 2.1.3 Conventional Honeynet Composing Components

The five major Honeynet components such as :

- **Firewall** : The main function of Firewall is to control all internal and external network packets for whether they conform to the Firewall Security Policy, inspect the Network Protocols, IP Address and Port.
- **IPS** : Intrusion Prevention System (IPS) [23] is mainly functions to identify the threats in network and can effectively defend through the deep packet inspection technology. If any violation of the IPS Security Policy occurs, administrator will regard this as an abnormal warning and return the information back to the monitor end.
- **Honeypot** : The Honeypot seems like an insecure system with defects and vulnerabilities. Its main purpose is to be used as a trap against the intrusion activity, or an early warning mechanism for the network security defense.
- **Backend DB System** : The major function of the Database System is to store related system data, system event, system log and all information into this backend database system.
- **Monitor & Management System** : The primary function of this system is to monitor and collect all data and to log from various software tools, equipments and devices. It can be defined in accordance with monitoring scope.

## 2.2 Overview of Virtualization

The virtualization technologies (That includes Physical Operating System (OS) and many types of computer systems and platforms) are used to consolidate multiple physical machines onto a single physical machine. In other words, the multiple virtual machines (VMs) can run on a single physical machine.

VT is a matured and important technology and also an innovate methodology in Computer Science domain, and meanwhile it is also a low-cost concept at present. The earliest conception of virtualization was a method of that proposed by IBM in 1960, and set up on the Mainframe of IBM System 360-67. VT has definitely become an important tool and technique [8] for the design and evolutionary development of computer systems. Furthermore, VT enables security researchers to run multiple operating systems (OS) concurrently on a single physical machine, where each of the OSs runs as a self-contained computer. VT can be used for research or support of business to cost-effective utilization of IT infrastructure. However, in this thesis, VT is a fundamental technological innovation that allows skilled IT security experts or security researchers to apply and design creative new solutions to such security issue challenges. Virtualization includes various types of technologies; this Section will briefly introduce the Full VT, Hardware-assisted VT and Paravirtualization. Those are presented in Section 2.2.1 to Section 2.2.3.

### 2.2.1 Full Virtualization

Full VT will establish a whole new virtual operating system, also known as Guest OS, which is able to operate directly in the local operating system, and can make use of VMM [7] to control the system resources without needing to modify the Guest OS or application program (Apps). Currently, Full virtualization Technology still adopt the Binary Translation

(BT) [11] approach, as a result of implementing CPU command under the Ring 0, thus the hardware equipment of the lowest level can be directly accessed and then sent to VMM for further implementation. The advantage of the full virtualization is that it is able to establish most diverse platforms without modifying the kernel of Guest OS; however, relative lower efficiency is its weakness.

### 2.2.2 Hardware-assisted Virtualization

Currently, Intel ® VT-x and AMD-v™ are two types of the hardware virtualization technologies at present. This approach is added to x86 processors in 2006. Both of them was added a new implementation mode into CPU, which is called as the root mode. Such mode can make Virtual Machine Monitor (VMM) [7] to operate under the root mode and it locates beneath Ring 0 and on the same layer as VMM. The status of Guest OS will preserve in the Virtual Machine Control Structure or the Virtual Machine Control Block of AMD-v™ CPU that supports Intel ® VT-x and AMD-v™ can use the hardware-assisted virtualization function.

### 2.2.3 Xen Paravirtualization

Xen Paravirtualization is acknowledged as the most rapid and safe software virtualization technology in the industry at present. Compared with full virtualization, it only needs about less than 10% system efficiency consumption and requirement. As for the basic testing of the Xen Source Virtual Machine, it generally turns to consume less than 5% of system efficiency, and that can be regarded as a breakthrough for related technologies. On the contrary, the system efficient consumption with adopting other VT will be reached 35%, or even higher [12][15]. Therefore, the main advantage of using the Paravirtualization is that it

can provide a higher application scope of efficiency than the full VT, but the weakness is the need to modify its Guest Operating Systems kernel.

Xen [12][13] initiated its development in 2002, and developed by the x86 platform, the Computer Experiment Lab of University of Cambridge, UK. In addition, based on the open-source software, and is conformed to the agreement of GNU Public License (GPL) to do the development. Its main purpose is to make use of the most simplified approaches to modify the current operating systems in the current x86 Architecture; at the same time, perform more optimum virtualized efficiency in the current virtualization technologies [9][14].

Compared with the VMware [7][9] that proposed in 2005, and it is a Paravirtualization interface, or named as the Virtual Machine Interface (VMI) [14]. The Paravirtualization technology that adopted by Xen, the Xen Guest OS Kernel [12] can only be operated in the Xen ® Hypervisor [10][12][14][15], but VMI Guest OS can be supported to various hypervisors. If adopting the Xen Paravirtualization, then it needs to re-compiler and modify the Operating System Kernel for Guest OS. Therefore, due to their respective advantages and defects, they can be flexibly deployed in accordance with the real requirement.

The leader and founder of Xen Source R&D Team, Dr. Ian Pratt [12] indicates that the main feature of Paravirtualization is that it can provide the similar speed to its operating system. Thus, Paravirtualization possesses a higher calculation environment, which also the most rapid and secure virtualization technologies at present.

Xen adopted the Borrowed Virtual Time scheduling algorithm (BVT), which was proposed by Kenneth J.D., and David R.C.[24] in Stanford University in 1999. And the main purpose of applying such algorithm to Xen by its R&D Team is to reduce the system events that may influence the operating efficiency for the virtualization system. When a domain received an event, such algorithm possesses more low-latency after event occurred.

# Chapter 3 Design and Implementation

Using the Xen and Honeynet open-source software as the basis; this work mainly use the Paravirtualization and Hardware-assisted VT to support VHSP System architecture and network environment. In addition, as for the virtual Honeynet components, the security module adopted open-source software of Honeywall [2] for design and implementation, and meanwhile the concept of DDN and the concept of Green IT design are used in ours platform.

Xen virtual machine monitor architecture is presented in Section 3.1. Section 3.2 introduces the conceptual design of the Defense-in-Depth Network. Section 3.3 describes Green IT design concept. The VHSP system and network architecture overview is presented in Section 3.4. Section 3.5 introduces how to design and implement the VHSP Modules in our virtual system architecture. Section 3.6 describes the backup and recovery strategies for VHSP. Section 3.7 describes how to design the Virtual Honeypot Redirect Mechanism (VHRM). In order to solve the Guest OS parallel issues, this work proposes a VHRM and implement an Isolated Virtual Network (IVN) module based on the Honeynet deploy approach that can redirect packets of HM to NBD of the Xen Dom 0. Section 3.8 provides an overview of the VHSP framework and networking.

## 3.1 Xen Virtual Machine Monitor Architecture

### 3.1.1 Xen Virtual System Architecture

Xen virtual system, as defined in figure 3.1, includes, Xen managerial programs, virtual domain, virtual domain management and control module, Paravirtualization and hardware virtual machine Guest OS.

The major components of Xen [12] [13] [14] system architecture is shown as follows:

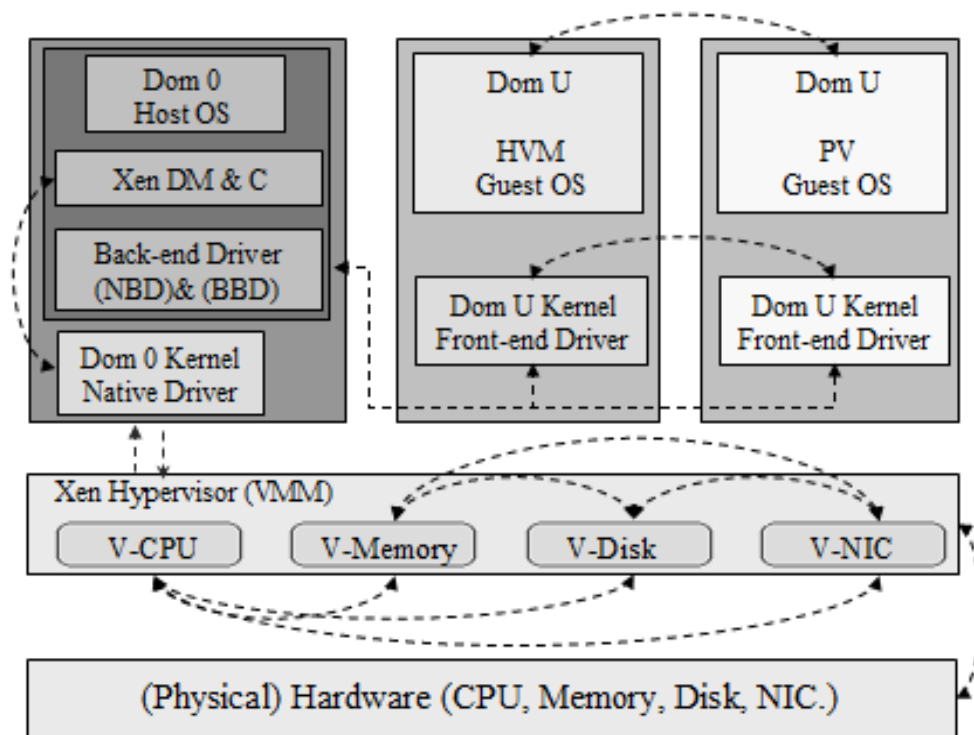


Figure 3.1: The architecture of Xen 3.x Hypervisor environment.

- **Hardware** : The Physical hardware resources. (It includes all of the computer components and physical device).
- **Xen Hypervisor VMM** : Xen Virtual Machine Monitor (Xen VMM) or named as Xen ® Hypervisor, its major function is to allocate and manage the hardware resources to each Guest OS. In Xen ® Hypervisor, through Hypercalls [12], it can request services from the system, such as using the system call to communicate in the kernel of physical operating system. The Virtual Components are based on OS resources that include Virtual CPU, Virtual Memory, Virtual Disk, and Virtual Network Interface Card (NIC) etc..
- **Dom 0** : Domain 0 also known as Dom 0, which possesses special authority in the Xen architecture, and can access physical I/O resources and coordinate the system and network operations of Guest OS. Dom 0 includes two drivers. They respectively are (1) Network Back-end Driver (NBD) and (2) Block Back-end Driver (BBD). Dom 0 environment is based on Dom 0 kernel.

- **Xen DM & C** : Xen Domain Management & Control (Xen DM&C), the major function of Xen, is conducted comprehensive control and management for all system services, processes and virtual domain environment on all Dom U. (Dom 0 Kernel and Dom U kernel).
- **Back-end Drivers (BDs)** : Includes 2 drivers, both Network Back-end Driver (NBD) and Block Back-end Driver (BBD), where NBD is responsible for network communication and BBD is for the disk storage.
- **Front-end Drivers (FDs)** : This is in the Virtual environments and the Front-end Driver of Dom U, and it will respond for the Back-end Driver of Dom 0.
- **Dom U** : This includes PV Guest OS and HVM Guest OS. Dom U environment is based on Dom U kernel.
- **PV Guest OS** : Paravirtualization Guest OS: All of the virtual machines will be called as the Domain U or Dom U or Paravirtualization (PV) Guest OS, and XenLinux that operated in the Xen Hypervisor. Its System Kernel should be the re-compiler of Xen.
- **HVM Guest OS** : Hardware Virtual Machine Guest OS will simulate the BIOS of physical operation system through the Xen Virtual Firmware.

### 3.1.2 Xen Virtual Network Environment

The Xen virtual network [14] as defined in the following :

- **Xen Virtual Network-Bridge** : Dom 0 Network Default adopts the Virtual Network-Bridge technology to preset all network communication of Dom U, and it should be connected to the network using the outbound bridge method.
- **Xen Virtual Network-Route** : Xen Virtual Network-Route will generate the routing table for all Dom U in the Dom 0, and use IP route approach to communicate physical network.
- **Xen VLANs** : Xen Virtual Local Area Network (Xen VLAN) supports multiple tagged IEEE 802.1Q VLANs technology. It is mainly used to conduct the network subnet segregation for Dom U. Meantime; it can also communicate other Dom U with network cross-subnet.



### 3.1.3 Xen Virtual Network-Bridge Algorithm

The Xen Virtual Network-Bridge (XVNB) algorithm and procedure are described as follows:

**Algorithm** [Xen Network-Bridge]

**Output** : The physical interface and the dom0 interface separated.

**Method** :

```
1: if (There is no XVNB) then
2:   Creates the new bridge of XVNB (xenbr0) /* Xen Network-Bridge = xenbr0 */
3:   Stop and shutdown the Real eth0 (Original IP:MAC)
4:   /* In order to keep the Original IP:MAC address in Real eth0 */
5:   if (Real eth0 is brought down) then
6:     Duplicate the Real eth0 (Original IP:MAC) → Virtual eth0
7:   while (Virtual eth0 (IP:MAC) = Real eth0 (Original IP:MAC)) do
8:     Creates the peth0 of XVNB
9:     peth0 ← Rename the Real eth0 (Original IP:MAC)
10:    /* The peth0 is only support for MAC Layer */
11:    Real eth0 (IP:MAC) ← Virtual eth0 (IP:MAC) is renamed
12:   if (Real eth0 (IP:MAC) = peth0 (MAC)) then
13:     Creates the vif 0.0 of xenbr0
14:     xenbr0 ← Attaches both of the peth0 and vif 0.0
15:   if (Both of the peth0 and the vif 0.0 is bound to xenbr0) then
16:     Restart the Real eth0, peth0, vif 0.0, and xenbr0
17:   else (There is XVNB) then
18:     Not do any of that peth0 stuff, nor will it add vif 0.0 → xenbr0 yet
19:   end if
```

At step 1, Xen will creates a new virtual network-bridge named *xenbr0* (*xenbr0* in the VHSP based on XenNetworking). At step 2, the Real Ethernet interface *eth0* is brought down. Next, at step 3, the Real *IP* and *MAC* address of *eth0* are duplicated to *Virtual eth0* virtual

interface in XenLinux. At step 4, the Real Ethernet interface *eth0* is renamed *peth0* virtual interface. At step 5, *Virtual eth0* virtual interface is renamed *eth0* real network interface. At step 6, the *peth0* virtual interface and *vif0.0* virtual interfaces are attached to bridge *xenbr0*. Finally at step 7, the Real Ethernet interface *eth0*, *xenbr0*, *peth0*, and *vif0.0* are brought up. The Xen Virtual Network-Bridge path and virtual Ethernet interfaces is shown in figure 3.2.

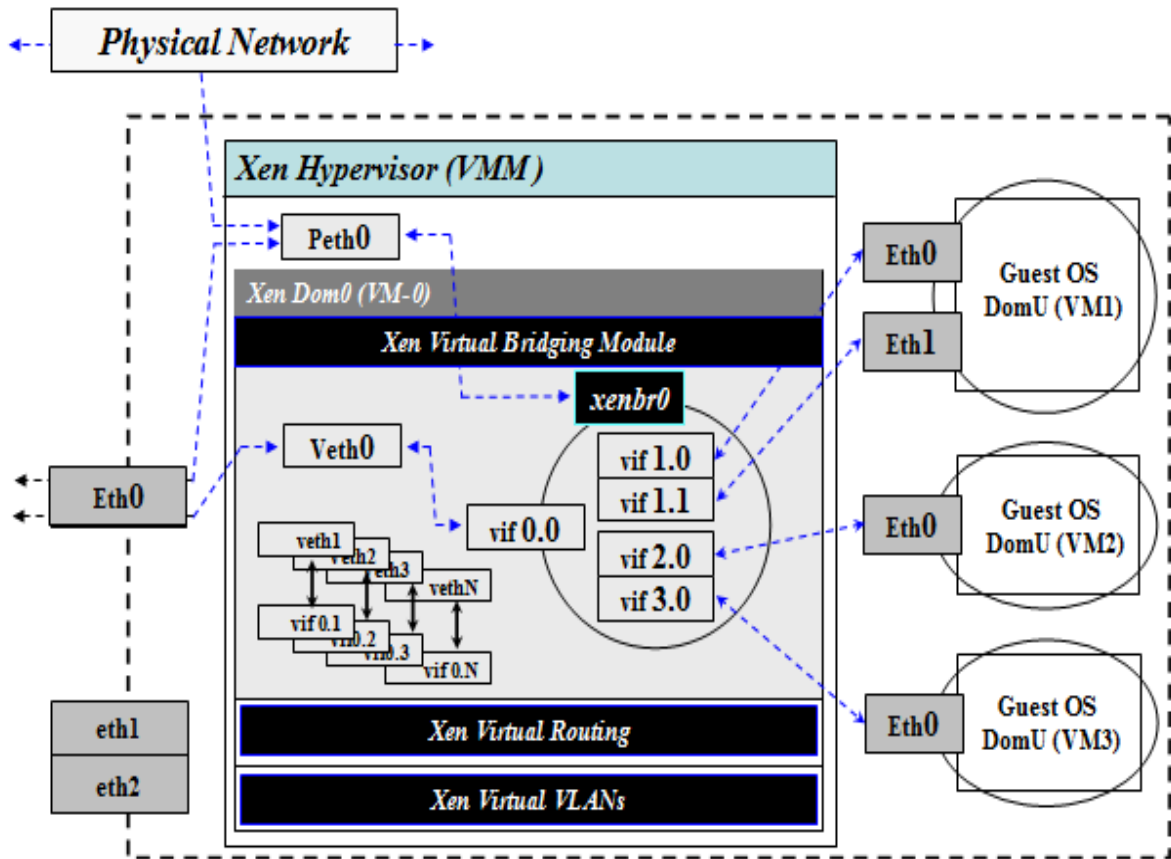


Figure 3.2: Xen Virtual Network-Bridge path and virtual ethernet interfaces

### 3.2 Conceptual Design of the Defense-in-Depth Network

Virtualization Technologies (VT) is used as a platform for design and development. Combined with Honeynet Technologies (HT) and the concept of Defense-in-Depth Network (DDN) [21], a new Virtual Honeynet Architecture can be developed.

The main function of the DDN is to strengthen and enhance the defense level for

network security. As shown in figure 3.3, It is not a single barrier (Zone 1, 2, 3.) that can be intruded, but it is a complex and multilayer defense system (Security Check Point 1, 2, 3.) with integrating various information security technologies and security policies.

It is an integrated application that includes various network security components, such as Firewall, IPS, Honeypot, Monitor System, and various applicable software defense systems and security policies.

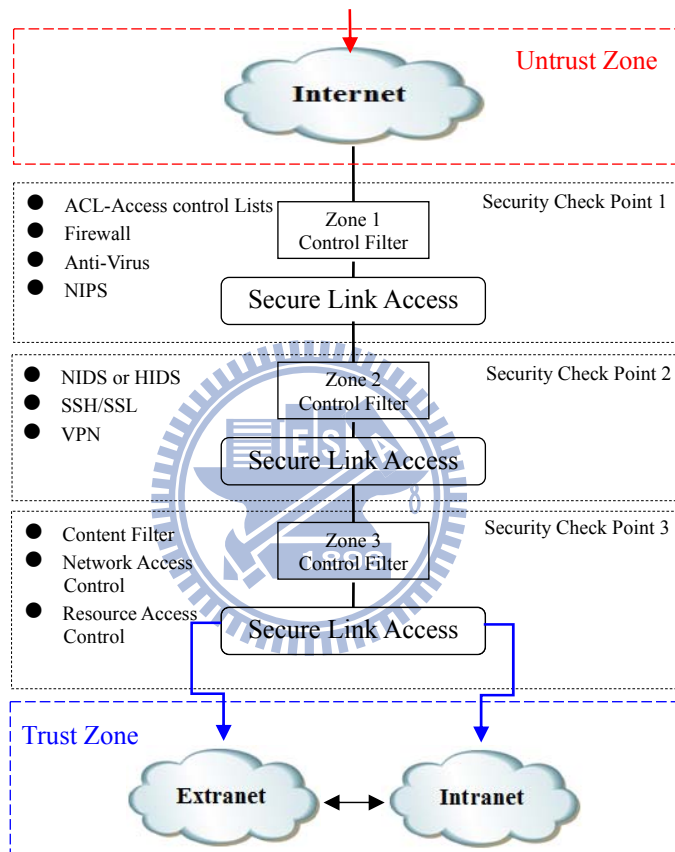


Figure 3.3: The concept of Defense in Depth Network.

### 3.3 VHSP based on Green IT Design Concept

The Green IT [16][17][18] design is a new concept and techniques for our energy efficiency improvements to reduce cost in time and dollars as well as avoid potential disruptions to existing IT infrastructure. Besides, the virtualization and facility efficiency management can each improve energy efficiency up to 60% over typical deployments [20].

Those are very important things of our IT environment at present. Reducing the number of physical equipment required in an organization could lower acquisition and maintenance costs.

This work integrates virtualization offering immense promise to improve energy efficiencies in our platform and could also improve VHSP reliability by isolating Operating System of VMs crashes. The VM is a single image file format, and it can do quick backup and recovery. The virtualization enables the Virtual Firewall, Virtual IPS, Virtual Honeynet, Virtual Database and VMI to run in a Virtual Operating System platform, allowing our design module to run multiple VMs on a single host OS of VHSP. This work applies the Honeypot and Honeynet technologies with a concept of all-in-one architecture design. VHSP and a design approach of the VHRM are presented in next Section.

The Green IT design concept is applied to all of the IT domains, and is able to move towards more energy-efficient computing and improves energy-efficiency. However, these technologies are also increase the complexity of IT architecture design and require significant up-front investment of time, equipment costs, high-level technical people and more computer resources.

### **3.4 Design of VHSP System and Network Architecture**

The virtual system and network architecture is shown in figure 3.4. This work adopts the concept of Defense in Depth Network with integration of the Firewall, IPS, Honeypot and Database as the basis of data collection, and through the Web Management Interface to conduct VHSP system monitor and management.

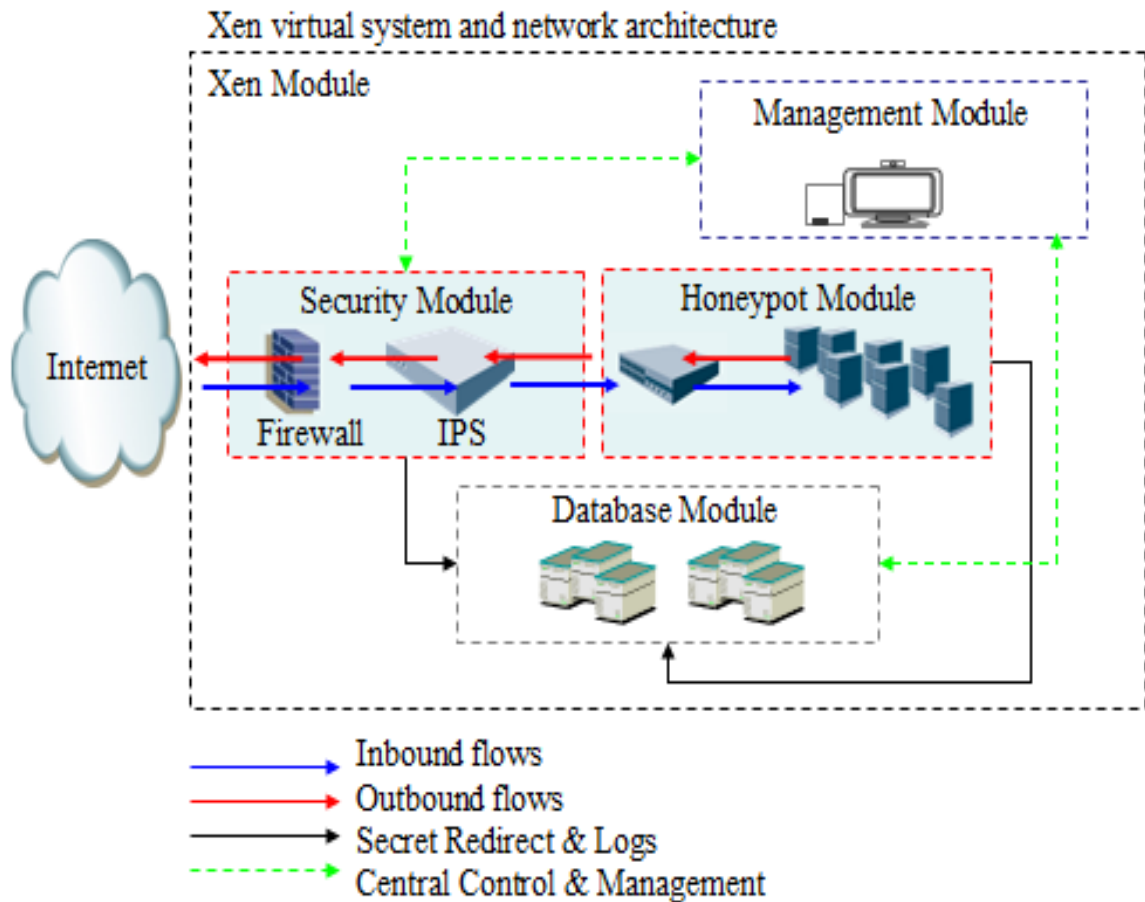


Figure 3.4: The virtual system and network architecture of VHSP operation based on XenLinux environment.

### 3.5 Implementation of VHSP Modules

The five basic modules are Xen Module, Security Module, Honeypot Module, Database Module, and Management Module in VHSP. The detail is shown in figure 3.5:

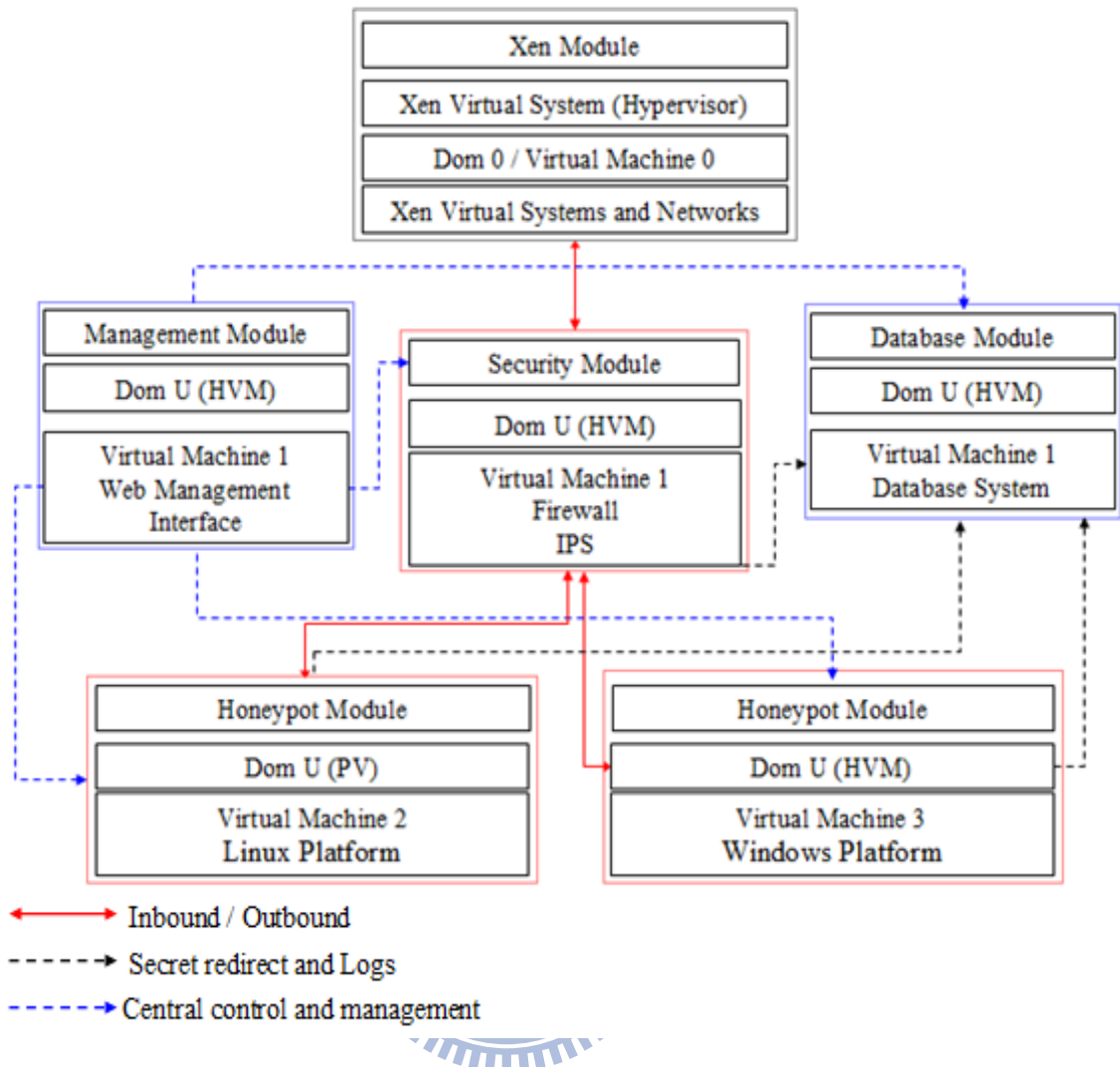


Figure 3.5: The virtual architecture of VHSP modules based on XenLinux environment.

### 3.5.1 Xen Module

The detail of Xen Module (XM) has been described in Section 3.3.1 and figure 3.1. This is a basic component of the VHSP system and network operation in this work.

### 3.5.2 Security Module

The Security Module (SM) includes two major components:

- **Virtual Firewall** : This part adopts iptables as the checkpoint of the network interface connection and establishes the regulation of packet inbound and outbound.

- **Virtual Intrusion Prevention System (Virtual IPS)** : This part adopts the technological integration of IDS Snort-inline as the components of intrusion detection prevention and deep packet inspection to this virtual platform.

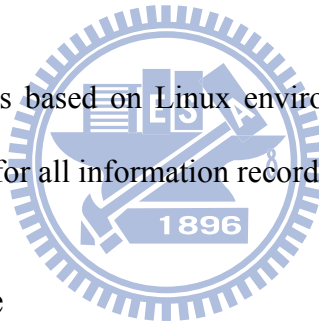
### 3.5.3 Honeypot Module

As for all Virtual Honeypots that are included in the Honeypot Module, the Honeypot adopts Rootkit based on Sebek technology.

The Sebek client can confidentially return the information and log of Honeypot back to Sebek server for the purpose of data collection. This work mainly takes Linux-based and Windows-based platforms as the components of HM.

### 3.5.4 Database Module

Database Module (DM) is based on Linux environment. This work adopts the MySQL Database as the storage center for all information record and logs.



### 3.5.5 Management Module

Management Module (MM) adopts HTML and Perl webpage program as the management and monitor components for the module and network events in the virtual system.

## 3.6 Backup and Recovery Strategies for VHSP

As shown in figure 3.3, by considering the future flexible deployment and application strategy, all of the VHSP modules can be dynamically added, removed and modified. Furthermore, under the VHSP, all of the Guest OS, regardless of PV Guest OS or HVM Guest OS, are the image format files and can be strategically made backup and recovery.

## 3.7 Design Approach of VHSP Virtual Networking

### 3.7.1 Design of VHRM based on HoneyNet Deploy Approach

The Network-Bridge Driver (NBD) [10][12][13][14][15] of Dom 0 is the backend of all outbound network communication for frontend of Dom U. Therefore, it shall go through NBD at first, and then directs connection to the Physical Ethernet Driver (PED). Therefore, all of packet flows of HM are parallel between Linux HoneyNet of Dom U and Windows HoneyNet of Dom U; thus, all packets that go inbound and outbound from HM will be processed by xenbr0 of NBD.

As shown in solid black-line path of figure 3.4, based on the Xen Networking, all of the HM is in the Dom U of the XenLinux environment. The default packet flows of the HM are outbound from Veth0 of Dom U going through xenbr0 of Network-Bridge Driver (NBD) in Dom 0. Next, packets flow are into the Physical Ethernet Driver eth0 and then into the internet. However; HM will ignore the SM and cannot monitor packet flows and store all event logs.

In order to solve the network flow and parallel issue, thus, this work modifies the Guest OS of Dom U to be parallel. This thesis proposes and designs the Virtual HoneyNet Redirect Mechanism (VHRM) for solving the SM and the HM of the virtual system bypassed problems.

Design approach is to direct packet flows from HM to Network Backend Driver (NBD) of the Dom 0 at first, and then from NBD redirect to Veth0 of SM of Dom U after being processed by IVN module, and then from Veth1 to Veth0 of SM to xenbr0 of NBD, finally, the packet flows from xenbr0 outbound to Physical Ethernet Driver eth0. As shown in dotted red-line path of figure 3.6.



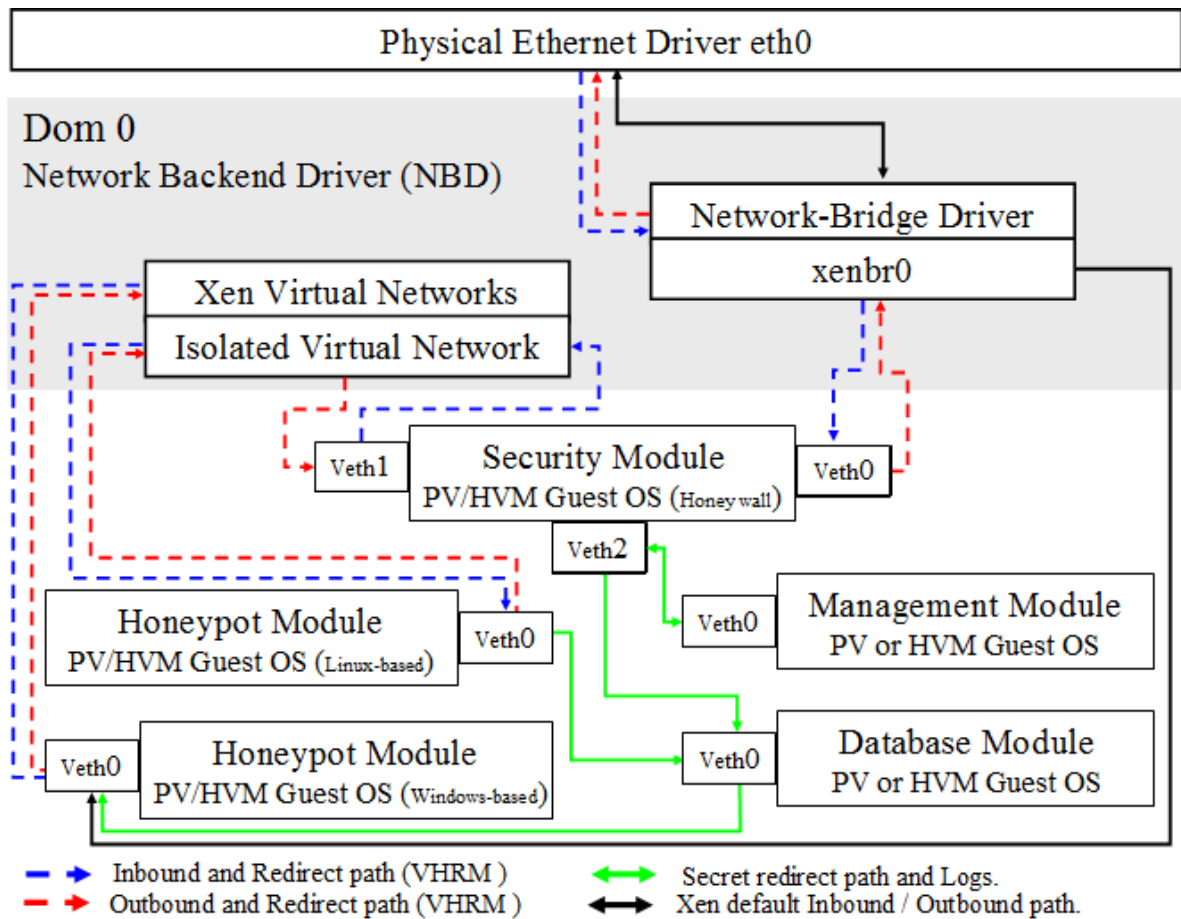


Figure 3.6: Design of Isolated Virtual Network module based on VHRM.

### 3.7.2 Implementation of Isolated Virtual Network Module

The shortest path will be considered as the design concept, The main advantage of VHRM adopts Layer 2 network protocol to connect when network packets flowing outbound VHSP. In addition, compared with this, it can be slowed and reduced more efficiency consumption for NBD in Dom0 than Layer 3. The efficiency flows in Layer 2 are more rapid than in Layer 3. Moreover, the virtual network interface of Security Module that includes the Veth0 and Veth1 also adopts the same approach of Layer 2 to conduct the network packet flows, and meanwhile uses the transparent mode to transfer packet flows. At the same time, through MM and DM to collect data, it logs and monitors all network events.

Outbound design flow is described as the solid red-line path, as shown in figure 3.7.

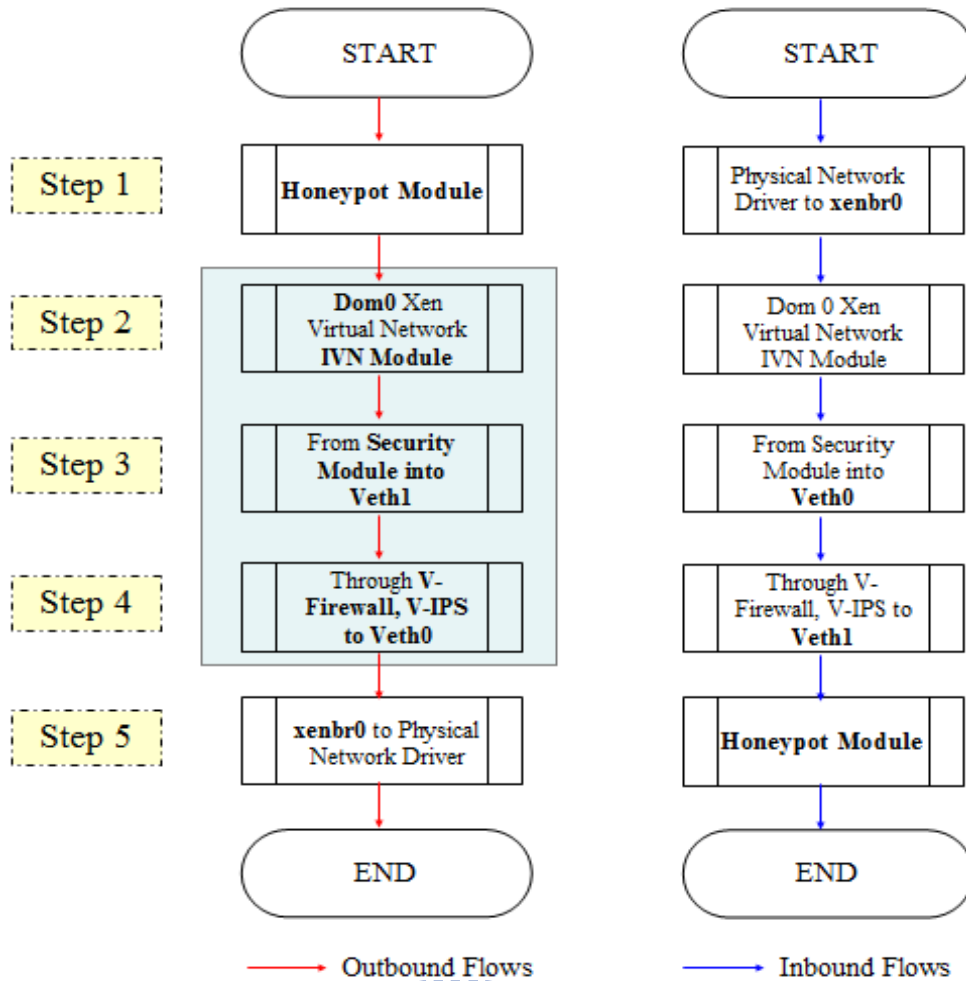


Figure 3.7: Flow chart of VHSP design approach.

- **Step 1: Honeypot module to IVN module.**

The first step is to make all packets inside the Honeypot Module to redirect toward the IVN module of Xen Virtual Networks in Dom 0.

- **Step 2: IVN module to Veth1 of Security Module.**

From IVN module it directs connection to the Veth1 of Security Module.

- **Step 3: Veth1 to Veth0.**

Step 3 uses layer 2 to lead packets from IVN module to the network interface Veth1; they go through Veth0 of Security Module in Dom U.

- **Step 4: Veth0 to xenbr0.**

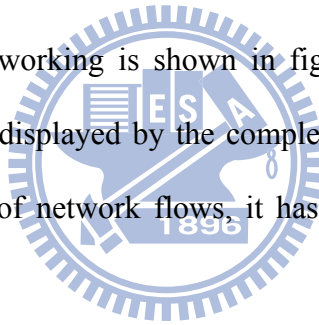
All data flows go through the network interface Veth1 of Security Module to Veth0 passing Virtual IPS and Virtual Firewall respectively. Next, packet flows will direct connect to xenbr0 of Network-Bridge Driver.

- **Step 5: xenbr0 to Physical Network Driver eth0.**

Data flows are also go through the bridge connection mode from Veth0 direct connect to the xenbr0, which is the xenbr0 of Network-Bridge Driver (NBD) component in Dom 0. Finally, in the XenLinux environment, this work use bridge mode to transfer all data to the eth0, a Physical Ethernet Driver (PED) via xenbr0. The design flow of inbound flows is reverse.

### **3.8 VHSP Virtual Networking**

Firstly, VHSP virtual networking is shown in figure 3.8. The part of VHSP network operating flow design will be displayed by the complete flow of the overall system module and network. As for the part of network flows, it has been described in Section 3.7.2 and figure 3.7 in this Chapter.



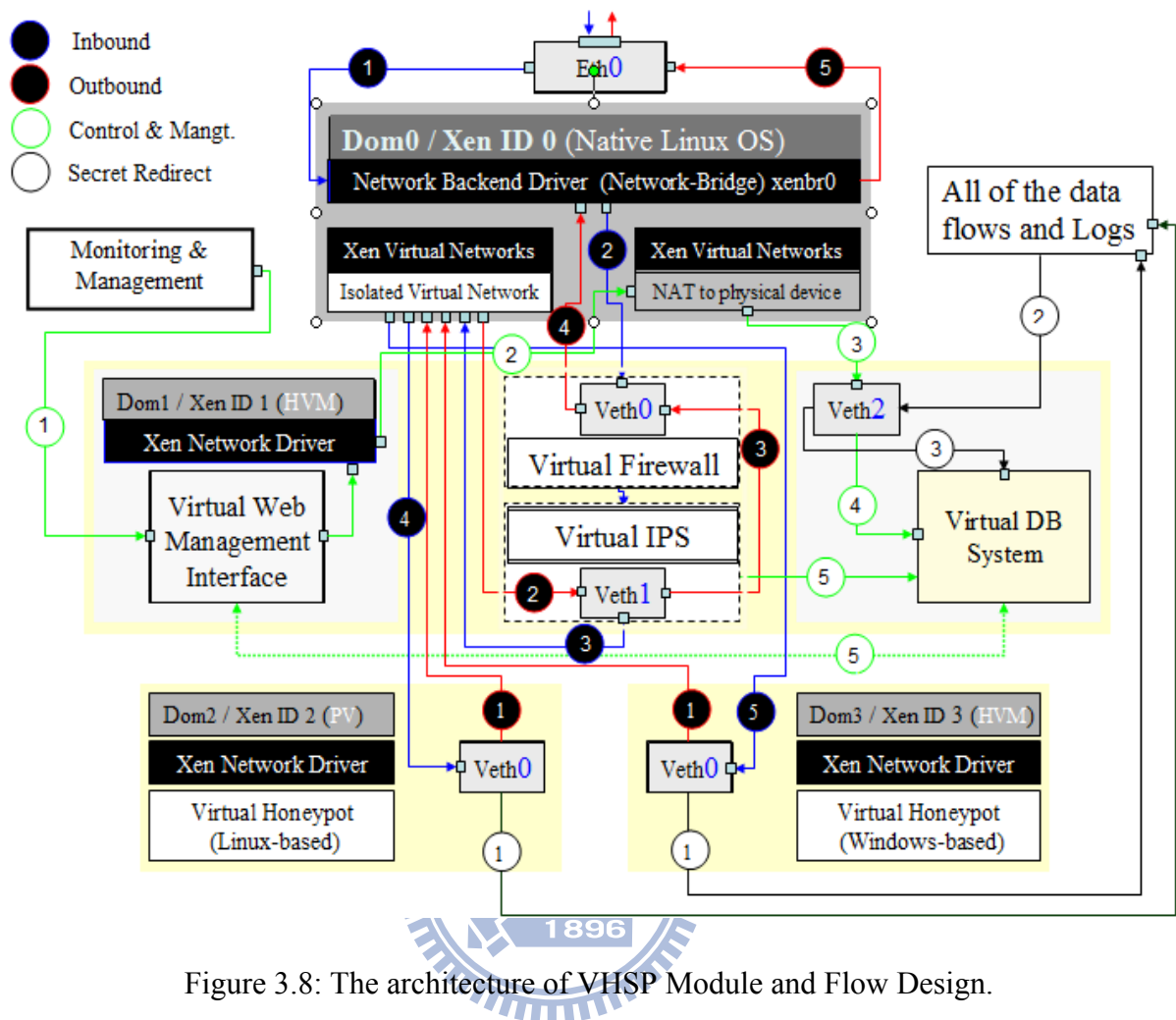


Figure 3.8: The architecture of VHSP Module and Flow Design.

Secondly, as for the part of using the WMI management interface to monitor and manage, researchers can externally access VHSP via the “NAT to physical device” in “Xen Virtual Networks”. Veth2 adopts the private IP logon to improve the security of connection. Via the external physical network to logon the Virtual WMI, and access to the Veth2 to manage the Virtual Firewall, Virtual IPS and Virtual Database systems. However, all connections are controlled and limited by Network Security Policy.

# Chapter 4 Results and Discussions

This experiment in this Chapter is used to verify the feasibility of VHSP architecture and virtual networking.

The simulation scenarios for VHSP validation is used Nessus [22] vulnerability scanning tool to verify our platform, network path and VHSP modules, and also check and review the virtual networking and event logs from WMI of VHSP. Finally, this work comparison result of Honeynet deploys and design methodology that includes the minimum Honeynet deployment requirement. The minimums number of hardware devices and comparison of different feature of Honeynet in the last section 4.3.

## 4.1 Simulation Scenario

In this experiment, the Nessus vulnerability scanning tool is used to verify our platform of VHSP which is designed in this section.

Nessus is a set of open-source software, and a well-functioned Nessus vulnerability scanner. Nessus uses the client / server model. It is a modular architecture consisting of centralized servers that conduct security holes scanning, and support remote client login for system management. The main function is to conduct the in-depth security test and system vulnerabilities analysis aiming at various system and network vulnerabilities for all of platforms. Nessus also can detect of missing system patches and Hot-fix packages, and do execution of the security tests scenarios in the various network environments. For example, it can simulate attacks to pinpoint vulnerabilities and detection of system security holes in local or remote operating system.

However, Nessus is used here to simulate various system vulnerabilities and online Hackers network attacks. In this scenario, the Linux virtual Honeypot uses CentOS and window 2000 virtual Honeypot. Now, there are currently 31558 different plugins used by Nessus, covering local and remote flaws. In this experiment the scan policy and simulation scenario use default setting.

As shown in the above figure 4.1, we take out 2 sections of Inbound and Outbound network logs from a large number of VHSP logs, which are indicated in figure 4.1. Thus, with the logs information we could clearly understand and determine that the overall network process and operation conforms to the design stated in figure 3.6.

```
Feb 24 16:05:51 vhpwh kernel: INBOUND TCP:  
IN=br0 OUT=br0 PHYSIN=eth0  
PHYSOUT=eth1 SRC=#.#.#.# DST=#.#.#.#  
LEN=48 TOS=0x00 PREC=0x00 TTL=128  
ID=41656 DF PROTO=TCP SPT=3992  
DPT=11214 WINDOW=65535 RES=0x00 SYN  
URGP=0  
  
Feb 24 16:06:47 vhpwh kernel: OUTBOUND UDP:  
IN=br0 OUT=br0 PHYSIN=eth1  
PHYSOUT=eth0 SRC=#.#.#.# DST=1#.#.#.#  
LEN=90 TOS=0x00 PREC=0x00 TTL=128  
ID=40835 PROTO=UDP SPT=137 DPT=137  
LEN=70
```

Figure 4.1: VHSP inbound and outbound network operation.

## 4.2 Simulation Validation

The audit vulnerability scan policy and well-known ports scanning policy are default setting in this scenario.

First of all, we connect to Web Management Interface (WMI) in VHSP via web browser, as shown in figure 4.2. The overall operating system and network with the logon record of abnormality or attack can be observed from the administration page. Next, on the upper-right corner, a heading lists as: “Created: Wed Jan 14 08:50:49 2009 Last Update: Tue Feb 24 16:24:33 2009”, and shows a simple real-time flow curve diagram.

In such a diagram, a right triangle can be seen. It means that there are more than 2000 Kbytes transferred at the time of “Feb 24 16:24:33 2009”. The most important feature is that the number of flows in “1 Hour” has hit the mark as high as 18212. At the same time, the administrator correspondingly reflects the rightmost “ids” on this column and that indicates 8 warning logs in 18212 logons.

Next, a column titled as the “Top 10 Honeypots” on the lower-left corner of the WMI management screen, shows 16 events of sending 19779 session IDS for one IP connection within a short period of time (most of them are the attack connections).

In figure 4.2, there are other related basic information for network statuses existed respectively, including the information of Host, connections, IDS events, etc. It can implement the real-time analysis on the abnormal flow status and warning.

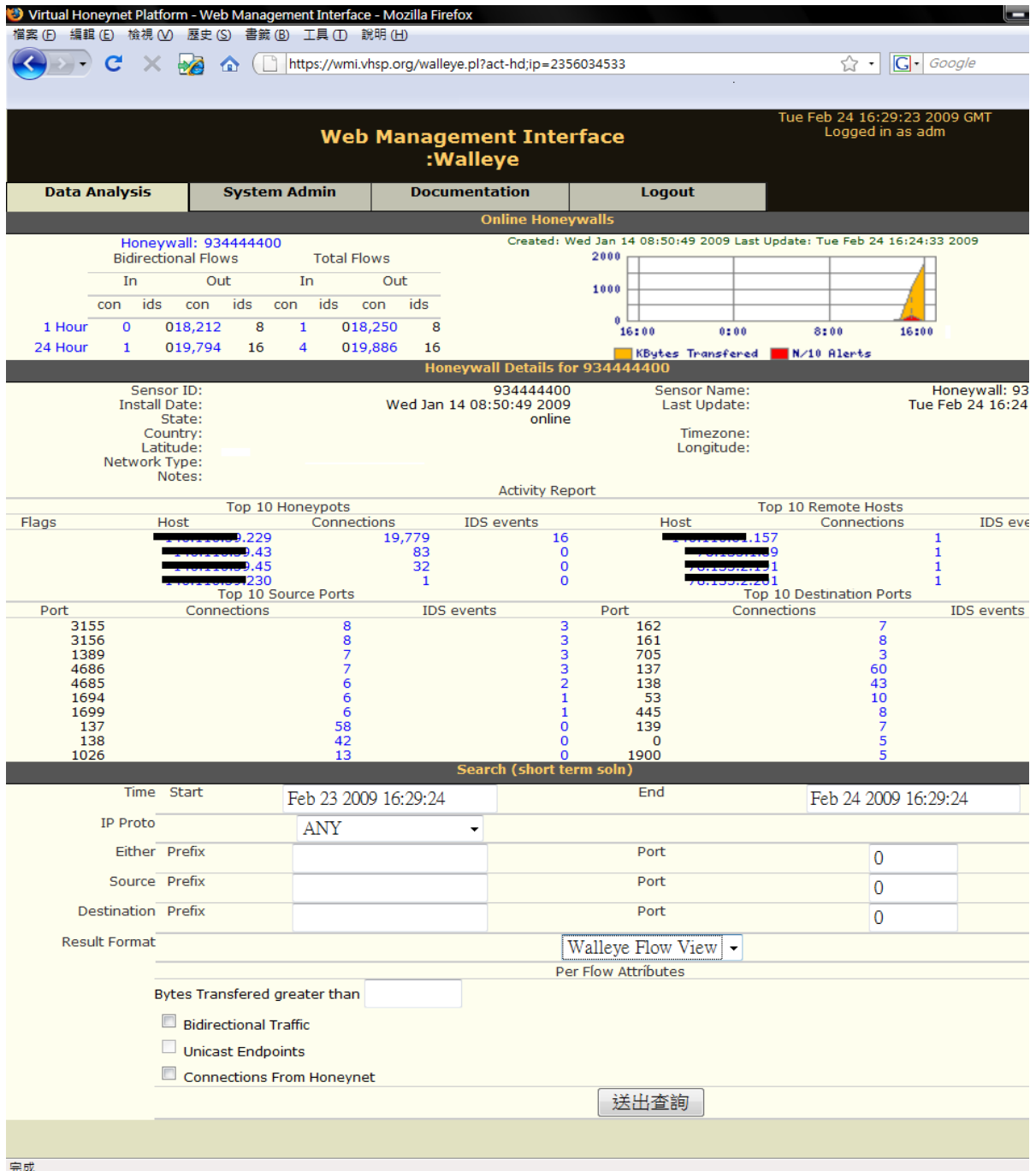


Figure 4.2: The Web Management Interface status monitor which is based on Virtual Machine Monitor environment.

As displayed in figure 4.3, a researcher can make use of the Virtual Machine Manager for Xen control software to monitor the allocation status of Real-Time (RT) resources for Xen module and the efficiency analysis for each module's operation.



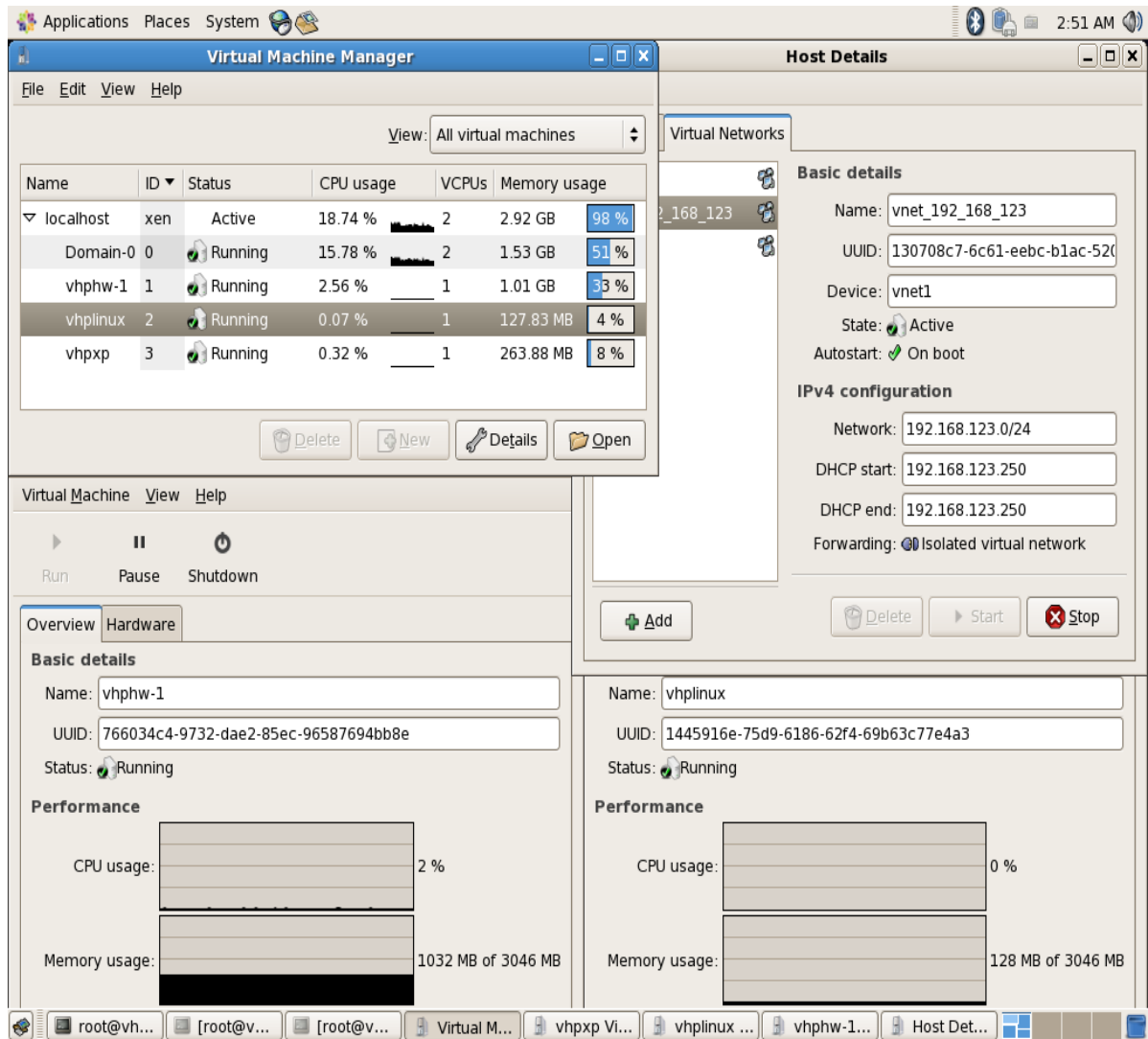


Figure 4.3: Virtual Machine Monitor based on VHSP Monitor interface.

### 4.3 Comparison Results of Honeynet Design Methodology

In this section, we give a brief introduction for comparison on Honeynet deployment requirement, Honeynet hardware cost, devices, and different feature of Honeynet to describe the Honeynet design methodology. Section 4.3.1 compares the minimum Honeynet deployment requirement. Section 4.3.2 compares the minimum number of hardware devices for Honeynet deployment. Comparison with different features of Honeynet, including equipment cost, resources utilization, flexibility, re-configurability, deploy strategies, and the system performance, is described in the last section 4.3.3.

### 4.3.1 Comparison of the Minimum Honeynet Deployment Requirement

This Section compares from Table 4.1, the minimums Honeynet deployment requirement and discovers that integrated VHSP of virtualization technology will be able to lower consumption and usage rate of both hardware resources and physical equipment.

Taking hardware cost into consideration, we compares the minimum Honeynet deployment requirement in Table 4.1. The conventional Honeynet requires at least six entities (including an optional equipments), while the proposed VHSP of ours needs a maximum of only two necessary entities equipment. (This is including an optional device.)

Table 4.1: Minimums Honeynet deployment requirement.

Minimum Number of Hosts	Feature	<i>Conventional</i>	<i>Modified</i>	<i>Hybrid</i>	<i>This Work (VHSP)</i>
n >= Host >= 1	(1) WH	1	1		
	(2) LH	1	1		
	(3) FM	1			
	(4) IPSM	1			
	(5) MI	1	1	1	
	(6) HiFI		1	1	
	(7) VMiH			1	
	(8) OOM	1	1	1	1
	(9) VHSP				1
<b>Total</b>		<b>6</b>	<b>5</b>	<b>4</b>	<b>2</b>

- (1) WH = Windows-based Honeypot
- (2) LH = Linux-based Honeypot
- (3) FM = Host-based Firewall or Network-based Firewall
- (4) IPSM = Host-based IPS or Network-based IPS
- (5) MI = Management Interface
- (6) HiFI = Honeywall integration of the Firewall & IPS
- (7) VMiH = Virtual Machine integration of the Honeypot
- (8) OOM = Other Option Module
- (9) VHSP = Virtual Honeynet Security Platform

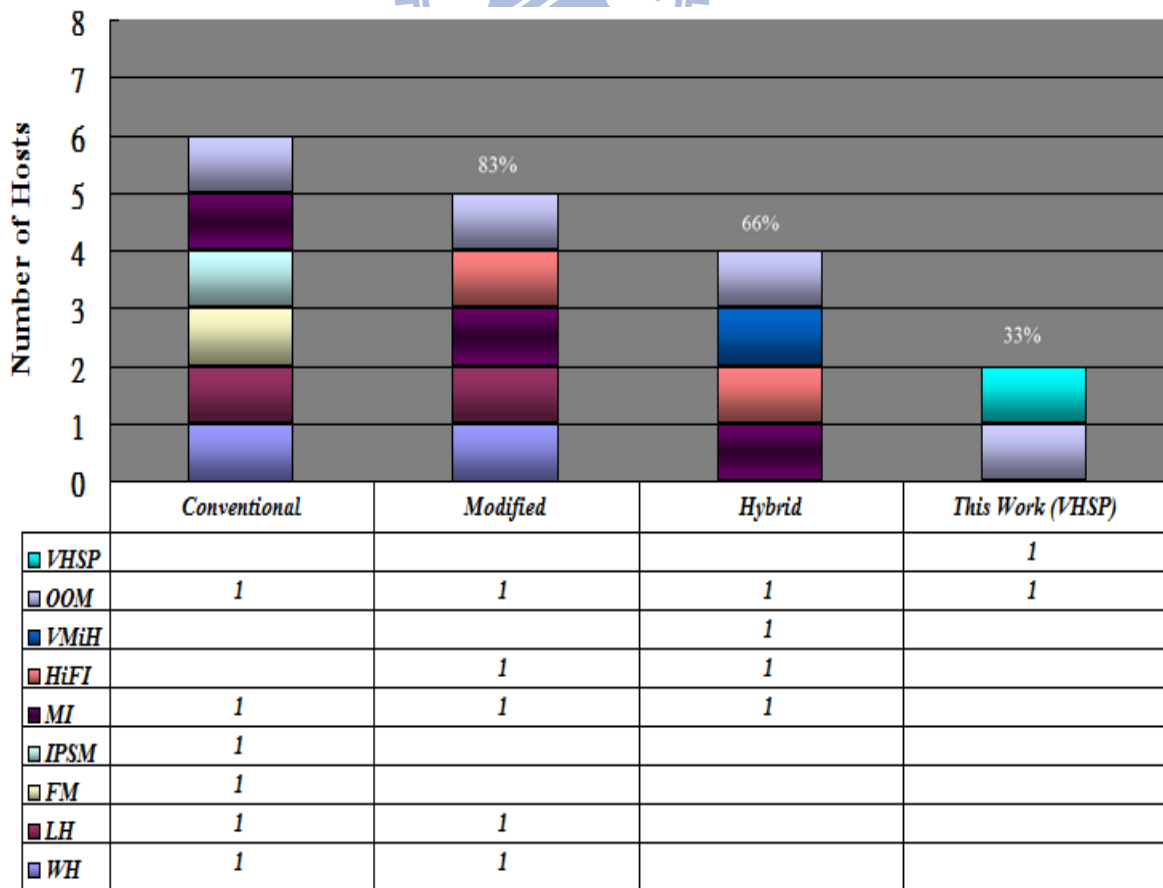
### 4.3.2 Comparison of the Minimums Number of Hardware Devices

As shown in Table 4.2, this part compares the entities required for the use of hardware quantities and resources consumption of each different deployment strategies, as stated above; the VHSP can greatly reduce hardware costs and waste of resources.

Supposing that a researcher uses the conventional deployment as a benchmark to quantify; in this work it only needs 33% of hardware requirements, saving 66%, only 1/3 of the conventional one.

In addition, for consideration on performance, one unit of entities equipment is rather difficult to compare with many units of entities equipment. But if researchers focus for their resource utilization, flexibility in deployment strategies, re-configurability, as well as the most important of the above-mentioned hardware cost, VHSP should be a better way.

Table 4.2: Minimums number of hardware devices of Honeynet comparison results.



### 4.3.3 Comparison of Different Feature of Honeynet

In this Section, we give a brief introduction for comparison of hardware and equipment cost, flexibility and deploy strategies, hardware resource utilization, systems / networking performance and re-configurability, as shown in Table 4.3.

A new virtual Honeynet architecture is developed, and the VHSP is implemented. Comparison shown in Table 4.3 and discovers that integrated VHSP virtualization technology will be able to lower consumption and usage rate of both hardware resources and physical equipment, and be able to increase hardware resource utilization, flexibility & deploy strategies, and Honeynet re-configurability in VHSP.

Table 4.3: Comparison of different features of Honeynet.

<b>Feature / Honeynet Type</b>	<i>Conventional</i>	<i>Modified</i>	<i>Hybrid</i>	<i>This Work (VHSP)</i>
<b>Hardware &amp; Equipment Cost</b>	High	High	Medium	Low
<b>Flexibility &amp; Deploy Strategy</b>	Low	Low	Medium	High
<b>Hardware Resource Utilization</b>	Low	Low	Medium	High
<b>Networking &amp; System Performance</b>	High	High	Medium	Low
<b>Re-configurability</b>	Low	Low	High	High

Researchers know that the international HoneyNet Project [5] against the proposed virtual HoneyNet [14] possible method and approach [3] are using VMware [7] [9] and other VT for the HoneyNet system and the conventional HoneyNet architecture with network design, while our proposed design of virtual HoneyNet adopted Ian Pratt's [12] emphasis Paravirtualization that it can provide the similar speed to the physical operating system.

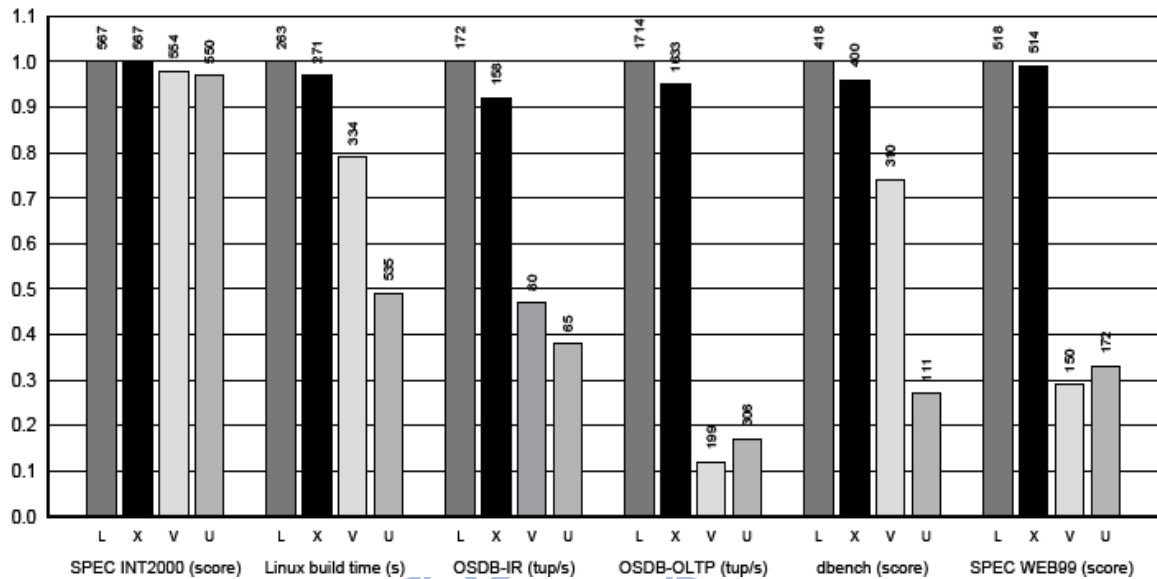


Figure 4.4: Comparison of the performance of Native Linux (L), XenLinux (X), VMware workstation 3.2 (V) and User-Mode Linux (U).

As shown in figure 4.4, The Native Linux (L) can provide similar speed to real Operating System [12]. Although a researcher needs to modify the Kernel of PV Guest OS in VHSP, but we take better performance and meantime consideration for hardware and software resources, our research proposes a virtual HoneyNet of the All-in-One VHSP platform, using paravirtualization based on XenLinux virtual environment. And it will be the best solution for HoneyNet architecture design in the near future.

# Chapter 5 Conclusion and Future Works

## 5.1 Conclusion

The main purpose of this thesis is to improve the design and concept of conventional Honeynet architectures. This work proposes the VHSP and designs the VHRM for solving SM and HM of the virtual system bypassed problems. Based on our approach, the VHSP modules can choose an appropriate method in the Virtual Machine-based scheme for various design conditions. Compared with the conventional deployment as a benchmark to quantify, in this work it only needs about 33% of hardware requirements, saving 66%. Therefore, this work has contributed a better operating strategy of software and hardware resources, from the perspective of technological development and innovation. Our proposed VHSP is not only a cost-effective design but also a more flexible security research platform. Moreover, the VHSP also conforms to the latest concept of Green IT design. In the near future, the VHSP can support and include those approaches and methodology to the Taiwan Honeynet Project.

## 5.2 Future Works

- The VMI of the Honeynet and VMM needs to be technologically integrated.
- The first security issue for Taiwan Honeynet Project is to collect malware samples.
- Distributed VHSP, (DVHSP) is a good direction for research and design in Taiwan's Research Network (Taiwan Honeynet Project).
- In order to enhance the VHRM performance, a new mechanism called Physical Network Interface Honey-pot Redirect Mechanism (PNIHRM) that is a good direction to enhance and support packet throughput performance.

- A new security research architecture is the HoneyNet Central Management System (HCMS). All of Distributed VHSP components that include system resources and all processes will be Real-Time Monitoring (RTM) and Management.



# References

- [1] Bhatia, J.S., Sehgal, R., Bhushan, B., and Kaur, H., “A case study on host based data analysis & cyber criminal profiling in Honeynets”, in *Proceedings of the IEEE COMSNETS International Communication Systems and Networks and Workshops*, pp. 11-21, 2009.
- [2] Chamales, G., “The Honeywall CD-ROM”, *IEEE Journal of Security & Privacy*, vol. 2, pp. 77-79, Mar. 2004.
- [3] Honeynet Project, Know Your Enemy: Learning about Security Threats, Second Edition, *Addison-Wesley Professional Publishers*, May 27, 2004.
- [4] Raynal, F., Berthier, Y., Biondi, P., and Kaminsky, D., “Honeypot forensics part 1: analyzing the network”, *IEEE Journal of Security & Privacy*, vol. 2, pp. 72-78, Aug. 2004.
- [5] Spitzner, L., “The Honeynet Project: trapping the hackers”, *IEEE Journal of Security & Privacy*, vol. 1, pp. 15-23, Mar. 2003.
- [6] Raynal, F., Berthier, Y., Biondi, P., and Kaminsky, D., “Honeypot forensics part 2: analyzing the compromised host”, *IEEE Journal of Security & Privacy*, vol. 2, pp. 77-80, Oct. 2004.
- [7] Nance, K., Bishop, M., and Hay, B., “Virtual Machine Introspection: Observation or Interference”, *IEEE Journal of Security & Privacy*, vol. 6, pp. 32 - 37, Oct. 2008.
- [8] Uhlig, R., Neiger, G., Rodgers, A.L., Martins, F.C.M., Anderson, A.V., Bennett, S.M., Kagi, A., Leung, F.H., and Smith, L. “Intel virtualization technology”, *IEEE Journal of Computer*, vol. 38, pp. 48-56, May. 2005.
- [9] Xu, X., and Zhou, F., “Quantifying Performance Properties of Virtual Machine”, in *Proceedings of the IEEE ISISE International Symposium on Information Science and Engineering*, pp. 24-28, 2008.
- [10] Zhang, X., and Dong, Y., “Optimizing Xen VMM Based on Intel Virtualization Technology”, in *Proceedings of the IEEE ICICSE International Conference on Internet Computing in Science and Engineering*, pp. 367-374, 2008.
- [11] Chen, W., Lu, H., Shen, L., Wang, Z., Xiao, N., and Chen D., “ A Novel Hardware Assisted Full Virtualization Technique“, in *Proceedings of the IEEE ICYCS International Conference for Young Computer Scientists*, 2008, pp. 1292-1297, 2008.
- [12] Barham, P., Dragovic, B., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A., “Xen and the art of virtualization”, in *Proceedings of the ACM symposium on Operating systems principles*, pp. 164-177, 2003.



- [13] Govindan, S., Choi, J.G., Nath, A.R., Das, A., Urgaonkar, B., and Sivasubramaniam, A., “Xen and Co.: Communication-Aware CPU Management in Consolidated Xen-Based Hosting Platforms”, *IEEE Trans. on Computers*, vol. 58, No. 8, pp. 1111- 1125, 2009.
- [14] Zanolamy, W., Zakaria, A., Rohaidah, S., and Norazah A., “Deploying virtual honeypots on virtual machine monitor”, in *Proceedings of the IEEE ITSim International Symposium on Information Technology*, pp.1-5, 2008.
- [15] Xianghua, X., and Peipei, S., “Performance Evaluation of the CPU Scheduler in XEN”, in *Proceedings of the IEEE ISISE International Symposium on Information Science and Engineering*, pp.68-72, 2008.
- [16] Cameron, K.W., “The Road to Greener IT Pastures”, *IEEE Journal of Computer*, vol. 42, pp. 87-89, May. 2009.
- [17] Murugesan, S., “Harnessing Green IT: Principles and Practices”, *IEEE Journal of IT Professional*, vol. 10, pp. 24-33, Jan. 2008.
- [18] Ruth, S., “Green IT More Than a Three Percent Solution”, *IEEE Journal of Internet Computing*, vol. 13, pp. 74 – 78, Aug. 2009.
- [19] Chang, C.-H., “A Low-Cost Green IT Design and Application of VHSP based on Virtualization Technology”, in *Proceedings of the IEEE SecureCom'09 International Symposium on Secure Computing*, Aug. 2009, Vancouver, Canada.
- [20] Chang, C.-H., and Hsiao, T.-C., “A Low-Cost Green IT Concept Design of VHSP based on Virtualization Technology”, Accepted and to appear in *2009 IEEE SMC International Conference on Systems, Man, and Cybernetics*, Oct. 2009, San Antonio, Texas, USA.
- [21] Huang, N.-F., Kao, C.-H., Hun, H.-W., and Lin, C.-L., “Apply data mining to defense-in-depth network security system“, in *Proceedings of the IEEE AINA International Conference on Advanced Information Networking and Applications*, vol. 1, pp. 159-162, 2005.
- [22] Yoshimoto, M., Bista, B.B., and Takata, T., “Development of security scanner with high portability and usability”, in *Proceedings of the IEEE AINA International Conference on Advanced Information Networking and Applications*, vol. 2, pp. 407-410, 2005.
- [23] Koller, R., Rangaswami, R., Marrero, Smith, G., Barsilai, M., Necula, S., Sadjadi, S.M., Tao Li, and Merrill, K., “Anatomy of Real-Time Intrusion Prevention System”, in *Proceedings of the IEEE ICAC International Conference on Autonomic Computing*, pp. 151-160, 2008.
- [24] Kenneth J.D., and Cheriton D.R., “Borrowed-Virtual-Time (BVT) scheduling”, in *Proceedings of the ACM symposium on Operating systems principles*, pp. 261-276, 1999.

# Vita

Chih-Hung Chang is currently an Information Security Engineer at National Center for High-performance computing (NCHC), Taiwan. He is joining the Information Security Team. He obtained B.S. in Information Management from the College of Computer Science & Informatics, and obtained B.S. in Foreign Languages and Literature, at Chung Hua University. He is currently a Master degree student at College of Computer Science, National Chiao Tung University (NCTU), Taiwan. His research interests include Network Security, Virtualization, and Cloud Computing.

