MARA UNIVERSITY OF TECHNOLOGY

HARDENING OF THE SMART EXPERIMENTAL LEARNING

AHMAD ABDUL AZIM BIN MANSOR

BACHELOR OF SCIENCE (Hons.)
IN DATA COMMUNICATION AND NETWORKING

FACULTY OF
INFORMATION TECHNOLOGY AND QUANTITATIVE SCIENCE

NOVEMBER 2007

HARDENING OF THE SMART EXPERIMENTAL LEARNING

By

AHMAD ABDUL AZIM BIN MANSOR

(2005730677)

A project paper submitted to

FACULTY OF INFORMATION TECHNOLOGY AND QUANTITATIVE SCIENCES

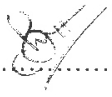
MARA UNIVERSITY OF TECHNOLOGY

In partial fulfillment of requirement for the

BACHELOR OF SCIENCE (Hons.) IN DATA COMMUNICATION AND NETWORKING

Major Area: Security

Approved by the Examining Committee:



.....

Mr. Mohd Ali Bin Mohd Isa

Project Supervisor

MARA UNIVERSITY OF TECHNOLOGY, MALAYSIA

NOVEMBER 2007

CERTIFICATION OF ORIGINALITY

I declare that the work in this thesis was carried out in accordance with the regulations of MARA University of Technology. It is original and is the result of my own work, unless otherwise indicated or acknowledged as referenced work. This topic has not been submitted to any other academic institution or non-academic institution for any other degree of qualification.

In the event that my thesis be found to violate the conditions mentioned above, I voluntarily waive the right of conferment of my degree and agree be subjected to the disciplinary rules and regulations of MARA University of Technology.

……

AHMAD ABDUL AZIM BIN MANSOR

2005730677

ACKNOWLEDGEMENT

In the name of ALLAH, the most gracious and the most merciful for without Him, it is impossible for me to finish up my project and delivering this report on time.

First and foremost, I would like to express my deepest gratitude to my academic supervisor, Mr. Mohd. Ali Bin Mohd. Isa for his cooperation and suggestions in providing me with vital information, materials and expertise as well as useful insights on common issues relating to my project. Only Allah could repay your deeds.

My sincere gratitude also goes to Faculty of Information Technology and Quantitative Science lecturer, Mr. Adzhar Bin Abd. Kadir for the guidance, knowledge, encouragement and advice.

Special thanks to all of my friends who has helped me through thick and thin, for giving me their full support, a lot of valuable information, guidance, motivation and opportunity as well as pleasant relationship that will not be forgotten.

Last but not least, I would like to thank my family for giving me their support; cooperation and everything that can make me feel comfortable in order for me to finish my project.

ABSTRACT

Across the world, there exists a lack of computer security components in many computer science curriculums. For those programs that do have such components in computer security, a common difficulty is to integrate "real-world" labs into the courses, in order to provide hands-on experiences to the learners. Due to concerns for security breaches and network hacking, system administrators are reluctant to allow computer security labs involving network sniffing, virus scripting, etc. to be deployed in the campus network. Without hands-on, real-world projects, it is difficult for the learners to integrate the acquired security theories and knowledge with up-to-date security technologies and practices. For this reason, the purpose of this project paper is to develop and implement a virtual lab module for undergraduate computer security course covering the basics of network security. This project is called Smart Experimental Learning. This virtual lab can be set up for a relatively small investment and allows students to learn attacker techniques and how to defend against them. Little specialized knowledge is required to set up the lab hardware and configure the machines. The hardening process was also being done using various software available. Basically, the results show that the hardening process can be done even the operating systems were installed virtually and students can take this advantage to learn how to harden a system in an environment whereby the results does not effect the real network on the campus. It is hoped that by designing and implementing this virtual lab module successfully, students can learn and experience what we called computer security.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

For the past decade, partly due to the widespread use of the Internet, computer security has become a top issue in industry, academic and government. The demand for well-trained security professionals has grown dramatically. The integration of security into computing curricula, however, has not kept up with this demand. A related problem is, most existing computing courses lack security components. The problems are even more serious for universities such as Universiti Teknologi MARA (UiTM) where resources tend to be limited.

For those programs that do have courses in computer security, a common difficulty is to integrate "real-world" labs into the courses, in order to provide hands-on experiences to the learners. Due to concerns for security breaches and network hacking, system administrators are reluctant to allow computer security labs to be deployed in the campus network. Unless deployed in an isolated computer lab, projects involving hacking techniques, such as network sniffing and virus scripting, are generally

prohibited in the campus network. Without hands-on, real-world projects, it is difficult for the learners to integrate the theories and knowledge acquired in the classroom with up-to-date security technologies and practices.

In order to solve this problem, Smart Experimental Learning will be designed and implemented to help students especially who are taking computer security courses in UiTM to experience and explore the natural world of security. This lab module will be designed for students to learn on how to harden an operating system. Smart Experimental Learning model helps assure an optimum learning experience for the student. Based on VMware, virtual machine technology, users are able to perform the required hardening on the guest operating system. By doing this lab module, students should be able to experience the real world of computer security.

1.2 PROBLEM STATEMENT

Teaching security courses is proved to be difficult because students cannot experience the real environment in order to protect the campus network. The cost to setup a real laboratory for security purposes is also proved to be costly. Student cannot experience the real environment if they want to defend a system from being attacked. Student also has less practical experience on the network and web security.

The Smart Experimental Learning will provides hands-on experience for students to study cutting-edge computer security technologies, and serves as a test bed for projects which are otherwise impossible to implement in general-purpose labs. By doing the hardening, student can be exposed to the real networking environment whereby this will help them when they work in the future.

1.3 RESEARCH QUESTION

The research questions for this project are:

i) What are the tools that will be used in order to develop Smart Experimental Learning?

ii) How to harden the Smart Experimental Learning?

1.4 OBJECTIVES

The objectives of this research are:

i) To develop Smart Experimental Learning using VMware.

ii) To harden the Smart Experimental Learning using various tools available.

1.5 SCOPE OF RESEARCH

The scopes of this research are:

i) Smart Experimental Learning will be developed using VMware virtual machine. It will also use various types of operating system and software in order to do the required hardening.

ii) This research will focus more on how to harden the system.

iii) Smart Experimental Learning will be targeted to be used by students who are taking security courses in UiTM.

1.6 SIGNIFICANCE OF RESEARCH

Smart Experimental Learning will be developed in order to give the students real environment and let them experience it all. Students can work in real environment where they can learn on how to defend a host or network from being attacked. This can certainly open their mind and teach them the real security life. It will permit students to safely explore the usage, effects and defense against various attacks.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

Computer network security is generally taught with two varying approaches. One approach, which is typical of graduate level courses, has a theoretical focus. However, within industry, computer security, particular network security training, takes a much more applied focus. Here, the same tools used by attackers to illegally scan, probe, and gain access to computer networks are taught to and ultimately used by the computer security professional. There are two aspects to this latter approach. First, the attacker tools can be used by the defender to probe the network for security vulnerabilities which having been discovered can then be eliminated. Secondly, an understanding of how the attacker tools work is vital to understanding how to defend a network against them. Such a lab can be set up for a minimal expenditure of resources. Counter Hack by Ed Skoudis is an excellent textbook for this approach.

2.2 HISTORY

Computer networking has become so ubiquitous that every computer system of any importance is networked to others. The Internet Domain Survey (see www.nw.com) reports that there are 162,128,493 hosts in the DNS, as of July 2002. The older security problems of insider breaches of security are now compounded by attacks carried out remotely through the network. This is fueling the growth of what are known as "network firewalls" and increased deployment of security tools. Graduates from typical degree programs in Computer Science and Engineering are familiar with stories of security breaches as reported in the media, but do not have a technical grasp of the security issues. There is an urgent need for training the students in computer and network security.

2.3 CHALLENGES IN DESIGNING SECURITY LAB

The use of specialized computer security labs for teaching computer security related courses has long been advocated by computer science educators. There exist many challenges in setting up realistic computer security laboratories and assignments. Some of the challenges are listed below:

i) Need to protect campus networks

Due to the widespread virus attacks and hacking incidents, the system administrator is justifiably concerned with the health of the networks that he/she is responsible for. To avoid security breaches caused by security and hacking programs running in the computer security lab, most universities isolate their computer security labs from the rest of the campus network.

ii) Need to access the Internet

Students and faculty who use the computer security lab do have the need to connect to the Internet to, for example, get information, download security software, etc. It is inconvenient for a whole class of students to share one or two workstations in order to access the Internet.

iii) Resource needs for students to develop their solutions

Students may not always have their own computers, and their own computers may not mimic the environment of the proposed assignments. The computer security lab needs to provide needed resources for students to develop their assignment solutions. Furthermore, to provide consistency in assignment evaluations, it is desirable for students to work in identical environments, even when they can set up similar environments in their own computers.

iv) Easy and secure access to the resources

The resources available in the lab should be easily accessible. Students may choose to use the lab either locally or remotely (e.g., from home or from their work place). Remote access to the lab, however, tends to prompt security concerns, mainly due to the widespread hacking incidents launched over the Internet.

v) Incorporation of latest technological development

Computer technologies are notorious for their fast-paced change. New technologies are constantly created. To accommodate the latest technological development, such as wireless networking, secure remote access, etc., it is important that the design of the security lab be scalable, in the sense that additional test beds or components may be easily added to the existing network. A security lab has to take extensibility and reconfigurability into account to accommodate new technologies.

2.4 VIRTUAL MACHINE

One of the earliest Virtual Machine (VM) systems was CP-40, developed for an IBM System/360 Model 40. A VM system was defined as a computing system in which the instructions issued by a program may be different from those actually executed by the hardware to perform a given task. More generally, a VM can be categorized as a "software abstraction with the looks of a computer system's hardware (real machine)". Advantages of VMs including:

- Concurrently executing multiple operating systems supporting different users
- Developing software for one machine on a different machine
- Insulating one software environment executing on a VM from failures in others

VM systems were also purported to enhance computer system security if the programs of independent and possibly malicious users are to coexist on the same computer system. It was recognized that a combined virtual machine monitor/operating system (VMM/OS) approach to information system isolation provides substantially better software security than a conventional multiprogramming operating system approach. In addition, VM systems offered a very suitable basis for system development given the independence of VM's and their support for debugging and monitoring of systems activities.

Recently, the use of VM concepts has experienced a rebirth in computing. VM software such as Xen and VMware introduces an abstraction layer between operating systems and computer hardware which enables multiple virtual servers to share a common pool of hardware resources. Other VM software such as VMware Workstation creates virtual hardware environments within a host operating system that enables other operating systems and their applications to execute within encapsulated virtual machines. This form of VM software is commonly used to test system and application

software under multiple operating systems or operating system versions on a single physical machine. Modern VM software such as VMware provides the benefits described earlier including concurrently executing multiple operating systems in separate VMs and isolating each VM from bugs and malicious code in other VMs. Additional benefits include the ability to configure virtual networks. Early VM's consumed about 10-15% of the underlying computing resources. Modern VMs consume just a few percentage points of the physical machine resources.

Prior to the return of VM's, the academic computing community was forced to fund dedicated computer labs in order to each advanced classes covering topics such as system administration, network design, and information security. However, such labs are expensive and often feasible only with external funding. The reemergence of VM software enables educational institutions to configure shared, multi-course computing labs in which students in advanced courses can undertake system-level projects. VM features such as isolation, compatibility, and encapsulation allow an instructor to build virtual network topologies encompassing multiple, independent operating systems and networks.

2.5 TECHNOLOGY

VMware Workstation is a virtual machine software suite for x86 and x86-64 computers from VMware, a division of EMC Corporation. This software suite allows users to set up multiple x86 and x86-64 virtual computers and to use one or more of these virtual machines simultaneously with the hosting operating system. Each virtual machine instance can execute its own guest operating system, such as Windows, Linux, BSD variants, or others. In simple terms, VMware Workstation allows one physical machine to run multiple operating systems simultaneously. Other VMware products help manage or migrate VMware virtual machines across multiple host-machines.

Besides bridging to existing host network adapters, CD-ROM devices, hard-disk drives, and USB devices, VMware Workstation also provides the ability to simulate some hardware. For example, it can mount an ISO file as a CD-ROM, and .vmdk files as hard disks; and can configure its network adapter driver to use network address translation (NAT) through the host-machine rather than bridging through it (which would require an IP address for each guest-machine on the host network).

VMware Workstation also allows the testing of LiveCDs without first burning them onto physical discs or rebooting the computer. One can also take multiple successive snapshots of an operating system running under VMware Workstation. Each snapshot allows you to roll back the virtual machine to the saved status at any time. The ability to use multiple snapshots makes VMware Workstation useful as a tool for sales-people demonstrating complex software products, and for developers setting up virtual development-environments and virtual test-environments. VMware Workstation includes the ability to designate multiple virtual machines as a team which administrators can then power on and off, suspend, and resume as a single object — making it particularly useful for testing client-server environments.

Keeping the hardware and virtual machine terminology unambiguous is essential to discussions of virtualization. VMware refers to the physical hardware computer as the host machine, and identifies the operating system (or virtual appliance) running inside a virtual machine as the guest. This is common for personal and enterprise-wide VMware software. Like an emulator, VMware software provides a completely virtualized set of hardware to the guest operating system. VMware software virtualizes the hardware for video adapter, network adapter, and hard disk adapters. The host provides pass-through drivers for guest USB, serial, and parallel devices.

Thus, VMware virtual machines are highly portable between computers, because every host looks nearly identical to the guest. In practice, a virtual machine guest can pause operation, be moved or copied to another physical computer, and there

resume execution exactly where it left off. Alternately, for enterprise servers, a feature called VMotion allows the migration of operational guest virtual machines between similar but separate hardware hosts sharing the same SAN.

However, unlike an emulator, such as Virtual PC for PowerPC Macintosh computers, VMware software does not emulate an instruction set for different hardware than is physically present. Problems occur when the virtual machine guest is moved between hardware hosts using different instruction sets (such as found between Intel and AMD CPUs), or between hardware hosts with a differing number of CPUs.

Conventional emulators (such as Bochs) emulate the microprocessor, executing each guest-CPU instruction by calling a software subroutine on the host machine that simulates the function of that CPU instruction. This abstraction allows the guest machine to run on host machines with a different type of microprocessor, but also operates very slowly. Dynamic recompilation offers an improvement on this approach: it dynamically compiles blocks of machine instructions the first time they execute, and later uses the translated code directly when the code runs a second time. Microsoft's Virtual PC for Mac OS X takes this approach.

VMware Workstation takes an even more optimized path, and uses the CPU to run code directly whenever possible (as, for example, when running user-mode and virtual 8086 mode code on x86). When direct execution cannot operate, VMware software re-writes code dynamically. This occurs with kernel-level and real-mode code. VMware puts the translated code into a spare area of memory, typically at the end of the address-space, which it can then protect and make invisible using the segmentation mechanisms. For these reasons, VMware operates dramatically faster than emulators, running at more than 80% of the speed that the virtual guest operating-system would run on hardware. VMware Inc. boasts an overhead as small as 3%-6% for computationally-intensive applications.

Although VMware virtual machines run in user-mode, VMware Workstation itself requires installing various drivers in the host operating-system, notably in order to dynamically switch the GDT and the IDT tables. Many people erroneously believe that VMware and Virtual PC replace offending instructions or simply run kernel-code in user-mode. Both of these approaches run into difficulties on x86-based platforms. Replacing instructions means that if the code reads itself it may fail to find the expected content; one cannot protect code against reading and at the same time allow normal execution; replacing in-place becomes complicated. Running the code unmodified in user-mode will also fail, as most instructions which just read the machine-state do not cause an exception and will betray the real state of the program, and certain instructions silently change behavior in user-mode. One must always rewrite; performing a simulation of the current program counter in the original location when necessary and (notably) remapping hardware code breakpoints.

2.6 SIMILARITIES AND DISSIMILARITIES FROM PREVIOUS RESEARCH

There are 15 previous researches that had been studied to make a comparison to my project:

2.6.1 Implementing a Minimal Lab for an Undergraduate Network Security Course (Tom Wulf, 2003)

The researcher describes the implementation and usage of a minimal lab in an undergraduate computer security course covering the basics of network security. The similarity is the implementation of a security lab for computer students.

2.6.2 Designing a Flexible and Multipurpose Remote Lab for the IT Curriculum (Steve Rigby, 2006)

This research is all about analyzing current trends in remote lab design and explores a desig[v]t intends to increase utilization between courses, lower costs, ease management, and reduce the time needed to implement remote labs. The similarities are designing and implementing a remote lab.

2.6.3 A Laboratory-based Course on Internet Security (Prabhaker Mateti, 2003)

The researcher discusses the security course and explains how others can set up their own labs to teach this course. All the laboratory work is conducted in a laboratory. The similarities are the phase needed to setup a laboratory for security course

2.6.4 Design of a Distributed Computer Security Lab (Ping Chen, 2004)

This research is mainly about the general model of distributed computer security lab and the implementation of it. The similarities are the implementation of a computer security lab.

2.6.5 Using VMware for Dual Operating System (Rance D. Necaise, 2001)

The researcher reviews the common approaches and introduces the use of VMware for dual operating system. The similarities are the researcher used VMware to install dual operating system on a single host.

2.6.6 Using VMware and Live CD's to Configure a Secure, Flexible, Easy to Manage Computer Lab Environment (David Collins, 2006)

This research is discussing about how to setup and configure a computer lab environment using VMware and Live CD. The similarities are the tool used in this research is the same as my research.

2.6.7 Protecting Browser St 13 rom Web Privacy Attacks (Collin Jackson, 2006)

This research is about protecting web browser from being attack and methods done in order to counter the attack. The similarities are the study on the attack and the counteract measures done in order to overcome it.

2.6.8 Abstracting Application-Level Web Security (David Scott, 2002)

The researcher investigates new tools and techniques which address the problem of application-level of web security. He also presented a tool which can assist programmers to develop secure applications which are resilient to a wide range of attacks. The similarities are addressing the problem of application-level of web-security and how to defend against various attacks.

2.6.9 Scalable Network-based Buffer Overflow Attack Detection (Fu-Hau Hsu, 2006)

This research paper presents a network-based low performance overhead buffer overflow attack detection system called Nebula1, which can detect both known and zero-day buffer overflow attacks based solely on the packets observed without requiring any modifications to the end hosts. The similarities are method used to detect a network from being attacked.

2.6.10 Hardening Web Browsers against Man-in-the-middle and Eavesdropping Attacks (Haidong Xia, 2005)

This research is mainly about how to harden web browsers from being attacked. The similarities are about how to harden web browsers.

2.6.11 A Requires/Provides Model for Computer Attacks (Karl Levitt, 2001)

This research describes a flexible extensible model or computer attacks, a language for specifying the model, and how it can be used in security applications such as vulnerability analysis, intrusion detection and attack generation. The similarities are the study on various attacks on the networks and methods used to defend against it.

2.6.12 Learning Attack Strategies from Intrusion Alerts (Peng Ning, 2003)

This paper presents techniques to automatically learn attack strategies from correlated intrusion alerts, measures the similarity between attack strategies in terms of the cost to transform one strategy into another and some experimental results, which demonstrate the potential of the proposed techniques. The similarities are the study on the attack and strategy to overcome it.

2.6.13 Defending Against an Internet-Based Attack on the Physical World (Simon Byers, 2002)

The researcher discuss the dangers that scalable Internet functionality may present to the real world, focusing upon an attack that is simple, yet can have great impact and various solutions to this class of attack and hope to provide a warning to the Internet community of what is currently possible. The similarities are the study on the attack and various solutions to the attacks.

2.6.14 Towards Agile Security in Web Applications (Vidar Kongsli, 2006)

The researcher presents an approach that he used to address security when running projects according to agile principles. Penetration testing, system hardening and securing deployment also have been elaborated in this research. The similarities are the deployment of security and how to harden a system.

2.6.15 Real Attacks on Virtual Networks: Vivaldi out of Tune (Walid Dabbous, 2006)

In this paper, the researcher identifies different attacks against coordinates embedding systems and study the impact of such attacks on the recently proposed Vivaldi decentralized positioning system. The similarities are about different attacks and the study of the impacts of these attacks to the system. The dissimilarities are it used Vivaldi decentralized positioning system.