

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2006

WebISMS: (Web-Based Information Security Management System): A prevention information security tool

Nam Kim

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Information Security Commons](#)

Recommended Citation

Kim, Nam, "WebISMS: (Web-Based Information Security Management System): A prevention information security tool" (2006). *Theses Digitization Project*. 3525.

<https://scholarworks.lib.csusb.edu/etd-project/3525>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

WEBISMS (WEB-BASED INFORMATION SECURITY MANAGEMET SYSTEM)

A PREVENTION INFORMATION SECURITY TOOL

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by

Nam Kim

June 2006

WEBISMS (WEB-BASED INFORMATION SECURITY MANAGEMET SYSTEM)

A PREVENTION INFORMATION SECURITY TOOL


A Project
Presented to the
Faculty of
California State University,
San Bernardino

by

Nam Kim

June 2006

Approved by:



Arturo I Concepcion, Chair, Computer Science

08 May 2006
Date



Kerstin Voigt



Ernesto Gomez

© 2006 Nam Kim

ABSTRACT

The motivation and idea of this project came from my personal experiences as system and network administrator for five years in the College of Natural Sciences, CSUSB. The College and campus in general, are always under cyber attack by direct penetration methods, and all kinds of viruses, worms, and spywares. And there is no complete protection in information security.

The project, WebISMS, is a prevention approach in information security. The features of WebISMS are as following: Web based GUI to access the system any time and anywhere, Open-source network security tools integrated and implemented with LAMP(Linux, Apache, MySQL, PHP/Perl), PHP used for generating dynamic Web pages, Modular based System for easy upgrade, Back-end database in which all the results from the network security tools are stored.

The network tools integrated with WebISMS are NESSUS, NMAP, and SYSLOG-NG. NESSUS module scans the machine for vulnerability check. NMAP module checks for open ports and keeps a record in the database. SYSLOG-NG module receives logs remotely and save them in the database in real-time in order to monitor the client machine's network activities and system status.

WebISMS is now deployed in Institute of Applied Supercomputing lab, and is proven that this system is efficiently working as a information security assessment/audit tool.

Future direction of WebISMS consists of adding more information tools as a module-based, applying clustering technology, and developing more user-friendly GUI.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to many people whose help has been crucial to my success in completing my project. First of all, I am very thankful to Professor Concepcion, my project advisor, for his direction and wise guidance of my project. Secondly, I owe a great deal to Professor Voigt and Professor Gomez, two committee members, whose supervision and advice have contributed to my study.

My heartfelt appreciation also goes out to all the people of Network Groups who have helped me during the progress of working on my project. In particular, my special gratitude is extended to Kwon Soo Han, the systems administrator of the Network Groups, who has been generous with his time and has provided helpful advice and careful guidance in working out certain problems.

Finally, I would like to thank my wife who has always supported me and has been a source of much encouragement.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	v
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER ONE: INTRODUCTION	
1.1 Background	1
1.2 Tools	4
1.2.1 NESSUS	4
1.2.2 NMAP	4
1.2.3 SYSLOG-NG	5
1.3 Purpose of the Project	5
1.4 Organization of Documentation	6
CHAPTER TWO: SOFTWARE REQUIREMENTS SPECIFICATION	
2.1 Introduction	7
2.1.1 Purpose	7
2.1.2 Scope	8
2.1.3 Definitions, Acronyms, and Abbreviations	9
2.1.4 Overview	10
2.2 Overall Description	11
2.2.1 Product Perspective	11
2.2.2 System Interface	11

2.2.3	User Interface	15
2.2.4	Hardware Interface	16
2.2.5	Software Interface	17
2.2.6	Memory Constraints	18
2.2.7	Operations	18
2.2.8	Product Functions	19
2.2.9	User Characteristics	22
2.2.10	Assumptions and Dependencies	22
2.3	Software Specific Requirements	22
2.3.1	External Interface	22
2.3.2	Performance Requirements	47
2.3.3	Logical Database Requirements	48
2.3.4	Design Constraints	48
2.3.5	Software System Attributes	48

CHAPTER THREE: SOFTWARE DESIGN

3.1	Architecture Design	50
3.1.1	System Design	50
3.1.2	Database Design	53
3.2	Detailed Design	65
3.2.1	Login Class	65
3.2.2	Syslog Class	66
3.2.3	View24Hr_Result Class	68
3.2.4	View_7days_Result Class	71

3.2.5	NESSUS Class	72
3.2.6	Report_Nessus Class	73
3.2.7	Run_Nessus Class	74
3.2.8	NMAP Class	77
3.2.9	Run_Nmap Class	78
3.2.10	Report_Nmap Class	79
 CHAPTER FOUR: SOFTWARE QUALITY ASSURANCE		
4.1	Introduction	81
4.2	Unit Testing	81
4.3	Integration Testing	83
4.3.1	Login	83
4.3.2	Syslog Module	86
4.3.3	NESSUS Module	94
4.3.4	NMAP Module	106
4.4	System Acceptance Testing	112
 CHAPTER FIVE: INSTALLATION AND MAINTENANCE		
5.1	Directory Structure	127
5.2	Installation	129
5.2.1	NMAP and NESSUS Installation	129
5.2.2	SYSLLOG-NG Installation	129
5.2.3	Database Installation	132
5.2.4	WebISMS Installation	132
5.3	Maintenance	133

5.3.1 Syslog Module	133
5.3.2 NESSUS PLUG-INS Update	133
CHAPTER SIX: CONCLUSIONS AND FUTURE WORK	
6.1 Conclusion	134
6.2 Future Work	137
REFERENCES	139

LIST OF TABLES

Table 2.1. Hardware Specification for the Development Phase	17
Table 2.2. Software Specification for the Development Phase	18
Table 3.1. User Table	58
Table 3.2. Logs Table	59
Table 3.4. Results Table	60
Table 3.5. Timestamps Table	61
Table 3.6. Hoststats Table	62
Table 3.7. Portstats Table	63
Table 3.8. Targets Table	64
Table 3.9. Runlist Table	65
Table 4.1. Unit Testing Results	82
Table 4.2. System Testing Results	113

LIST OF FIGURES

Figure 2.1.	Assessment Mode Diagram	13
Figure 2.2.	Monitoring Mode Diagram	14
Figure 2.3.	Auditing Mode Diagram	15
Figure 2.4.	Use Case Diagram	21
Figure 2.5.	Login Page	23
Figure 2.6.	Main Page	24
Figure 2.7.	Syslog Front Page	26
Figure 2.8.	Detailed Log Page	27
Figure 2.9.	Syslog Search Main Page	28
Figure 2.10.	Syslog Search Result	29
Figure 2.11.	NESSUS Main Page	30
Figure 2.12.	Creating a Target File	31
Figure 2.13.	Creating a Report File	32
Figure 2.14.	NESSUS Result	33
Figure 2.15.	NESSUS Report Page,.....	35
Figure 2.16.	NESSUS Report Menu Page	38
Figure 2.17.	List of Vulnerable Machines	39
Figure 2.18.	Result Page from Clicking a Icon of the Previous Page	40
Figure 2.19.	NMAP Main	41
Figure 2.20.	Run NMAP	42
Figure 2.21.	NMAP Report Main	43

Figure 2.22. List of Machines Scanned	44
Figure 2.23. Scanned Result of Single Machine	45
Figure 2.24. Result Based on Port number and Protocol Type	46
Figure 2.25. list of RunID	47
Figure 3.1. Architecture in Class Diagram of WebISMS	53
Figure 3.2. Login Entity-Relationship Diagram	54
Figure 3.3. Syslog Entity-Relationship Diagram	55
Figure 3.4. NISSUS Entity-Relationship Diagram	56
Figure 3.5. NMAP Entity-Relationship Diagram	57
Figure 3.6. Login Class	66
Figure 3.7. Syslog Class	68
Figure 3.8. View24Hr_Result	70
Figure 3.9. View_7days_Result	71
Figure 3.10. NISSUS	72
Figure 3.11. Report_Nessus	74
Figure 3.12. Run_Nessus	76
Figure 3.13. NMAP	77
Figure 3.14. Run_Nmap	79
Figure 3.15. Report_Nmap	80
Figure 4.1. Demonstration of Main Page	84
Figure 4.2. Exception Handling Page of Login Error	85

Figure 4.3.	Module Menu Page	86
Figure 4.4.	Syslog Main Page	88
Figure 4.5.	Demonstration of zzyzx Bar Click	89
Figure 4.6.	Demonstration of Syslog Search	90
Figure 4.7.	Result of Syslog Search Demonstration	91
Figure 4.8.	Last 24 Hours Alert Messages	92
Figure 4.9.	Last 24 Hours Critical Messages	92
Figure 4.10.	Last 24 Hours System Error Messages	93
Figure 4.11.	Last 7 Days Log	94
Figure 4.12.	NESSUS Main	95
Figure 4.13.	Demonstration of Running NESSUS	97
Figure 4.14.	NESSUS Module Main Page	102
Figure 4.15.	Demonstration of Show All of Hosts Having Vulnerabilities	103
Figure 4.16.	Demonstration of a Icon Clicked	104
Figure 4.17.	Demonstration of Generating a Report Based on Single Host	105
Figure 4.18.	Results from Show a NESSUS Report for Single Host	106
Figure 4.19.	NMAP Main Page	107
Figure 4.20.	Demonstration of Single Host Scan	108
Figure 4.21.	Demonstration of Instant Result	109
Figure 4.22.	Demonstration of IP Block Scan	110
Figure 4.23.	Error Message from Typed in Incorrect	

	Runid	111
Figure 4.24.	Error Message from Typed in Incorrect IP	112
Figure 4.25.	Screenshot of NISSUS Process Started	115
Figure 4.26.	Screenshot of NISSUS process without Driving Web Page	116
Figure 4.27.	Screenshot of Wrong Username Entered ...	117
Figure 4.28.	Screenshot of Wrong Username or Password Handled	118
Figure 4.29.	Screenshot of Empty Target File Box	119
Figure 4.30.	Screenshot of Empty Box Error Message	120
Figure 4.31.	Screenshot of Block a Direct Access	121
Figure 4.32.	Screenshot of an Authorization Required Message	122
Figure 4.33.	Screenshot of Wrong Runid Entered	123
Figure 4.34.	Screenshot of Character Error Message	124
Figure 4.35.	Screenshot of Wrong IP Entered	125
Figure 4.36.	Screenshot of IP Error Message	126
Figure 5.1.	Directory Structure Diagram	127
Figure 5.2.	Configuration File of syslog-ng.conf	130
Figure 6.1.	Time Spent for Syslog Monitoring	136

CHAPTER ONE

INTRODUCTION

1.1 Background

This project is the result of my work experiences and ideas about information security. When I first joined College of Natural Science at CSU, San Bernardino five years ago, my first assignment was setting up a firewall for the College. In 2001, CSUSB did not have a campus wide firewall yet. Therefore, one of my supervisors, Dr. Torner (now the information security officer at CSUSB) was concerned about cyber attacks and urged me install a firewall. I actually implemented two firewalls: one for Computer Science and Mathematics Departments, and the other for seven departments in the College. It took me nine months to complete this work. After the firewalls are running, cyber attacks like penetration attack or denial of service was greatly reduced, but there were other problems found. The firewall minimizes open-network-services. If the system has a vulnerable service which is open to Internet from the firewall, it is possible to break in the system within a few days. Our College is a well known target from hackers in all over the world. They are

running scanning-tools or attacking tools to find vulnerabilities on an hourly basis. For example, a few years ago, many graduate students in Department of Computer Science were working on Web programming and running their own Web servers. They asked me open port 80 from the firewall in order to access their Web servers from the outside of the campus. However, one of their Web servers had a vulnerable web server program, and the machine was compromised the next day. This compromised machine became a media server illegally selling movies and songs over the Internet. After these incidents, Kwon Han, system administrator in Computer Science made a list of guidelines and policies for installing and managing servers for graduate students. He also provided first-hand help to the students for the machine setup. His crucial involvement really improved the security of the graduate project machines. Second limitation was from the viruses and worms. Most of Windows workstations in the university had anti-virus program installed, but the problem was from viruses infected laptops brought in by the faculty and the students. When the infected laptop was connected to the campus network, it infects hundreds of campus workstations within hours and campus network speed slowed down

drastically, and finally firewalls crashed. Fortunately, Information Security Office under IRT has been monitoring the campus network data jack by data jack. If any abnormal network activity is detected from the machine connected to the data jack port, IRT shuts off the data jack port to prevent further network breach and informs the personnel in charge of the suspected machine.

These incidents gave me a lesson. Firewalls are not enough to protect our network and systems. Cyber attacks including viruses and worms were inevitable (it even infects other technologies like wireless network, PDA, Cell Phone). There is always vulnerability in any kind of software of network system. When you go and check the Internet security report site, you will be surprised to find new security holes, virus/worm, or any kind of vulnerabilities almost everyday. Therefore, I conclude that prevention is a more practical and effective method than protection. In order to know more about intrusion incidents, I subscribed to the network security sites and check vulnerability reports daily. There are network security assessment/auditing tools running periodically to report any infected machine or vulnerable service. Monitoring servers and the network itself are also included.

For carrying on those tasks, I use network security tools, NMAP and NESSUS and logging system, SYSLOG-NG.

1.2 Tools

The tools, NMAP, NESSUS, and SYSLOG-NG that were used in this project will be explained in this section.

1.2.1 NESSUS

This network security assessment program is a very powerful security scanner in three ways. First, it is able to perform both remote and local security checks. Second, NESSUS is client/server technology. Servers can be placed at various points on a network allowing tests to be conducted from a central client. NESSUS use plug-ins technology. NESSUS plug-ins are very much like virus signatures in a common virus program. After a new vulnerability is released to the public, the NESSUS community writes a new NESSUS plug-in and releases it to the public. NESSUS always keeps updated plug-ins.

1.2.2 NMAP

This is a multifaceted utility that is used to scan a range of IP addresses, identify active systems, and determine which ports on those systems that are opened. Nowadays, many virus/worm/spyware have a feature that once

they take over the system, they create a random port to attack other system remotely or attempt to open a port for peer-to-peer connection. In this case, NMAP is a very useful utility to detect illegitimate ports.

1.2.3 SYSLOG-NG

SYSLOG-NG is a next generation SYSLOG system which is not only generating logs for a local system but also is able to receive logs remotely from multiple systems in real time.

1.3 Purpose of the Project

While using NMAP and NESSUS, I found two limitations. First, NMAP is UNIX shell command based and NESSUS is X-window GUI based. It is not easy to run those tools remotely. The environment to be able to run both tools remotely is X-windows, but X-windows are a vulnerable application. Second, scanning results from these tools are ephemeral. Their results are just shown on the screen.

To save their results, there is a need to use other utility or save them manually. To overcome these limitations, a program must be written to run these tools anytime and anywhere, the results are automatically stored in a database. A Web-based system is a perfect fit for

these conditions. If these security tools are implemented with Web server/client technology, the user will be able to drive these tools within the Web browser. This is the goal of this project, Web-based Information Security Management System.

1.4 Organization of Documentation

The remaining sections of this documentation will be organized as following: Chapter 2 describes the software requirements specification of WebISMS. Chapter 3 provides a description of the system architecture and detailed design. Chapter 4 describes the system test. Chapter 5 is the maintenance and user manual. Finally, Chapter 6 concludes the project and lists suggestions for future developments.

CHAPTER TWO
SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Introduction

2.1.1 Purpose

The purpose of the WebISMS project is to provide a framework for effective and efficient network vulnerability management in WAN or LAN environment. In order to do that, WebISMS will integrate network security tools as module based, save all the results had run from the tools to the database for analysis and monitoring purposes, and finally WebISMS will be operated through the Web browser for ubiquitous and timeless manner. In operational purpose, WebISMS accesses and monitors the computers in a network in three stages: First, testing stage that the computer is freshly installed without vulnerability check, Second, production stage that the computer is configured with application programs exposed to the Internet, Third, emergency situation that any suspicious network activities found such as enormous network bandwidth consumption, network scanning activity or any application layer malfunctions related to the network security breach.

Consequently WebISMS will be able to be operated for the network security auditing, monitoring and assessment.

2.1.2 Scope

WebISMS selects and implements two open-source network security tools, NESSUS and NMAP integrated with web server and database to enhance the information security through the three modules. In the first module, Assessment module, all of the servers and workstations before deploying in the production line, are assessed by NESSUS; NESSUS checks the computer's vulnerable ports and services, suggests patches, and saves the results in the database for the future reference. Second one is the monitoring module. Once the computer is in the production line after being patched, system logs are sent to the WebISMS's SYSLOG-NG for monitoring purpose. As WebISMS is collecting the logs in the database, the system administrator is able to monitor the status of system and network activities, including finding illegal activities through the Web browser. Third one is the auditing module. If any suspicious activities are found in the second stage, the system administrator is able to use scanning tool, NMAP to compare the services and ports currently open in the computer with the information based on the database to decide whether it is compromised.

If any vulnerable ports and services are found, the network auditing tool, NMAP will be used to audit the whole network for locating any other compromised or infected machine. Besides, database is a crucial component that support these modules as for analyzing results from the modules. Based on the information saved in the database, the system administrator is able to analyze how often and when is the peak time the major attacks are coming, and what services and ports are most vulnerable in the network environment. This information can help the system administrator to create or change the rule-sets of the firewall, patch the vulnerable programs, find the misconfigured system and virus infected system, and even plan to buy any needed security equipment.

2.1.3 Definitions, Acronyms, and Abbreviations

- DBMS - Database Management System
- HTML - Hyper Text Markup Language
- HTTP - Hyper Text Transfer Protocol

The client/server protocol that defines how messages are formatted and transmitted on the World Wide Web

- IP - Internet Protocol
- LAN - Local Area Network

- MySQL

MySQL is a popular and robust database that supports key subsets of SQL on both Linux and Unix system

- NESSUS

NESSUS is a comprehensive vulnerability scanning program. It consists of NESSUSD, the NESSUS daemon, which does the scanning, and NESSUS client, which presents the results to the user.

- NMAP

Network exploration tool and security scanner

- SYSLOG-NG

Service that collects system logs locally or remotely.

- WAN - Wide Area Network

2.1.4 Overview

The format of the rest of this project is as follows. Section 2.1 has an introduction that contains an overview of the entire software requirements specification. It defines the scope and purpose of the project and the requirements document, defines terms used in the SRS. In section 2.2, I discuss overall description that affect the product and its requirements including product perspective, system interface, user interface, hardware/software

interface, memory constraints, operations, and product functions. Lastly, I discuss software specific requirements.

2.2. Overall Description

2.2.1 Product Perspective

The WebISMS consists of a computer server with Web and database server applications. The server is loaded with Apache for Web server, PHP for internal Web programming, and MySQL is for the back-end database. Besides, information security scanning tools such as NESSUS and NMAP are installed in the WebISMS for information security assessing and auditing. SYSLOG-NG server is also loaded for the monitoring module, and PERL will be used for storing logs from SYSLOG-NG server.

2.2.2 System Interface

All system interfaces provide the administrator with three different modules to start and stop the information security assessment/monitoring/auditing and to get results from the database.

a) Assessment Mode (see Figure 2.1 below)

- This mode is based on the network assessment tools, NESSUS and NMAP
- The user launches NESSUS from his/her Web browser for assessing the computer which is not yet in the production line or not exposed to the internet
- Results of assessment are stored in the database
- Based on the results, if necessary, the machine will be patched, updated, reconfigured before it is deployed in the production line.
- NMAP scans the machine's network ports whether only necessary ports are open.
- Results are stored in the database for a future reference

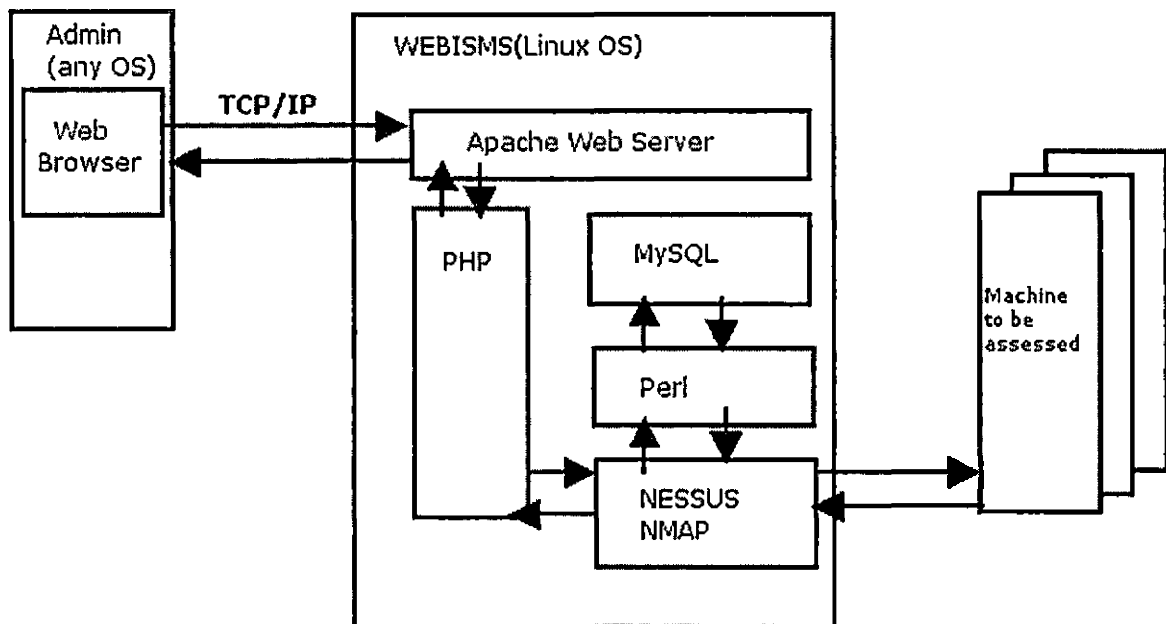


Figure 2.1. Assessment Mode Diagram

b) Monitoring Mode (see Figure 2.2 below)

- SYSLOG-NG is a core program for this module
- Once the machine is in the network, their logs are transferred to the database of WebISMS in real-time mode by SYSLOG-NG server.
- All of logs collected in the database are able to be queried by the system administrator through his/her Web browser for monitoring status of machine or of any network activities.

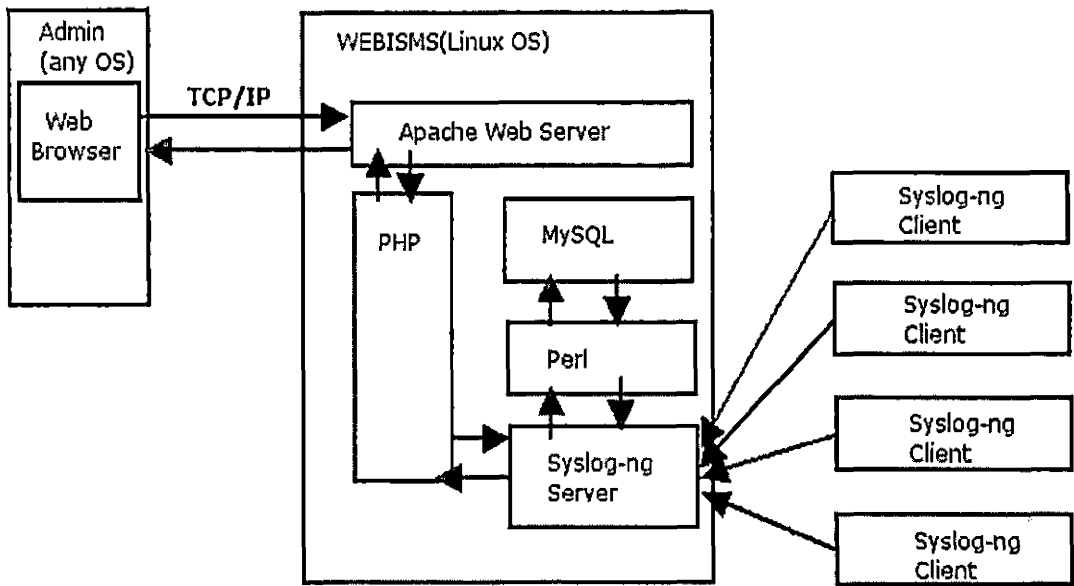


Figure 2.2. Monitoring Mode Diagram

c) Auditing Mode(see Figure 2.3 below) :

- NMAP and NESSUS is a core tool for this mode.
- In case of any suspicious or illegal network activities are found from the monitoring mode, NMAP will be launched to scrutinize the machine.
- If unknown ports or services are running, the machine will be disconnected from the network.
- Rest of network will be scanned by NESSUS for finding any updated vulnerabilities.

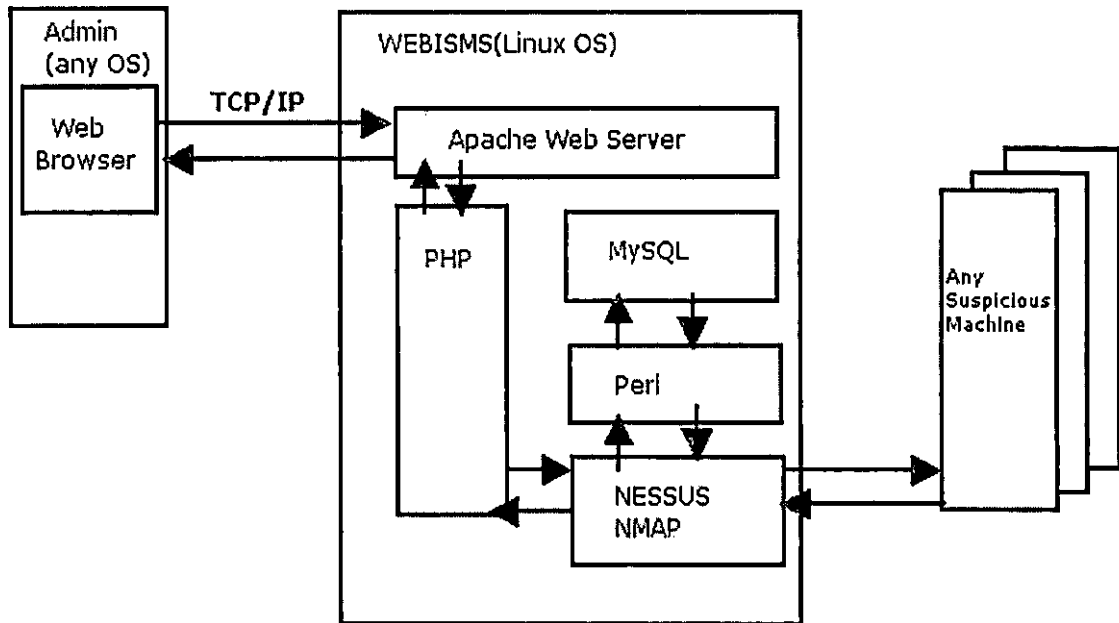


Figure 2.3. Auditing Mode Diagram

2.2.3 User Interface

The only user of WebISMS is the system administrator or any person in charge of managing network information security. WebISMS is a intranet server that the access from the internet must be denied for the security reason. WebISMS provides basically three modules, NESSUS for assessment, SYSLOG for monitoring, and NMAP for auditing. NESSUS has two sub functions, "Running NESSUS" and "Report." NMAP has also two sub functions similar to NESSUS, "Running NMAP" and "Report." Because SYSLOG is

monitoring module, it has quick-search sub functions that are able to view the monitoring results one click.

Let's imagine a scenario using WebISMS. When a machine is just installed Operating System and applications, WebISMS is assessing the machine whether it is safe to connect to internet or not. If any vulnerability is found, it is recommended to patch a security holes. Afterward, the machine is in the production line and configured to send its system logs to WebISMS's SYSLOG-NG modules to be monitored. If any suspected logs are found in the SYSLOG-NG module, NMAP, auditing module, will be launched to scan the ports of the machine. If unknown ports are open, the machine will be quarantine stage that machine will be disconnected from the network and report to information security office for further investigation.

2.2.4 Hardware Interface

Due to the sensitivity of this project that running network security tools such as NESSUS and NMAP, using this project might be like peeking inside of someone's house or checking all of doors and windows without owner's permission, this project is developed and tested within Computer Science Research Lab Network environment. Although a single machine is used for the development phase

of this project, multiple machines are recommended when WebISMS is implemented in the production line. Table 2.1 is the hardware specification for the development phase.

Table 2.1. Hardware Specification for the Development Phase

Hardware	Specification
Processor	2.5 GHz
Memory	512 MB
Storage	60 GB

2.2.5 Software Interface

To implement this project, network security tools, NESSUS and NMAP, web server, database server are needed in the Linux operating system. Moreover, Photoshop, Dreamweaver, PHP, and PERL are used for developing dynamic web pages and graphics.

Table 2.2. Software Specification for the Development Phase

Software	Specification
Operating System	Fedora Core 4
Web Server	Apache 2.0
PHP	Php 5.0.4
PERL	Perl 5.8.6
Database	MySQL 4.1.1
Network Assessment tool	NESSUS 2.2.4
Network Scanning tool	NMAP 3.7.5
Syslog serer	SYSLOG-NG 1.4
PHP Graphic Library	JpGraph 1.2

2.2.6 Memory Constraints

MySQL and NESSUS are known to be memory intensive software, therefore it is recommended that the Linux server, have at least 512MB of memory in the system to effectively run the WebISMS.

2.2.7 Operations

The WebISMS will be maintained on the server based on x86 system. The server will need to be up 364 days a year allowing for one day of system maintenance.

2.2.8 Product Functions

Figure 2.4 shows Use Case Diagram that graphically depicts the users and principal functions of this system. The functions are further described in the following subsections 2.2.8.1 to 2.2.8.5

2.2.8.1 Running-NESSUS function. To launch a NESSUS, this function gets all the information to kick off the NESSUS. This function is generating a target file that contains a target machine's IP or a target local network IP. Once assessment scanning is successfully completed, the result is generated in a HTML format file and at the same time, stored in the database.

2.2.8.2 NESSUS-Report function. This function is simply generating a report from MySQL database. There are two sub functions. First, "report all machines having vulnerabilities. Second one is retrieving a vulnerability report by a IP based when a specific IP is typed in this function.

2.2.8.3 SYSLOG Monitoring function. This function generates a dynamic web page that shows a bar graph and a sub menu section. In bar graph, Number of logs based on the four priorities (alert, critical, error, warning) are drawn in a bar graph per each syslog client machine. If

any single bar is clicked, detail information is viewed. Sub menu section has six functions. First one is 'syslog search' function. It is like a search engine type of function that can easily search a log by time based, priority based, host based, or any specific message based. Second to fourth function shows detail messages based on the priority, alert, critical, and system error. Fifth function generate a bar graph similar to the main function, but it is based on logs in last seven days.

2.2.8.4 Running NMAP function. When system administrator finds any suspicious activities in the Monitoring stage, This is the function for next step. System administrator submits IP of a suspicious machine to scan the ports open. Once scan is complete, Report is generated in HTML format and saved in database.

2.2.8.5 Report NMAP function. History of scanning ports for a specific machine or a whole network is retrieved by this function. With this function, System administrator is able to find a suspicious port. For example, if a new virus is reported with a specific port attacking, System administrator is able to identify a machine the virus infected as querying the port open in the database by this function. Other example to find any

suspicious or vulnerable machine using this function is comparing open-ports history of the machine.

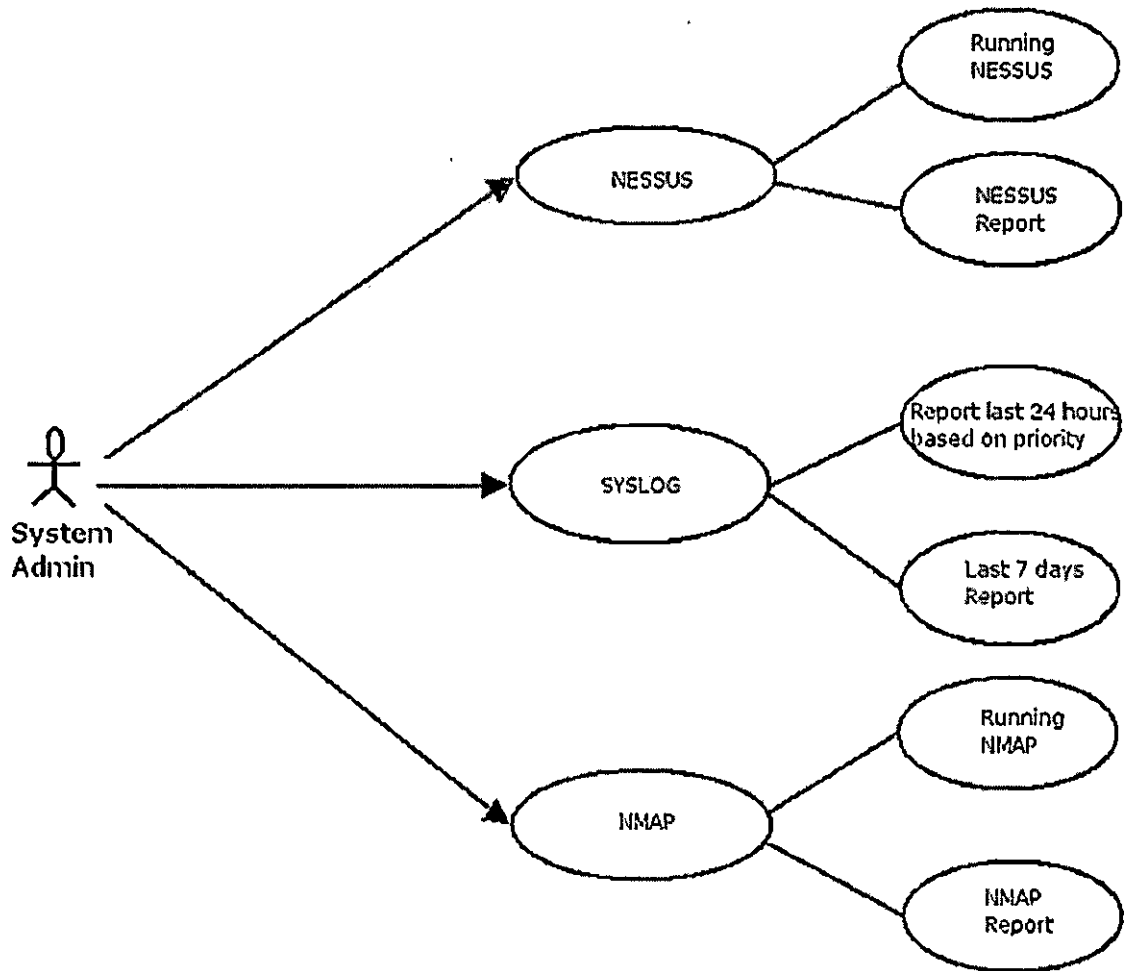


Figure 2.4. Use Case Diagram

2.2.9 User Characteristics

The intended users of the WebISMS are network administrator or system administrator managing multiple machines. The user will need to have substantial knowledge in information security and in addition will need to have a fundamental knowledge of network and able to read and understand system events and any other logs.

2.2.10 Assumptions and Dependencies

Although this system is being implemented on a Linux operating system using an MySQL database, it is possible to port the code to other platforms. Since the program is coded entirely in PHP and Perl, any platform with a PHP/Perl installed, a PHP compatible web server, and a SQL compatible relational database can conceivably, with small modifications, run the WebISMS program. Furthermore, any other scanning tools besides NESSUS and NMAP will be able to be implemented in this system.

2.3 Software Specific Requirements

2.3.1 External Interface

This section describes all detailed inputs and outputs of WebISMS.

2.3.1.1 Login. This is the first page of WebISMS. (See Figure 2.5). All other pages of WebISMS are accessed through this login page. A user provides his/her login ID and password to access WebISMS. When the user provides the correct login ID and password, this page authorizes the user and creates a new session.

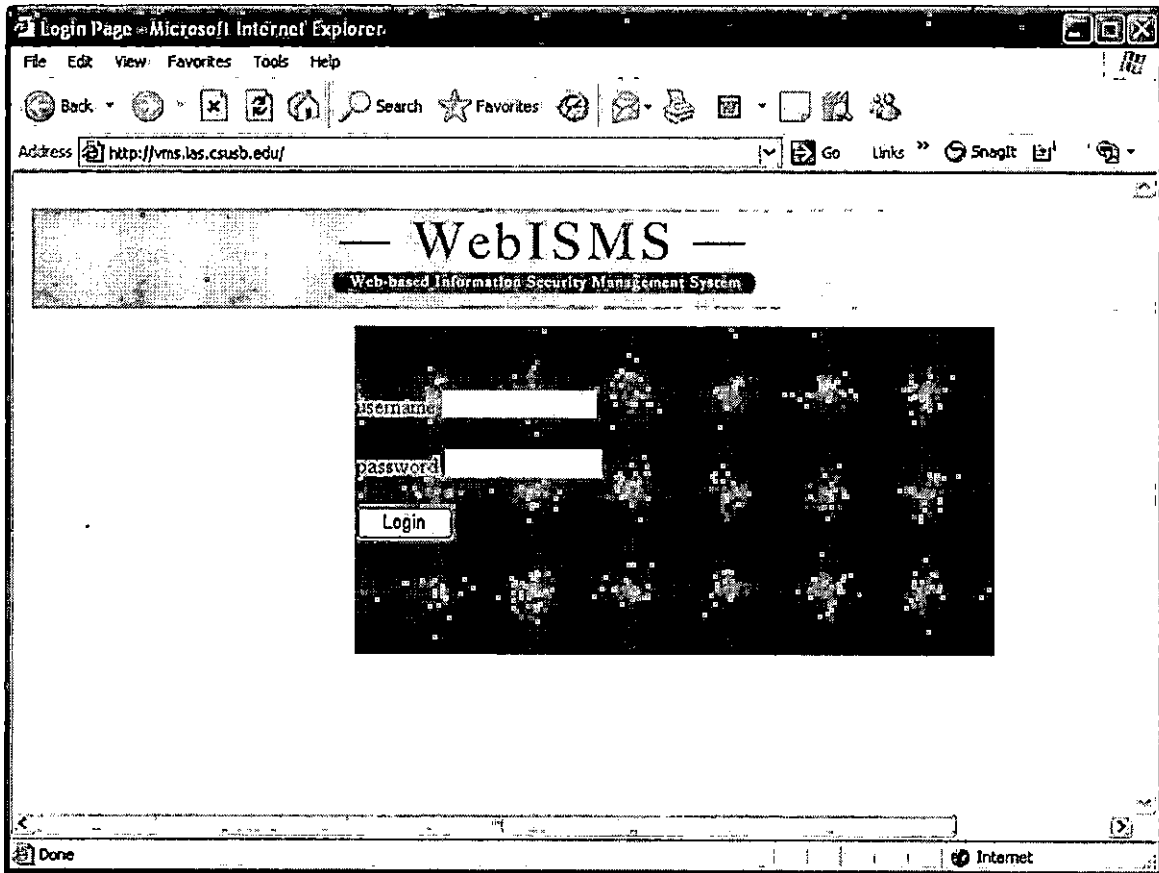


Figure 2.5. Login Page

2.3.1.2 Home. This page is the home of WebISMS (see Figure 2.6). This page provides three main modules, SYSLOG, NESSUS, and NMAP. If a user clicks a any one of them, it directs to the main page of the module.

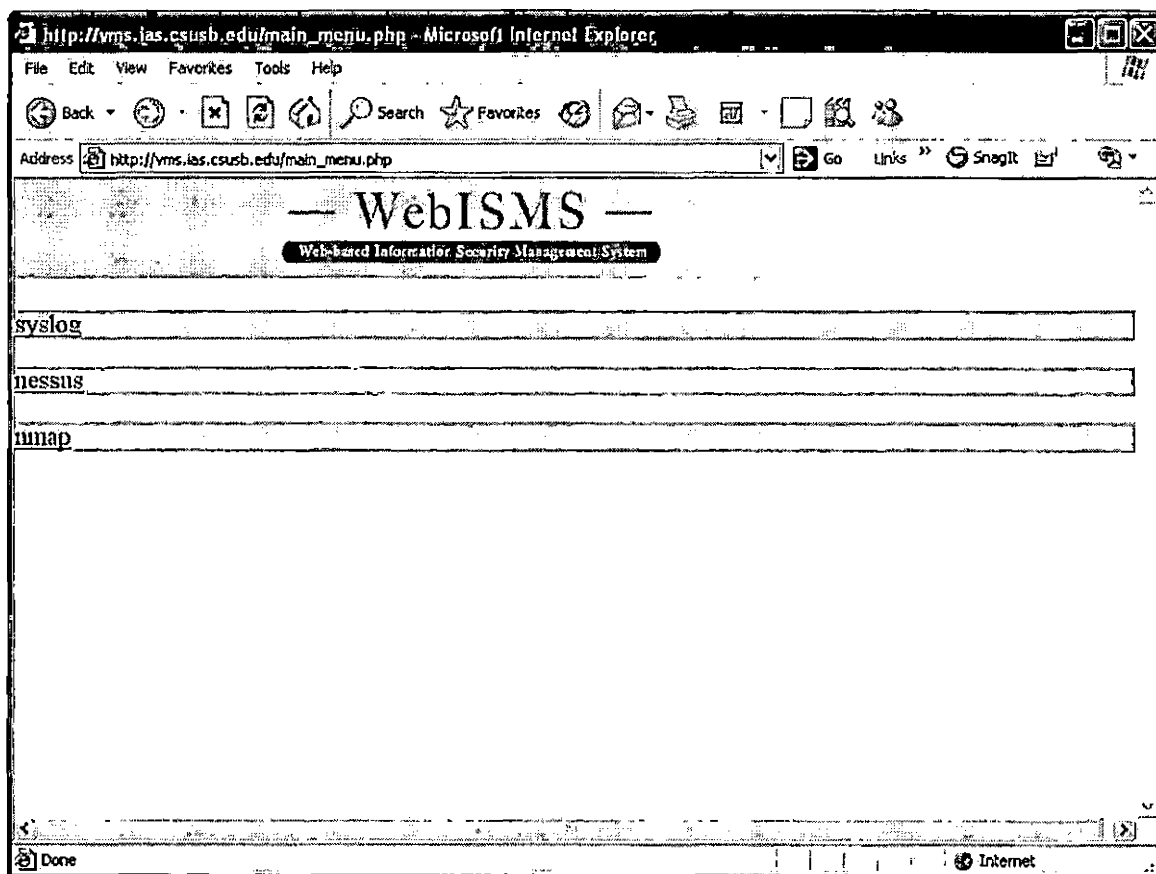


Figure 2.6. Main Page

2.3.1.3 Syslog. This is the main page of syslog module (Figure 2.7). It consists of a bar graph and sub functions. This main page shows each client's log status for last 24 hours in a bar chart. The color of each bar indicates alert as a red bar, critical as a orange bar, error as a brown, and warning as a yellow. Furthermore, if a user clicks a bar, it shows more detailed information. The sub functions are 'syslog search', 'Last 24 hours Alert Messages', 'Last 24 hours Critical error Messages', 'Last 24 hours System error Messages', and 'Last 7 days log.' 'Syslog search' function is a search engine type of function (I will explain it next page). 'Last 24 hours Alert Messages' shows alert logs for all of the client machines for last 24 hours. Similarly, 'Last 24 hours Critical error Messages', 'Last 24 hours System error Messages' show their category's logs in last 24 hours. When a user clicks 'last 7 days logs' function, it renders a bar charts which shows last 7 days logs.

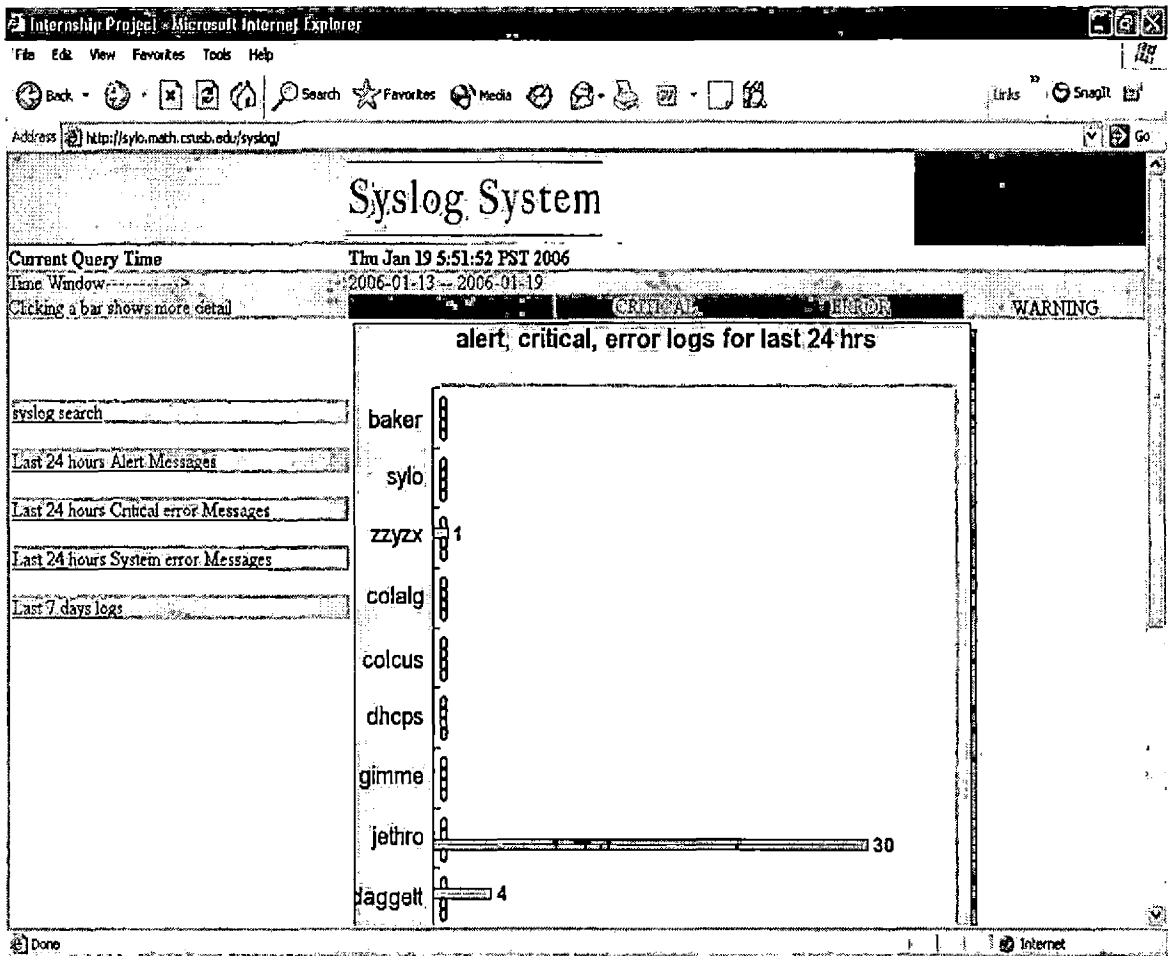


Figure 2.7. Syslog Front Page

2.3.1.4 bar-click result. When a user clicks any bar from the previous page, detailed information is queried. For example, syslog client machine, jethro has thirty system errors with a brown bar (see Figure 2.7). When the brown bar clicked, result page in Figure 2.8 below is shown.

It shows hostname, priority of log, date and time of log created, program where errors are created, detail message of the log.

host	priority	data	time	program	msg
jethro	err	2006-01-19	05:29:48	named	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied
jethro	err	2006-01-19	05:29:48	named	named[13918]: dumping master file: tmp-XXXXOpAM8i: open: permission denied
jethro	err	2006-01-19	04:26:27	named	named[13918]: transfer of 'csusb.edu/IN' from 139.182.2.1#53: failed while receiving responses: multiple RRs of singleton type
jethro	err	2006-01-19	00:16:42	named	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied
jethro	err	2006-01-19	00:16:42	named	named[13918]: dumping master file: tmp-XXXXdg8mok: open: permission denied
jethro	err	2006-01-18	22:40:31	named	named[13918]: transfer of 'csusb.edu/IN' from 139.182.2.1#53: failed while receiving responses: multiple RRs of singleton type
jethro	err	2006-01-18	18:45:44	named	named[13918]: dumping master file: tmp-XXXXxvsWM8: open: permission denied
jethro	err	2006-01-18	18:45:44	named	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied
jethro	err	2006-01-18	18:09:28	named	named[13918]: client 139.182.155.47#1068: update '155.182.139.IN-ADDR.ARPA/IN' denied
jethro	err	2006-01-18	18:00:24	named	named[13918]: transfer of 'csusb.edu/IN' from 139.182.2.1#53: failed while receiving responses: multiple RRs of singleton type
jethro	err	2006-01-18	17:46:33	named	named[13918]: client 139.182.155.47#3030: update '155.182.139.IN-ADDR.ARPA/IN' denied
jethro	err	2006-01-18	16:46:33	named	named[13918]: client 139.182.155.47#2894: update '155.182.139.IN-ADDR.ARPA/IN' denied
jethro	err	2006-01-18	15:46:32	named	named[13918]: client 139.182.155.47#2775: update '155.182.139.IN-ADDR.ARPA/IN' denied

Figure 2.8. Detailed Log Page

2.3.1.5 Syslog Search. This page shows an information input form that the user has to fill out to find any single log or a specific group of logs (see Figure 2.9) based on the options in the page. The user can find any logs by each client, by date, by time, by priority, or by search message, or by combined them.

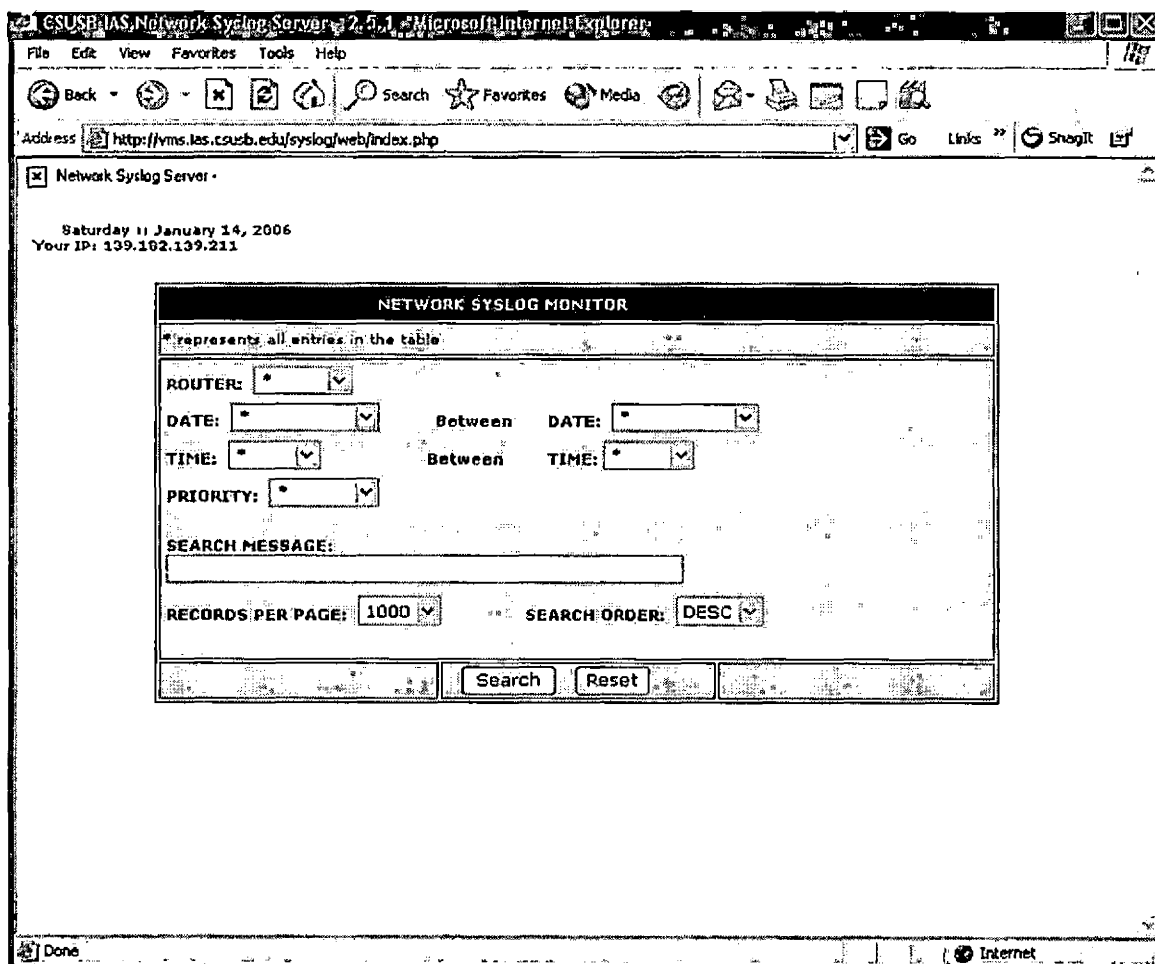


Figure 2.9. Syslog Search Main Page

2.3.1.6 Syslog Search result. This is a example output page from syslog search query (see Figure 2.10). Format of a output page is same as a bar-click output.

Thursday 11 January 19, 2006
Your IP: 139.182.139.211

NETWORK SYSLOG MONITOR RESULTS

BACK TO SEARCH
Number of Syslog Entries: 529

SEVERITY LEGEND
INFO DEBUG NOTICE WARNING ERR CRIT

SEQ	HOST	PRIORITY	DATE	TIME	MESSAGE
1036912	jethro	info	2006-01-19	05:44:04	crond(pam_unix)[25208]: session closed for user:root
1036890	jethro	info	2006-01-19	05:29:40	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: end of transfer
1036889	jethro	err	2006-01-19	05:29:40	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied
1036888	jethro	err	2006-01-19	05:29:40	named[13918]: dumping master file: tmp-XXXXOpAMBi: open: permission denied
1036867	jethro	info	2006-01-19	05:19:44	crond(pam_unix)[25672]: session closed for user:root
1036866	jethro	info	2006-01-19	05:19:44	crond[25673]: (root) CMD (run-parts /etc/cron.hourly)
1036865	jethro	debug	2006-01-19	05:19:44	pam_loginuid[25672]: set_loginuid failed opening loginuid
1036864	jethro	info	2006-01-19	05:19:44	crond(pam_unix)[25672]: session opened for user root by (uid=0)
1036803	jethro	info	2006-01-19	04:33:25	named[13918]: lame server resolving '150.129.158.67.in-addr.arpa' (in '129.158.67.in-addr.arpa?'): 12.23.66.134#53
1036802	jethro	info	2006-01-19	04:33:25	named[13918]: lame server resolving '150.129.158.67.in-addr.arpa' (in '129.158.67.in-addr.arpa?'): 12.23.66.134#53
1036801	jethro	info	2006-01-19	04:33:21	named[13918]: lame server resolving '150.129.158.67.in-addr.arpa' (in '129.158.67.in-addr.arpa?'): 12.23.66.134#53
1036800	jethro	info	2006-01-19	04:33:21	named[13918]: lame server resolving '150.129.158.67.in-addr.arpa' (in '129.158.67.in-addr.arpa?'): 12.23.66.134#53
1036799	jethro	info	2006-01-19	04:33:21	named[13918]: lame server resolving '124.120-29.84.161.212.in-addr.arpa' (in '120-29.84.161.212.in-addr.arpa?'): 212.74.78.33#53

Figure 2.10. Syslog Search Result

2.3.1.7 NESSUS Main. It shows two sub functions: 'running NESSUS' and 'report results'. Clicking a next button guides next steps.

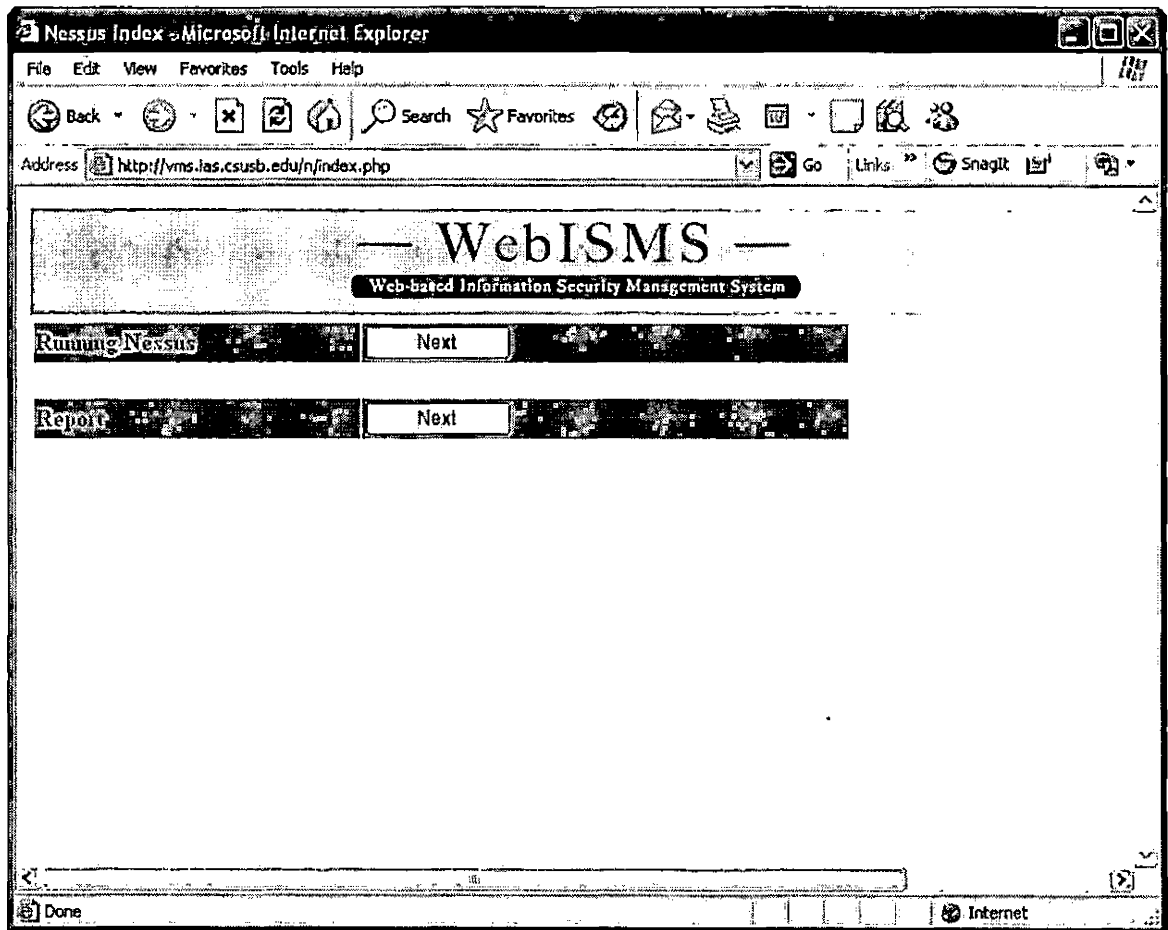


Figure 2.11. NESSUS Main Page

2.3.1.8 Creating a target file for NESSUS. This page guides the first step to run NESSUS that creates a target files. The user can enter a single IP or multiple IPs or just network IP with a netmask.

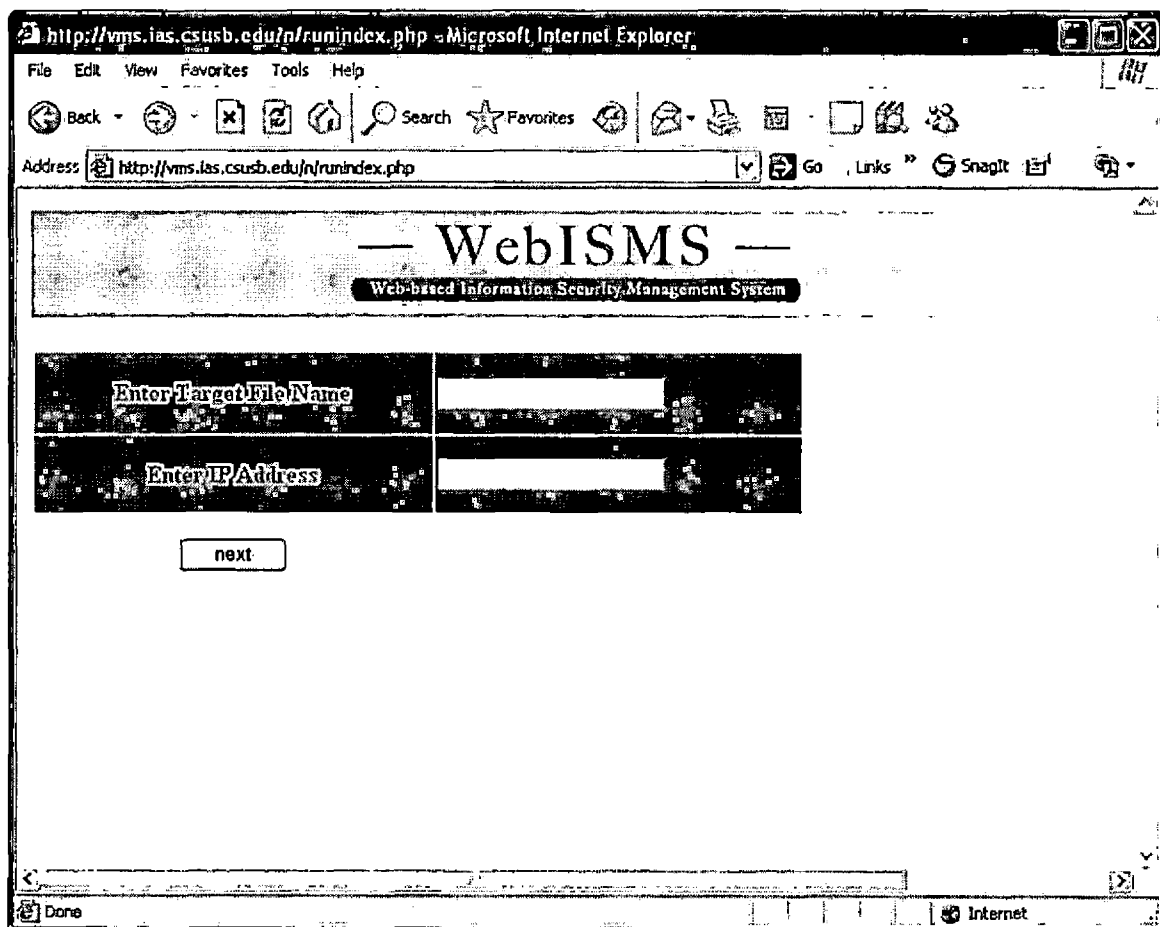


Figure 2.12. Creating a Target File

2.3.1.9 Creating a report file for NESSUS. This page shows whether the target file is generated successfully. If the right file is created, it requests a report file name. After typing in a file name and clicking a next button launches NESSUS processes.

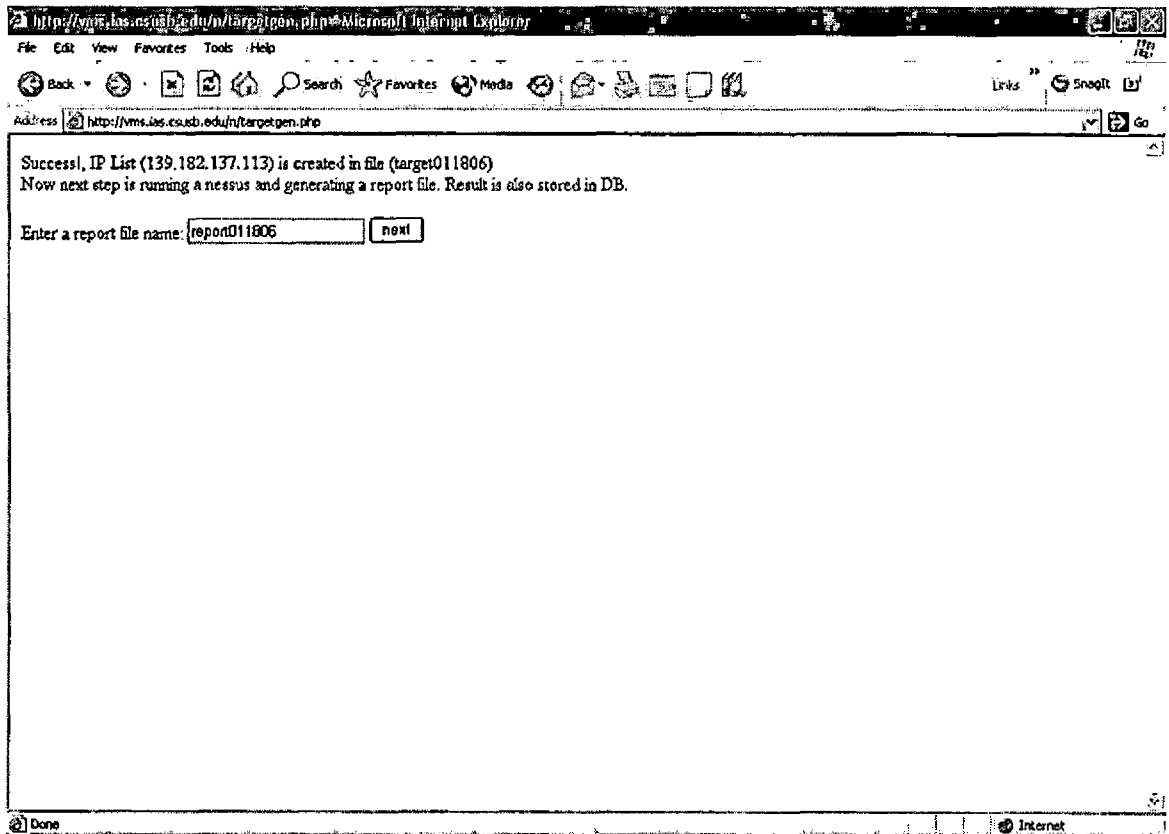


Figure 2.13. Creating a Report File

2.3.1.10 NESSUS result. If NESSUS scans a target machine successfully, this result page is generated. Clicking a highlighted link, 'click here to see the report' shows the detail results in HTML format.

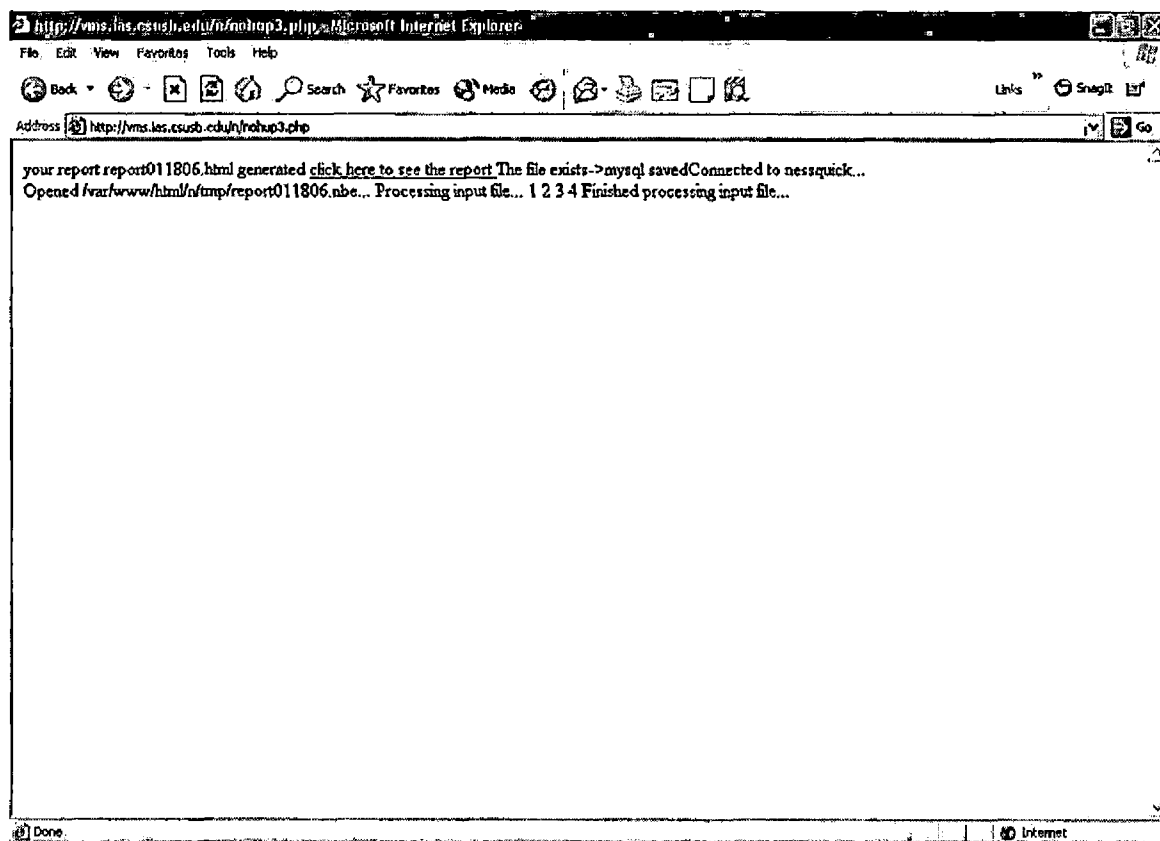


Figure 2.14. NESSUS Result

2.3.1.11 NESSUS detail result. After clicking a link, 'click to see the result' from the previous page, detail result of a target machine or network is appeared. The pages below show a sample output that shows very detail information about the target machine. The result page is a single page, but output is too long to capture it as a single page, so that this page is divided in three parts as below. The first part of this page shows IP and host name of the machine assessed followed by a list of ports scanned. In Figure 2.15, it shows a list of vulnerable applications with version number, port number, and a fixing suggestion.

Nessus Scan Report - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Search Favorites Media Links SnapIt Go

Address http://vms.las.csusb.edu/np130141.html

Nessus Scan Report

This report gives details on hosts that were tested and issues that were found. Please follow the recommended steps and procedures to eradicate these threats.

Scan Details	
Hosts which were alive and responding during test	1
Number of security holes found	0
Number of security warnings found	0

Host List	
Host(s)	Possible Issue
139.182.137.141	Security note(s) found
Return to top	

Analysis of Host		
Address of Host	Port/Service	Issue regarding Port
139.182.137.141	ssh (22/tcp)	Security notes found
139.182.137.141	finger (79/tcp)	Security notes found
139.182.137.141	ftp (20/tcp)	Security notes found
139.182.137.141	sunrpc (111/tcp)	Security notes found
139.182.137.141	squid-http (3128/tcp)	Security notes found
139.182.137.141	complex-link (5001/tcp)	Security notes found
139.182.137.141	ccetx (12000/tcp)	Security notes found
139.182.137.141	EBR (31337/tcp)	Security notes found
139.182.137.141	unknown (32769/tcp)	Security notes found
139.182.137.141	sunrpc (111/udp)	Security notes found
139.182.137.141	omad (32768/udp)	Security notes found
139.182.137.141	ntp (123/udp)	Security notes found
139.182.137.141	general/udp	Security notes found
139.182.137.141	general/tcp	Security notes found

Done Internet

(1)

Figure 2.15. NESSUS Report Page

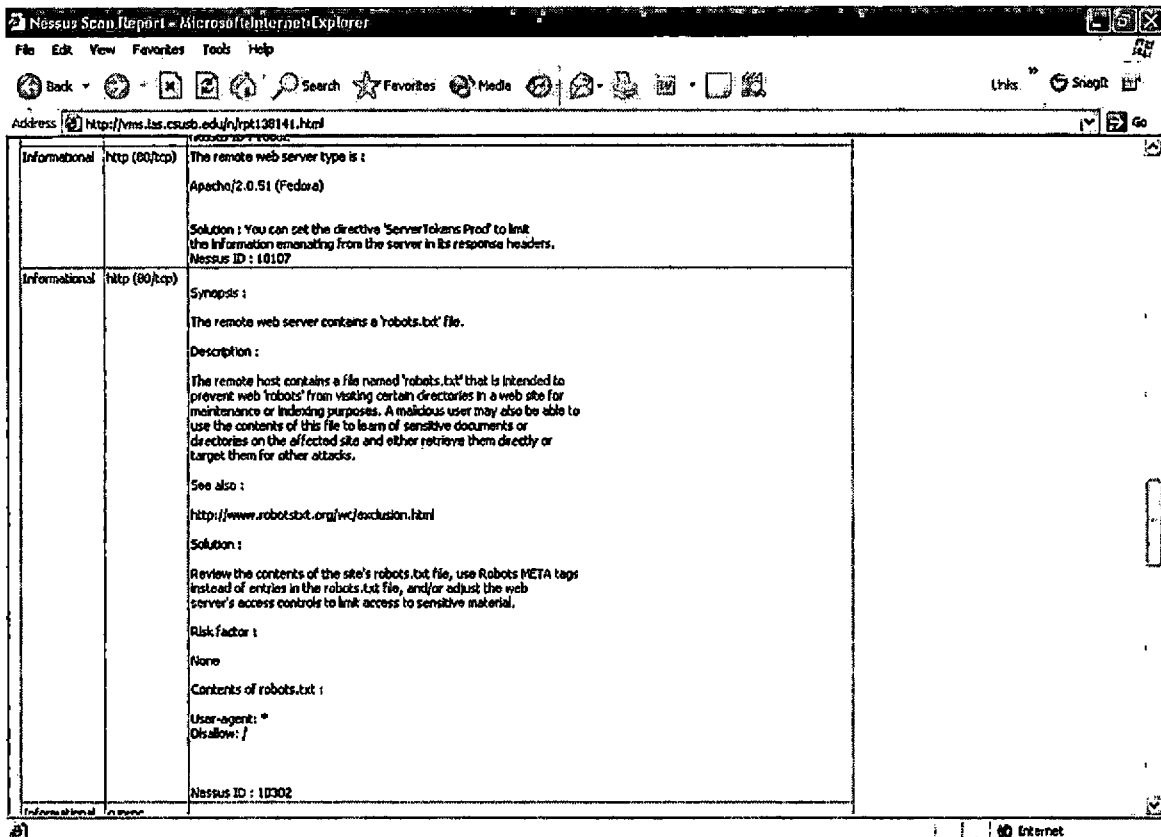
Nessus Scan Report - Microsoft Internet Explorer

Address: http://vms.lss.csusb.edu/rpt158141.html

Security Issues and Fixes: 139.102.137.141		
Type	Port	Issue and Fix
Informational	ssh (22/tcp)	An ssh server is running on this port. Nessus ID : 10300
Informational	ssh (22/tcp)	Remote SSH version : SSH-2.0-OpenSSH_3.9p1 Remote SSH supported authentication : publickey,keyboard-interactive Nessus ID : 10267
Informational	ssh (22/tcp)	The remote SSH daemon supports the following versions of the SSH protocol: - 1.99 - 2.0 SSHv2 host key fingerprint : cd:d4:f7:0b:aa:c1:75:09:fb:86:8e:f4:43:9d:09:96 Nessus ID : 10601
Informational	finger (79/tcp)	An unknown service is running on this port. It is usually reserved for Finger. Nessus ID : 10330
Informational	finger (79/tcp)	An unknown server is running on this port. If you know what it is, please send this banner to the Nessus team: 00: 54 68 09 73 20 64 61 63 68 69 6e 65 20 69 73 20 this machine is 10: 70 61 72 74 69 63 69 70 61 74 69 6e 67 20 69 6e participating in 20: 20 61 20 64 69 73 74 72 69 62 75 74 65 64 20 6e a distributed n 30: 65 74 77 6f 72 6b 2f 63 6f 64 70 75 74 69 6e 67 etwork/computing 40: 20 74 65 75 74 62 65 64 2e 0a 46 6f 72 20 66 75 tested.. For fu 50: 72 74 68 65 72 20 69 6e 66 6f 72 6d 61 74 69 6f rther informatio 60: 6e 2c 20 62 72 6f 77 73 65 20 74 68 65 20 77 65 n, browse the wo 70: 62 73 69 74 65 20 68 6f 73 74 65 64 20 6f 6a 20 bsite hosted on 80: 74 68 69 73 20 6e 6f 64 65 20 6f 72 20 76 69 73 this node or vis 90: 69 74 20 0a 68 74 74 70 3a 2f 2f 77 77 77 2e 70 r .http://www.p a0: 6c 61 6e 65 74 24 6c 61 62 2e 6f 72 67 2f 0a 0a lanet-lab.org... b0: 50 6c 65 61 73 65 20 64 69 72 65 63 74 20 61 6e Please direct an c0: 79 20 63 6f 6e 63 65 72 6e 73 2c 20 71 75 65 73 y concerns, ques d0: 74 69 6f 6e 73 2c 20 6f 72 20 63 6f 6d 6d 65 6e tions, or commen e0: 74 73 20 74 6f 3a 0a 73 75 70 70 6f 72 74 40 70 is bot.support@p f0: 6c 61 6e 65 74 2d 6c 61 62 2e 6f 72 67 0a 0a 54 lanet-lab.org...T 01: 68 61 6e 69 20 79 6f 75 2c 0a 0a 64 68 65 70 69 hand user. The 0

(2)

Figure 2.15. NESSUS Report Page (continued)



(3)

Figure 2.15. NESSUS Report Page (continued)

2.3.1.12 NESSUS Report Main. This page shows two options to view the results from NESSUS assessment scanning. First, clicking a button of 'show all of hosts with vulnerabilities' shows a list of machines with network security vulnerabilities. Second option shows a list of vulnerabilities of a machine that IP typed in the box.

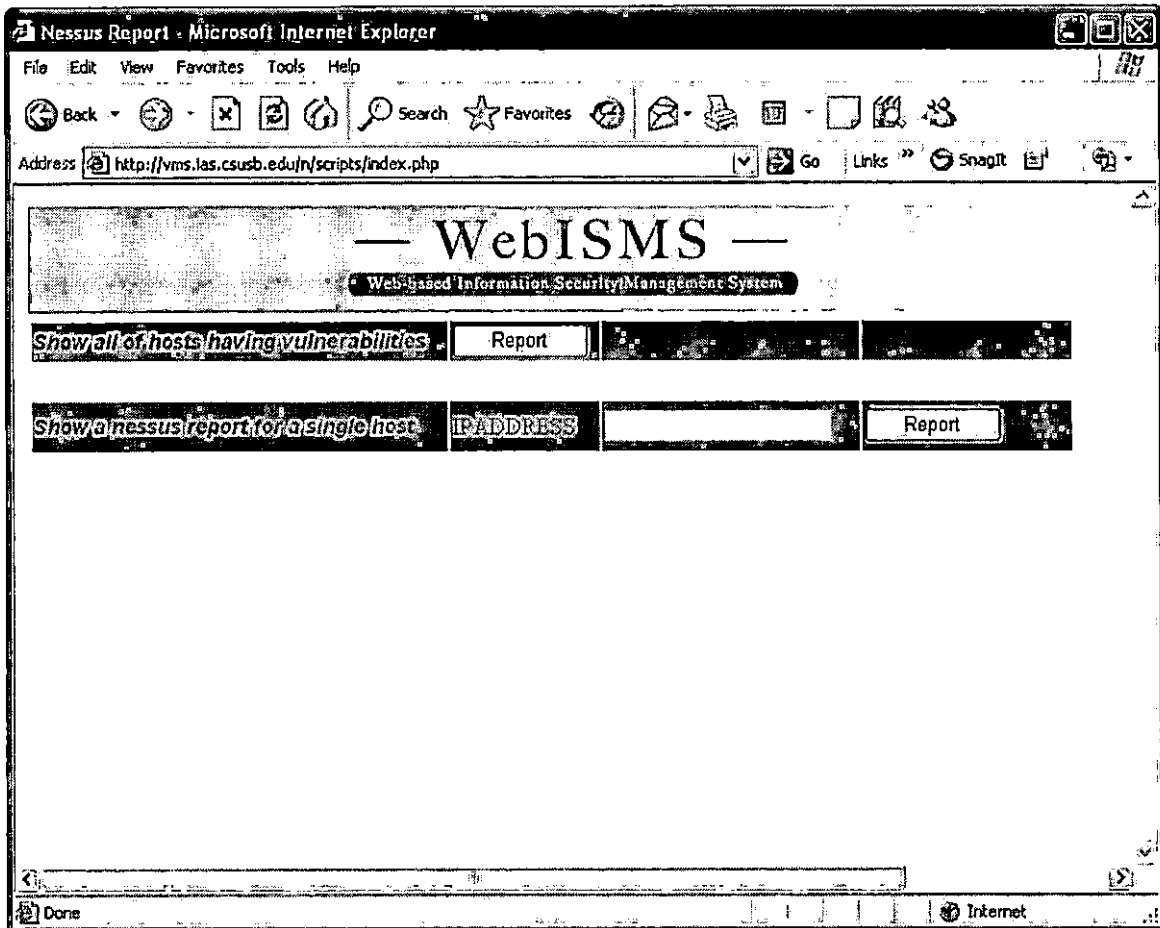


Figure 2.16. NESSUS Report Menu Page

2.3.1.13 List of Vulnerable Machines. After clicking "show all of hosts having vulnerabilities", this page generates a list of vulnerable machines. Red icon, next of IP of each machine is clickable. Clicking a red icon shows a detailed information about the machine.

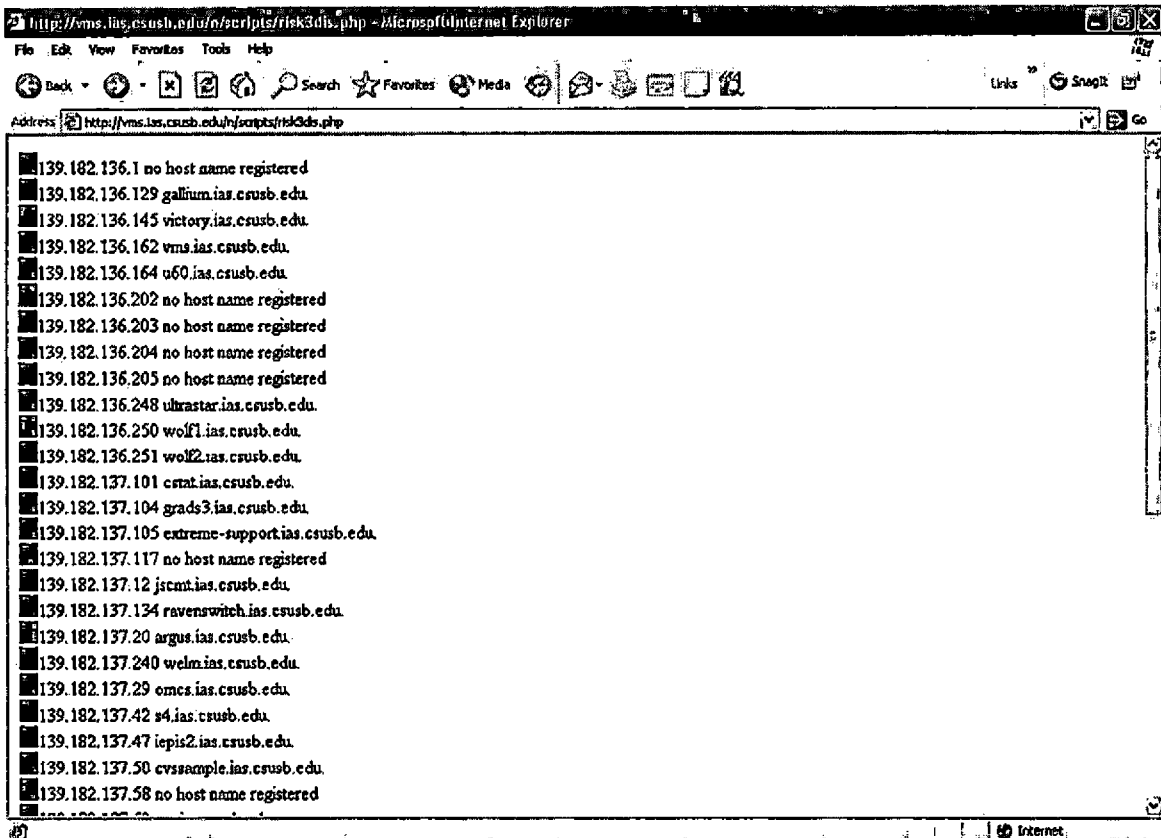


Figure 2.17. List of Vulnerable Machines

2.3.1.14 Result from Clicking a Icon of Previous Page.

This page shows a list of vulnerabilities of single machine from clicking a icon of the previous page.

host	msg	timestamp
139.182.139.140	Information about this scan : Nessus version : 2.2.5 Plugin feed version : 200510041915 Type of plugin feed : Registered (7 days delay) Scanner IP : 139.182.136.162 Port scanner(s) : nessus_tcp_scanner Port range : default Thorough tests : no Experimental tests : no Paranoia level : 1 Report Verbosity : 1 Safe checks : yes Scan Start Date : 2005/11/7 18:00 Scan duration : 1574 sec	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the DELETE method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : Medium BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the DELETE method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : Medium BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the PUT method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : High BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the DELETE method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : Medium BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the PUT method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : High BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the DELETE method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : Medium BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005
139.182.139.140	It seems that the PUT method is enabled on your web server Although we could not exploit this, you'd better disable it Solution : disable this method Risk factor : High BID : 12141 Other references : OWASP:OWASP-CM-001	Mon Nov 7 18:27:09 2005

Figure 2.18. Result Page from Clicking a Icon of the Previous Page

2.3.1.15 NMAP Main. This page shows two sub menu, 'running NMAP' and 'report'. Clicking a button of 'running NMAP' guides the user to the next page for running NMAP. 'report' option shows list of options to run a NMAP report.

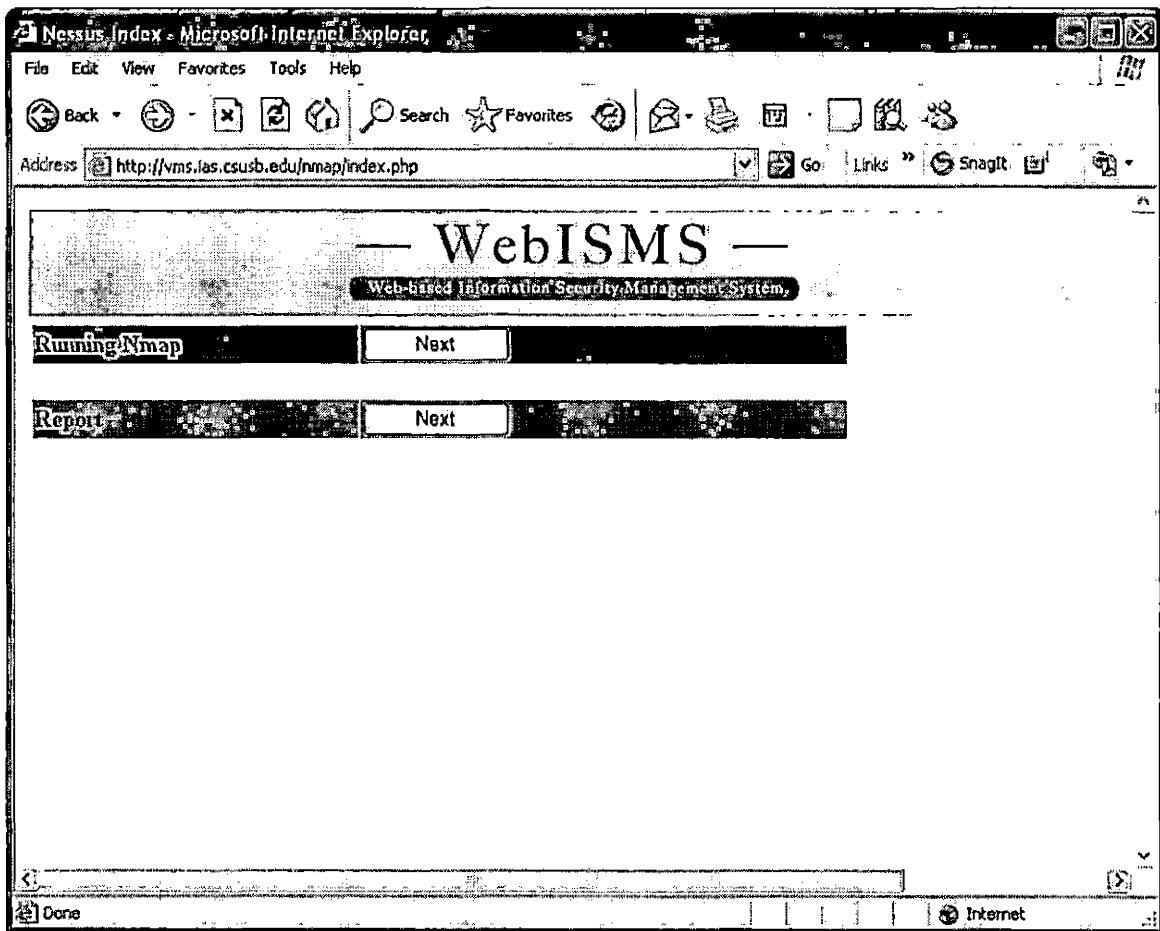


Figure 2.19. NMAP Main

2.3.1.16 Running NMAP. This page has input parameters to run NMAP. First, for scanning a single machine, a user just inputs a IP of target machine and a NMAP ID and click a next button. For scanning a whole network block, the user types in a NMAP ID and network IP.

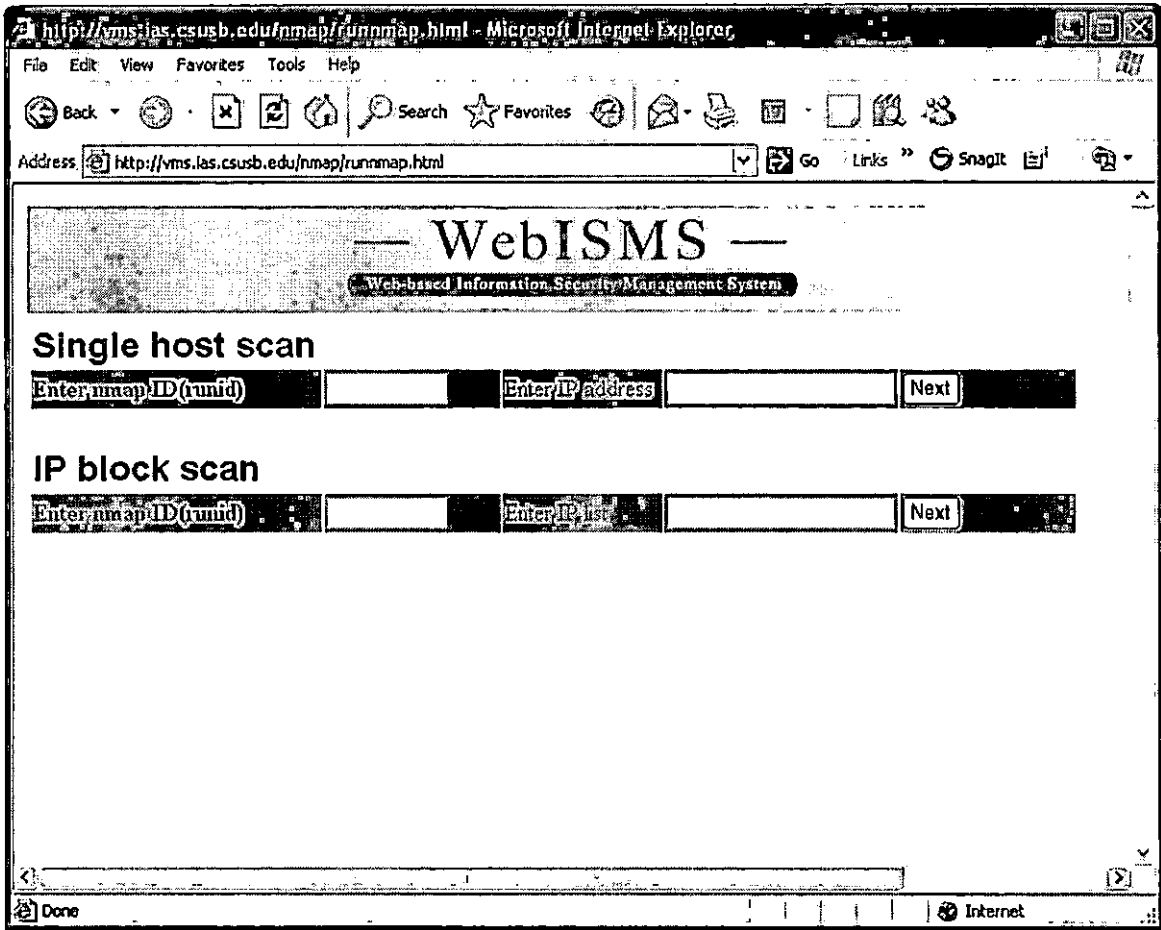


Figure 2.20. Run NMAP

2.3.1.17 NMAP Report Main. This page has four options to generate a NMAP report. First one is a link, 'click here to see open ports' that shows all the list of IP of machines scanned. Second one is a NMAP report for a single machine. Third one is based on the port number and the protocol type. Fourth one shows a list of NMAP RUN ID.

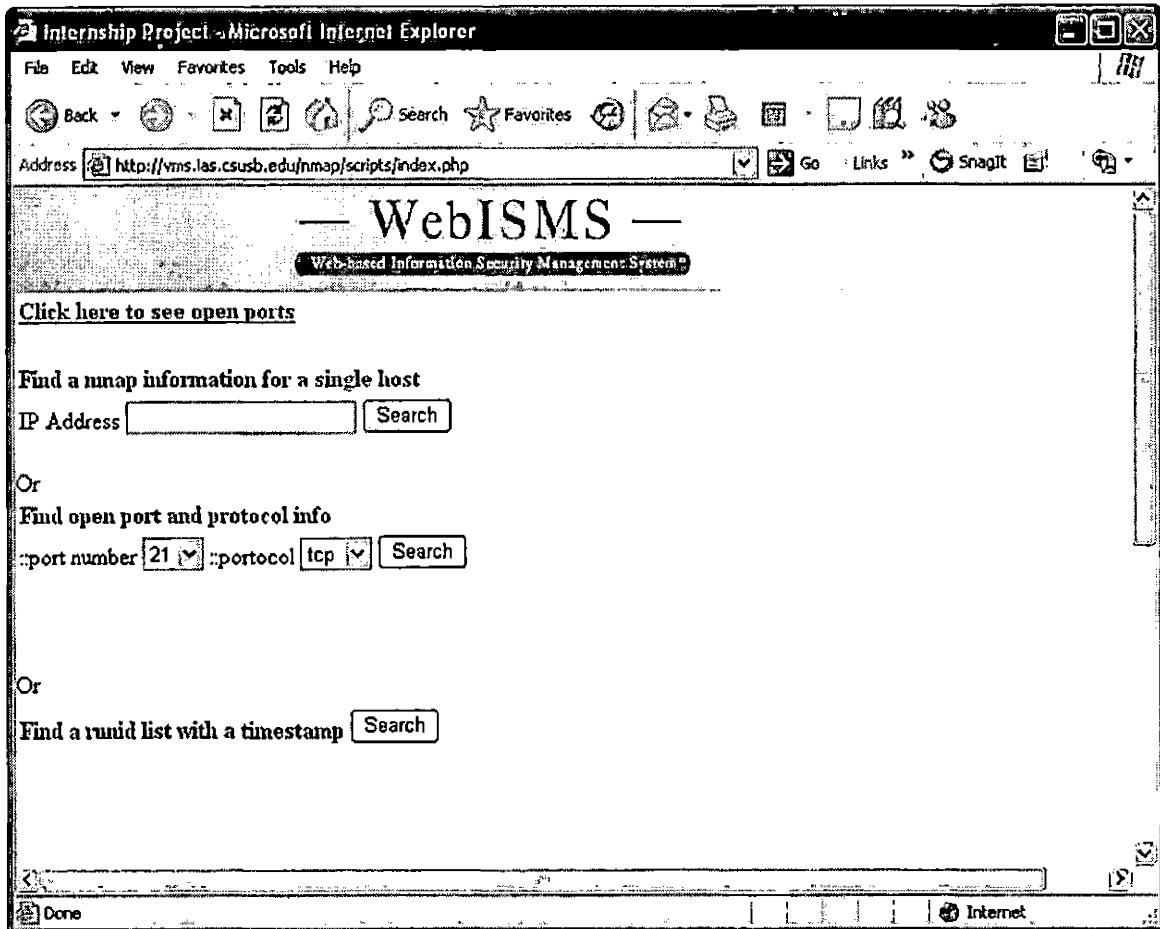


Figure 2.21. NMAP Report Main

2.3.1.18 List of machines scanned by NMAP. When the link, 'click here to see open ports' from the previous page, this page is generated. It shows all the list of port, protocol, service and IP with a red icon next to the IP. If the red icon is clicked, it shows a list of ports open for the machine.

Below is currently port open. Clicking a icon shows detail

Port	Protocol	Service	IP Address
7	tcp	echo	139.182.136.248 ultrastar.ias.csusb.edu.
7	tcp	echo	139.182.136.164 u60.ias.csusb.edu.
9	tcp	discard	139.182.136.248 ultrastar.ias.csusb.edu.
9	tcp	discard	139.182.136.164 u60.ias.csusb.edu.
13	tcp	daytime	139.182.136.248 ultrastar.ias.csusb.edu.
13	tcp	daytime	139.182.136.164 u60.ias.csusb.edu.
19	tcp	chargen	139.182.136.248 ultrastar.ias.csusb.edu.
19	tcp	chargen	139.182.136.164 u60.ias.csusb.edu.
21	tcp	ftp	139.182.136.164 u60.ias.csusb.edu.
21	tcp	ftp	139.182.136.251 wolf2.ias.csusb.edu.
21	tcp	ftp	139.182.137.241 pr1.ias.csusb.edu.
21	tcp	ftp	139.182.136.145 victory.ias.csusb.edu.
21	tcp	ftp	139.182.136.248 ultrastar.ias.csusb.edu.
22	tcp	ssh	139.182.137.19 mt2.ias.csusb.edu.
22	tcp	ssh	139.182.137.79 orans.ias.csusb.edu.

Figure 2.22 List of Machines Scanned

2.3.1.19 Output from Clicking a Icon of the Previous Page. In order to see more detail information of the port list from the previous page, simply clicking a icon next to the port number shows more detail information. It shows date, time, port number, IP of the machine.

d	t	port	protocol	state	target_ip
2005-09-18	17:59:15	22	tcp	open	139.182.137.50
2005-09-19	14:04:42	22	tcp	open	139.182.137.50
2005-10-06	12:15:44	22	tcp	open	139.182.137.50
2005-11-14	11:37:36	22	tcp	open	139.182.137.50

Figure 2.23. Scanned Result of Single Machine

2.3.1.20 Result from the Query Based on Port Number and Protocol Type. Output below shows the result based on port number and protocol type.

runid	d	t	target_ip	port	protocol	state
121	2005-09-18	17:38:41	127.0.0.1	80	tcp	open
136	2005-09-18	17:50:00	139.182.136.134	80	tcp	open
136	2005-09-18	17:50:00	139.182.136.158	80	tcp	open
136	2005-09-18	17:50:00	139.182.136.162	80	tcp	open
137	2005-09-18	17:59:15	139.182.136.134	80	tcp	open
137	2005-09-18	17:59:15	139.182.136.158	80	tcp	open
137	2005-09-18	17:59:15	139.182.136.162	80	tcp	open
137	2005-09-18	17:59:15	139.182.136.250	80	tcp	open
137	2005-09-18	17:59:15	139.182.136.251	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.19	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.50	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.134	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.140	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.141	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.240	80	tcp	open
137	2005-09-18	17:59:15	139.182.137.241	80	tcp	open
137	2005-09-18	17:59:15	139.182.139.100	80	tcp	open
137	2005-09-18	17:59:15	139.182.139.103	80	tcp	open
137	2005-09-18	17:59:15	139.182.139.140	80	tcp	open
2999	2005-09-19	14:01:45	139.182.0.148	80	tcp	open
3021	2005-09-19	14:04:42	139.182.136.134	80	tcp	open
3021	2005-09-19	14:04:42	139.182.136.158	80	tcp	open

Figure 2.24. Result Based on Port Number and Protocol Type

2.3.1.21 List of RunID. NMAP runID indicates the date and time of NMAP scanning. History of runID is necessary when any compromised machine found. Based on the list of runID, System Administrator is able to identify the port log of the machine.

runid	d	t
121	2005-09-18	17:38:41
136	2005-09-18	17:50:00
102	2005-09-18	17:58:15
137	2005-09-18	17:59:15
202	2005-09-19	13:36:34
202	2005-09-19	13:38:36
202	2005-09-19	13:40:03
203	2005-09-19	13:46:39
203	2005-09-19	13:49:51
204	2005-09-19	13:51:59
399	2005-09-19	14:01:03
2999	2005-09-19	14:01:45
3021	2005-09-19	14:04:42
45	2005-10-05	17:09:27
45	2005-10-05	17:09:52
567	2005-10-05	17:12:21
222243	2005-10-05	17:23:05
200	2005-10-05	17:25:21
789	2005-10-05	17:30:47
779	2005-10-05	17:35:19
1006	2005-10-06	08:37:56
10123	2005-10-06	12:15:44

Figure 2.25. List of RunID

2.3.2 Performance Requirements

Even though WebISMS is supporting multi-user logins, It is strongly recommended to use this system in single user mode because assessing tool, NESSUS and auditing tool, NMAP are memory and network bandwidth intensive programs. Multiple running of those tools at the same time will lag system performance seriously and might produce incorrect results.

2.3.3 Logical Database Requirements

WebISMS will store all the data in a DBMS, such as MySQL. There are four databases used for this project: login for user password, syslog for the monitoring module, NMAPlog for the auditing module, NESSUSquick for the assessment module.

2.3.4 Design Constraints

2.3.4.1 Standards Compliance. The data structures and algorithms will comply with those accepted in publicly available documents and texts.

2.3.5 Software System Attributes

2.3.5.1 Reliability. The server where WebISMS will be located is functional.

2.3.5.2 Availability. This project will be accessed to limited users such as system administrators or information security officers due to the confidential information that contains IP addresses, host names, list of vulnerable software, list of network ports serving.

2.3.5.3 Security. Even though this project is developed with open-source programs, the information collected from WebISMS is sensitive so that WebISMS should be placed in a secured area. WebISMS should be also used in intranet only, not exposed to internet.

2.3.5.4 Maintainability. The majority of WebISMS is implemented with PHP scripts and HTML codes, so any web development tools handling PHP and HTML like Macromedia Dreamweaver is easy way to maintain.

2.3.6.5 Portability. WebISMS will be designed to be operated from any web browsers with any Operating System platforms.

CHAPTER THREE

SOFTWARE DESIGN

3.1 Architecture Design

3.1.1 System Design

As a system administrator, using network security tools like NESSUS and NMAP is one of many routines that is run daily. However, while using those tools, there is always a feeling that something is missing. NMAP is running on UNIX Shell, while NESSUS is running on X-windows. Their result outputs are also one time snapshots. The idea of WebISMS project started from here. WebISMS must be drive network security tools from the web browser, and results are stored in the database.

The idea of developing a SYSLOG system began from two years ago, while having a meeting with my supervisor, he asked me to check the system log of servers more frequently. It was a very cumbersome and hectic job. There were twelve to fifteen servers, and login in and reading logs of each server took a huge amount of time. There is need for a centralized logging/monitoring system.

I have decided to use LAMP system for WebISMS. LAMP is the acronym for Linux/Apache/Mysql/PHP that is very

popular system for open-source community. Network security tools, NMAP and NESSUS and SYSLOG-NG are native programs from Linux so that they are very stable to run in this platform. Apache is also a popular Web server running on Linux. MySQL is handy and a fast database program which has many libraries of Apache. PHP is web programming language that is run as a module of Apache and has native MySQL libraries. These provide a very stable process and produce less glitch than any other web programming languages.

Figure 3.1 shows the architecture in class diagram of WebISMS. The class diagram shows the attributes and operations of each object and how they are related. WebISMS has nine classes. The detailed design section of each class will be explained in a later section. The function and purpose of each class is following:

- Login - head class of other classes. Login class has a CheckLogin function that check the user name and password. CheckLogin is also handling a Session Management that prevents from accessing other web pages directly

- Syslog - syslog creates a bar graphs that shows instant status of syslog client machines. Each bar is clickable that is able to query a detailed information. It also checks session access to sub classes
- View_24Hr_Result - syslog_search function is able to search a specific log by host, time, or specific text
- View_7days_Result - it is similar to syslog class that create a bar graph based on 7 days log
- NESSUS - it shows simple menu and checks session access to sub classes
- Run_Nessus - Target_Gen() function create a target file including a list of target machine IP. Process_Nessus() starts NESSUS program and ensures NESSUS process
- Report_Nessus - Data_Retrieve() function generates a report based on vulnerability or IP
- NMAP - it shows simple menu and checks session access to sub classes
- Run_Nmap - Pro_Nmap() function starts a NMAP process scanning a single IP or a whole network IP with ensuring a NMAP process running

- Report_Nmap - Result_View() generates a report based on the result NMAP scanned

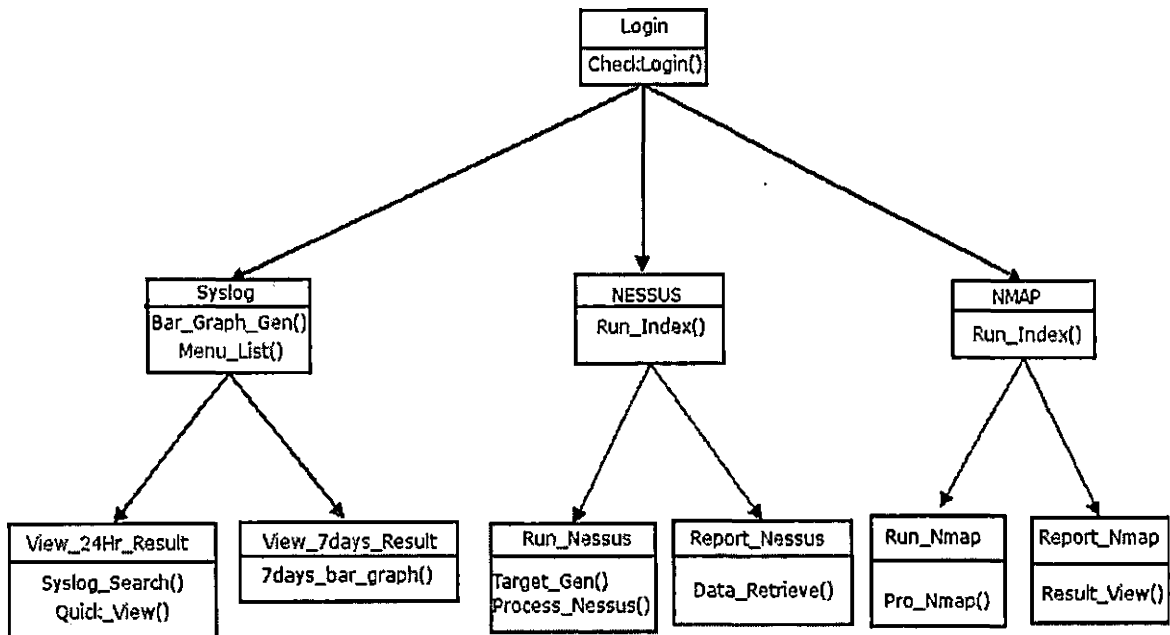


Figure 3.1 Architecture in Class Diagram of WebISMS

3.1.2 Database Design

The database has a crucial role in this project. The data generated by the modules, NESSUS, NMAP, and SYSLOG-NG are stored and retrieved in a database. The database design is shown in an Entity-Relationship Diagram. The ER Diagram is translated to a relational database schema. The

relational database schema shows all objects in ER Diagram as a series of related tables.

3.1.2.1 Entity-Relationship Diagram. The database design of WebISMS consists of four modules: login, syslog, nmaplog, and nessuslog. The ER diagrams show all entities, attributes, constraints, and relationships between entities.

- Login ER is for user management. The entity, users keep the user name and password for login.

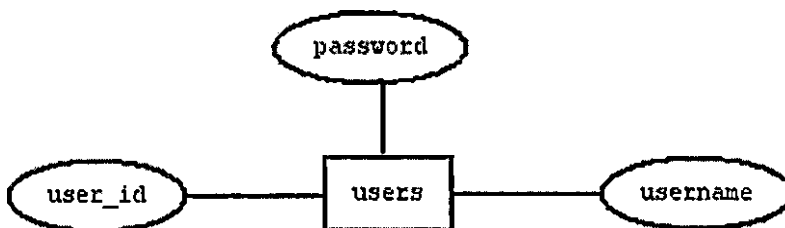


Figure 3.2 Login Entity-Relationship Diagram

- syslog ER is for syslog module. Attributes are created based on syslog template that consists of host, date, time, priority, program, and msg. Attribute, host is host name of client machine. Program is the program that log is created. Priority is risk

priority from 'alert' to 'critical' to 'error' to 'warning'. Msg is a text message part of log.

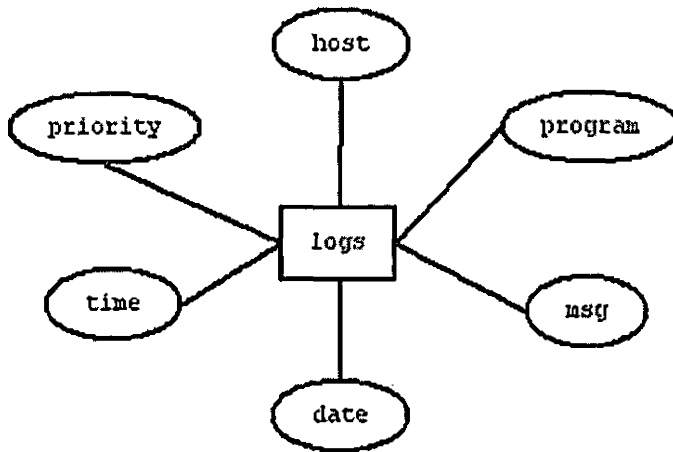


Figure 3.3 Syslog Entity-Relationship Diagram

- NESSUS ER is for NESSUS module. It has two entities, results and timestamp. Relationship between two entities is that timestamp belongs to results. They are one to one relation that when scanned result is created, timestamp is also created.

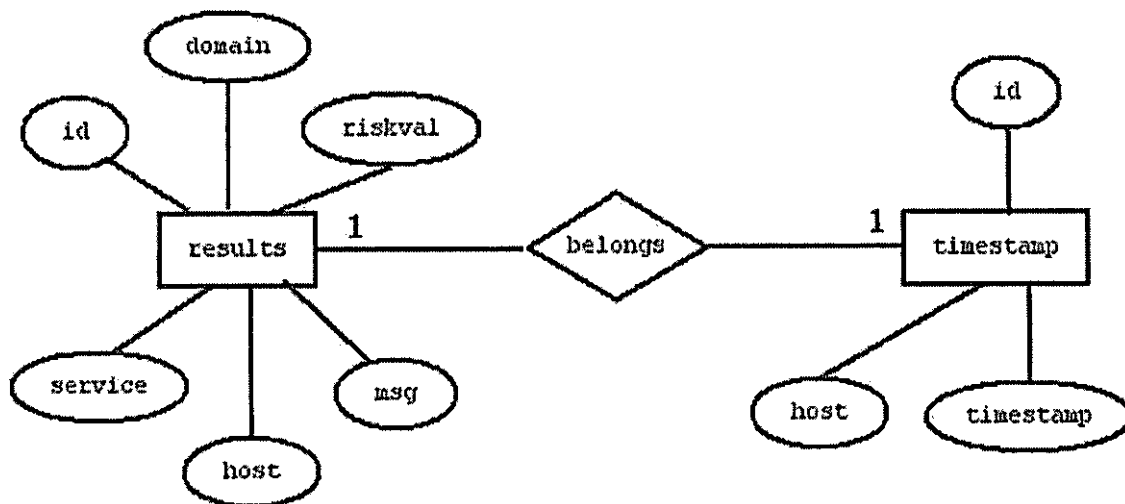


Figure 3.4 NESSUS Entity-Relationship Diagram

- NMAP ER has 4 entities: *hoststats*, *portstat*, *targets*, *runlist*. *Targets* contains information about the target host, including IP address, resolved hostname and OS guessed by NMAP. *Runlist* contains date and time information about each invocation of NMAP, including the host from which it was run. *Portstat* contains the port scan results. Each port reported by NMAP is recorded. *Hoststat* contains rudimentary statistics about the target host for each run of NMAP such as the total ports scanned and number of ports found open.

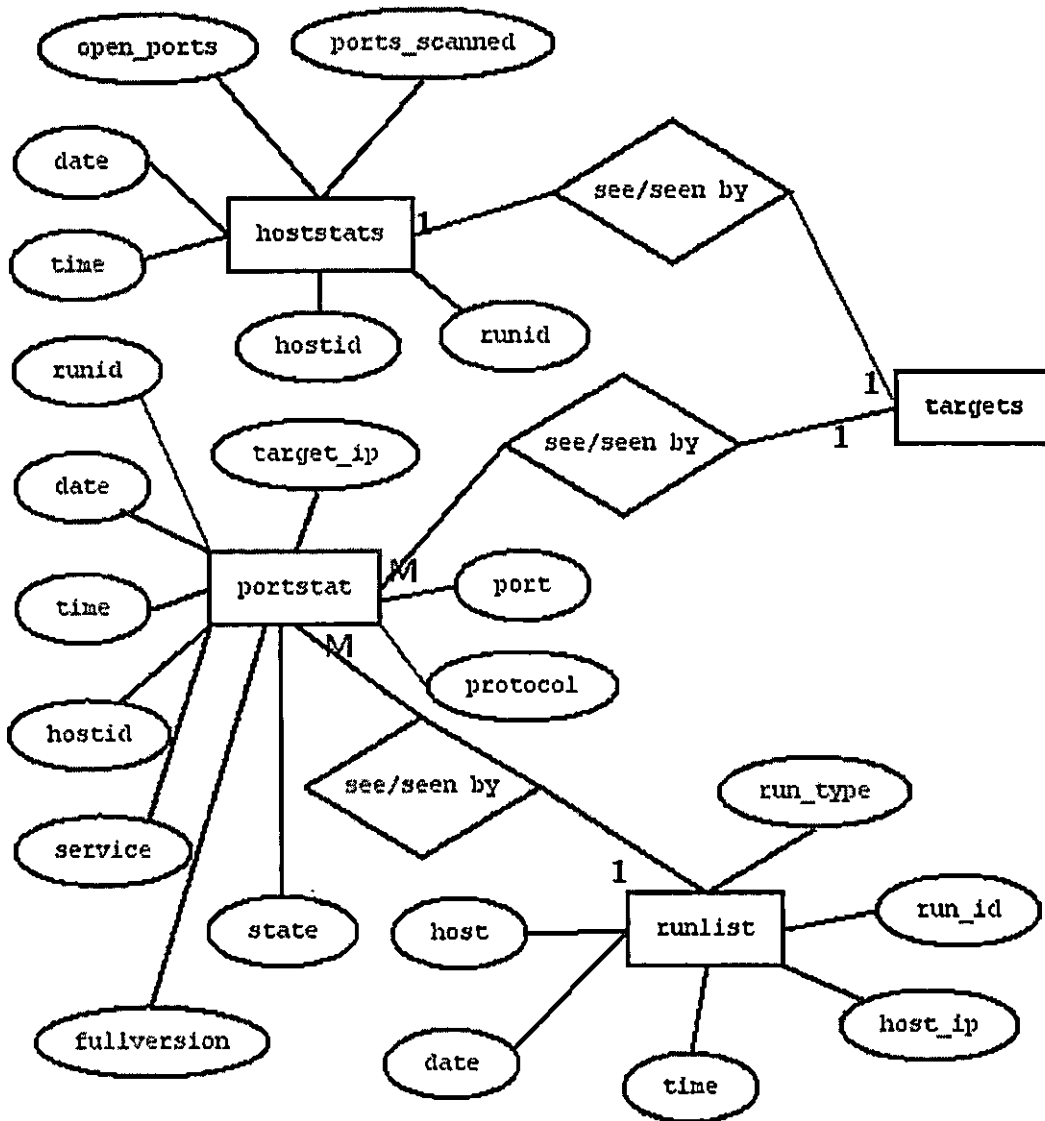


Figure 3.5 NMAP Entity-Relationship Diagram

3.1.2.2 Relational Database Tables. The relational database tables will be derived from the ER diagrams of login, syslog, nessus, and nmap.

3.1.2.2.1 login database. Login database is designed for user information of WebISMS. It has a single table, 'user' which has user ID, user name, and password.

Table 3.1 User Table

Field Name	Data Type	Description
user_id	int(8)	Identification number of each user. This number is incremental
Username	varchar(11)	This is login name of user.
password	varchar(32)	Password for user to login

3.1.2.2.2 syslog database. Syslog contains a single table, logs which consists of four fields. Each field is followed by the template in syslog-ng.conf.

Table 3.2. Logs Table

Field Name	Data Type	Description
host	varchar(32)	resolved host name of syslog client machine
program	varchar(24)	name of program that creates logs
priority	varchar(12)	risk level of log, named from 'alert', 'critical', 'error', and 'warning'
date	Date	date of log created
time	Time	time of log created
msg	varchar(200)	detailed message contained in a log

3.1.2.2.3 nessuslog database. NESSUS module uses nessuslog database which contains all of assessment result. Each time NESSUS scans, the result is saved in this database. Nessuslog consists of two tables, results and timestamps.

Table 3.4. Results Table

Field Name	Data Type	Description
Id	int(11)	ID of each assessment result
Domain	varchar(40)	Domain name of host when single host is scanned
Host	varchar(40)	host name of the machine scanned
Service	varchar(40)	service running on the machine
Riskal	tinyint(1)	only 3 numbers are used. 3 is high risk, 2 is middle, 1 is low risk
Msg	text	this field holds detail risk messages from NESSUS

Table 3.5. Timestamps Table

Id	int(11)	ID of each scan
Host	varchar(40)	host name of IP address scanned
Progress	varchar(40)	if same machine is scanned more than one time, this field shows progress description which is part of NESSUS output
Timestamp	varchar(40)	time stamp of each scan. Format is year-month-date hour:minute:second (0000-00-00 00:00:00)

3.1.2.2.4 nmaplog database. Nmaplog consists of four relational tables: target, runlist, portstat, hoststat. target has information about machines scanned. hoststat contains results of NMAP scan based on the host. portstat holds scanned results based on the port. runlist is time stamp information of each scan time.

Table 3.6. Hoststats Table

Field Name	Data Type	Description
Hosted	int(6)	ID of each host
Ip	varchar(16)	IP address of a host
User	varchar(32)	username of NMAP scan
D	date	scan date
T	time	scan time
open_ports	int(6)	open port number
Runid	int(6)	ID of each scan
ports_scanned	int(6)	number of ports scanned

Table 3.7. Portstats Table

Field Name	Data Type	Description
D	Date	scan date
T	Time	scan time
Hosted	int(6)	ID of host
target_ip	varchar(16)	IP address of host scanned
Port	int(6)	port number scanned
Protocol	varchar(10)	protocol type
Service	varchar(10)	service type
State	varchar(20)	state of a port scanned
fullversion	varchar(128)	version of Operating System of host scanned
Runid	int(10)	NESSUS scan ID based on the time scanned

Table 3.8. Targets Table

Field Name	Data Type	Description
Host	Varchar(64)	name of host
D	Date	date
T	Time	time
Ip	Varchar(20)	IP address
Mac	Varchar(32)	MAC address
Macvendor	Varchar(48)	network card vendor based on MAC address
Hosted	Varchar(6)	ID of host scanned
os_guessed	Varchar(100)	Operating System guessed by NMAP
os_known	Varchar(100)	Operating System found by NMAP
fingerprint	Varchar(254)	target machine's signature

Table 3.9. Runlist Table

Field Name	Data Type	Description
D	date	date
T	time	time
Runid	int(10)	ID of NMAP scan
host_ip	varchar(15)	IP address of host scanned
Host	varchar(64)	name of host

3.2 Detailed Design

This section describes the detailed design as pseudo-code algorithms of all classes shown in Figure 3.1.

3.2.1 Login Class

Login class handles the user login/password functions. It also manages sessions for the rest of the Web pages so that any other Web pages would not be accessed directly.

```

Class name: Login
Purpose: check the user login
Begin class

Session start

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function checkLogin
Begin
    Execute SQL statement
    If user is in database
        Go to next page
    Else
        Go to index page
End checkLogin

End class

```

Figure 3.6. Login Class

3.2.2 Syslog Class

This is the base module class for syslog. The major function of this class is creating a bar chart which express the log status of machines and the other is menu function which shows logs instantly. Logic of this pseudo-code is as following:

1. check the session that the access to this page is from
Login class

2. If the access is right, go on to the next function,
otherwise go back to the login class
3. connect to the database
4. query logs based on priority and host name
5. sort and save them in the priority array and host
array
6. generate a bar-chart that y-axis is number of logs and
x-axis is host names
7. using JpGraph libraries, SQL query statement is linked
to each bars to be able to be queried by clicking a
bar
8. at the side of the bar-chart, menu list is shown

```

Class name: Syslog
Purpose: render a bar chart for the number of logs of each
servers
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

/* generate a bar chart */
Function Bar_Graph_Gen
Begin
Execute SQL for counting logs based on priority
Create bar graph /* y-axis is number of logs x-axis is
host name */
End

/* menu statements */
Function Menu_List
Execute SQL for ssh logs OR Execute SQL for system errors
End

End class

```

Figure 3.7. Syslog Class

3.2.3 View24Hr Result Class

Querying the data from database is a major function of this class: search engine style and instant queries. The logic of pseudo-code is as following:

1. check the session that the access is from syslog class

2. if the access is permitted, go on to the next function,
otherwise stop and go back to login class
3. establish a connection to mysql
4. query logs from the database based on time, date, risk
priority, hostname, or any combination
5. syslog report generated based on the query

```

Class name: View24Hr_Result
Purpose: view logs last 24 hours by each different
conditions
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Syslog_Search
Begin
    If time based
        Execute SQL search by time
    If date based
        Execute SQL search by date
    If priority based
        Execute SQL search by priority
    Else any combination of above
        Execute SQL
End

Function Quick_View()
Begin
    Quick_View_List
End

End class

```

Figure 3.8. View24Hr_Result

3.2.4 View 7days Result Class

This class is similar to View24Hr_Result class, but the query is based on 7 day logs. Logic of pseudo-code is same as syslog class.

```
Class name: View_7days_Result
Purpose: render 7 days bar chart similar to 24 hours
Begin: class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function 7days_bar_graph
    Access 7 days database
    Sort the logs by priority
    Save logs in cache
    Call bar_chart library from JpGraphp
    Render a bar chart for 7 days logs
End

End class
```

Figure 3.9. View_7days_Result

3.2.5 NESSUS Class

This is the base class of NESSUS module. It has direction to either Running NESSUS or Generating a report. Logic of pseudo-code is as following:

1. check the session that the access is from login class,
otherwise go back to login class
2. establish a connection to MySQL
3. show a menu

```
Class name: NESSUS
Purpose: to display nessus menu
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Run_Index
/* print menu */
Begin
    Print menu
End

End class
```

Figure 3.10. NESSUS

3.2.6 Report Nessus Class

Query functions are here to generate a report.

Logic of pseudo-code is as following:

1. check the session that the access is from NESSUS class,
otherwise go back to login class
2. establish a connection to mysql
3. report is generated based if a single IP is typed in
the text box
4. report is generated if a network IP is typed in the
text box
5. if a vulnerable list is clicked, a web page based on
the vulnerable IP list is generated

```

Class name: Report_Nessus
Purpose: to display nessus result
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Data_Retrieve
Begin
    If a single IP address taken
        Retrieve the result of the single IP
    If a whole IP block taken
        Retrieve the result of the whole block IP
    Else
        Retrieve the result of vulnerable IP
End
End class

```

Figure 3.11. Report_Nessus

3.2.7 Run Nessus Class

Run_Nessus class contains functions to generate a NESSUS process. Target_Gen create a file that contains target IP address list. Process_Nessus start the NESSUS with options. Logic of pseudo-code is as following:

1. check the session that the access is from NESSUS class,
otherwise go back to login class
2. establish a connection to MySQL

3. create a target file for NISSUS process
4. if a target file is created, start a NISSUS scanning process
5. if the process is done, generate a report in HTML format and save the result in the database

```

Class name: Run_Nessus
Purpose: to run nessus
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Target_Gen
Begin
    If list of IP address is legitimate
        Create a file and save IP addresses
    Else
        Go back to Menu page
End

End class

Function Process_Nessus
Begin
    If target file is created
        Run the nessus within nohup and background
        /* once nessus process is started, */
        /* it is not dependent on the web browser process */

        by called mysql class, save the result in the database
    Else
        Go back to the Target_Gen() page
End

End class

```

Figure 3.12. Run_Nessus

3.2.8 NMAP Class

Main class of NMAP process. Logic of pseudo-class is as following:

1. check the session that the access is from login class, otherwise go back to login class
2. establish a connection to the database
3. print a menu

```
Class name: NMAP
Purpose: to display nmap menu
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Run_Index
/* print menu */
Begin
    Print menu
End

End class
```

Figure 3.13. NMAP

3.2.9 Run Nmap Class

This class handles a NMAP process with options. Logic of pseudo-code is as following:

1. check the session that the access is from NMAP class, otherwise go back to login class
2. establish a connection to the database
3. if a legitimate IP address is typed in, start a NMAP scan process, otherwise go back to the previous page
4. if NMAP process is done, generate a report in HTML format and save the result in the database

```

Class name: Run_Nmap
Purpose: to run nmap
Begin class

Check session

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Pro_Nmap
Begin
    If legitimate IP address is taken
        Start nmap process with nohup and background
    Else
        Go back to Run_index page
End

End class

```

Figure 3.14. Run_Nmap

3.2.10 Report Nmap Class

Querying and generating a report from the results of NMAP scan. Logic of pseudo-code is as following:

1. check the session that the access is from NMAP class, otherwise go back to login class
2. if a single IP address is requested, generate a report for the IP

3. if a whole block IP address is requested, generate a report for the block IP address
4. if only a list of vulnerable IP addresses is queried, generate a report for the list of vulnerable IP addresses

```
Class name: Report_Nmap
Purpose: to display nmap result
Begin class

/* connect to mysql database */
Function mysql_connect:
Begin
    Establish connection to the mysql
End

Function Result_View
Begin
    If a single IP address taken
        Retrieve the result of the single IP
    If a whole IP block taken
        Retrieve the result of the whole block IP
    Else
        Retrieve the result of vulnerable IP
End
End class
```

Figure 3.15. Report_Nmap

CHAPTER FOUR

SOFTWARE QUALITY ASSURANCE

4.1 Introduction

This chapter documents the software validation testing process for WebISMS. The purpose of software validation test is to guaranty the quality of software and its functionalities. Three testing processes are used to assure WebISMS software quality: unit testing, integration testing, and system acceptance testing.

4.2 Unit Testing

Unit testing greatly improved the quality of WebISMS. It also accelerated the development of this project, since unit testing allowed individual modules to be tested before the entire program was completed. Table 4.1 shows the result of unit testing for WebISMS.

Table 4.1. Unit Testing Results

Page	Unit	Tests Performed	Result
Login	Login	Test if only authenticated users access WebISMS	pass
	Buttons	Ensure all buttons work as expected	pass
Main menu	Links	Ensure all links work as expected	pass
Syslog main	Bar chart	Verify rendering all bars as counted from SQL queries	Pass
	Bar click	Verify clicking each bar in Chart shows detailed information correctly	pass
	Menu list	Verify clicking each menu goes to right page	pass
Syslog search	Scroll down menu	Verify all scroll down menu are displayed and work as expected	Pass
	Search message input form	Verify input field for search message works correctly	pass
NESSUS main	Buttons	Ensure all buttons work correctly	Pass
Running Ness	Input forms	Ensure whether input forms take values correctly	Pass
	Button	Ensure the button works correctly	pass

Table 4.1 Unit Testing Results (continued)

Generating Report	Input form	Ensure whether input form takes a correct value	pass
	Button	Verify whether click a button direct to the next page	pass
NESSUS report	Output format	Verify the report page has correct format	pass
NMAP main	Buttons	Ensure whether the buttons working correctly	pass
Running NMAP	Input forms	Ensure whether input forms take a correct value	Pass
	Button	Verify whether click a button direct to the next page	pass
NMAP report	Input forms	Ensure whether input forms take a correct value	pass

4.3 Integration Testing

This section shows sample runs of WebISMS showing all the functions working correctly including exception handling.

4.3.1 Login

This is main page of WebISMS. If incorrect username or password is entered, Exception handling page is appeared.

If username and password are accepted, it goes to the main menu page of the modules.

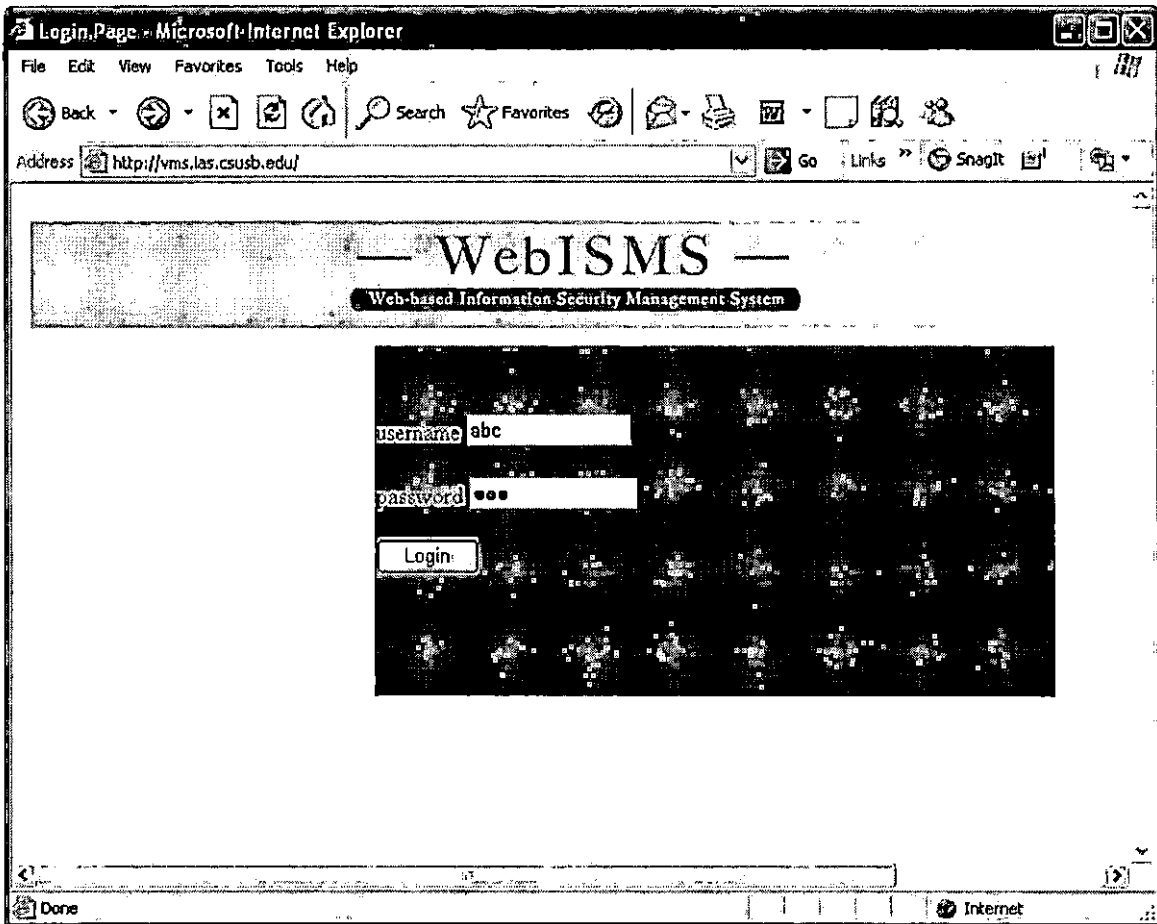


Figure 4.1. Demonstration of Main Page

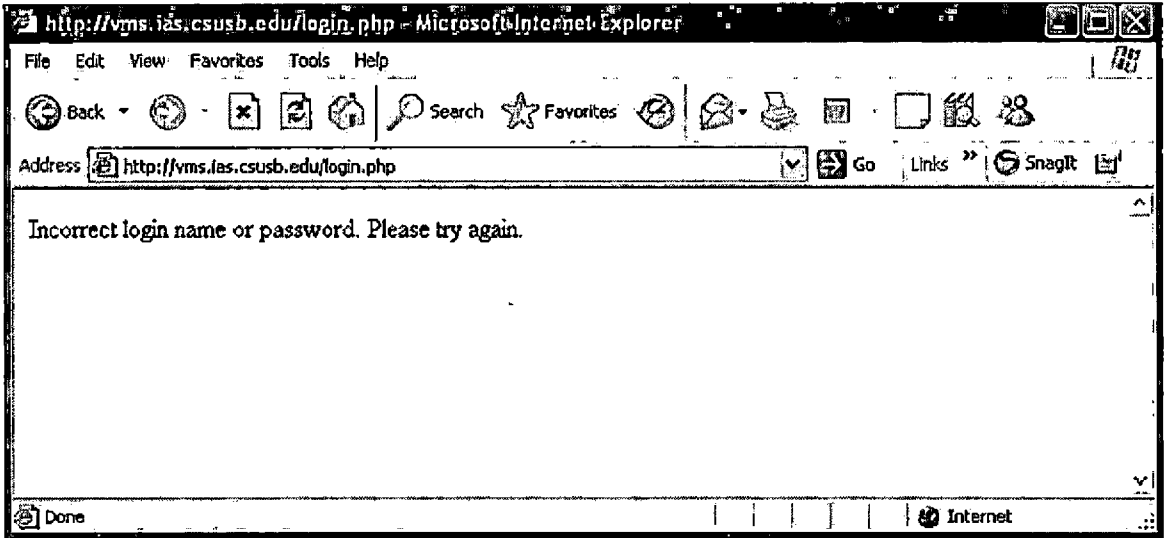


Figure 4.2. Exception Handling Page of Login Error

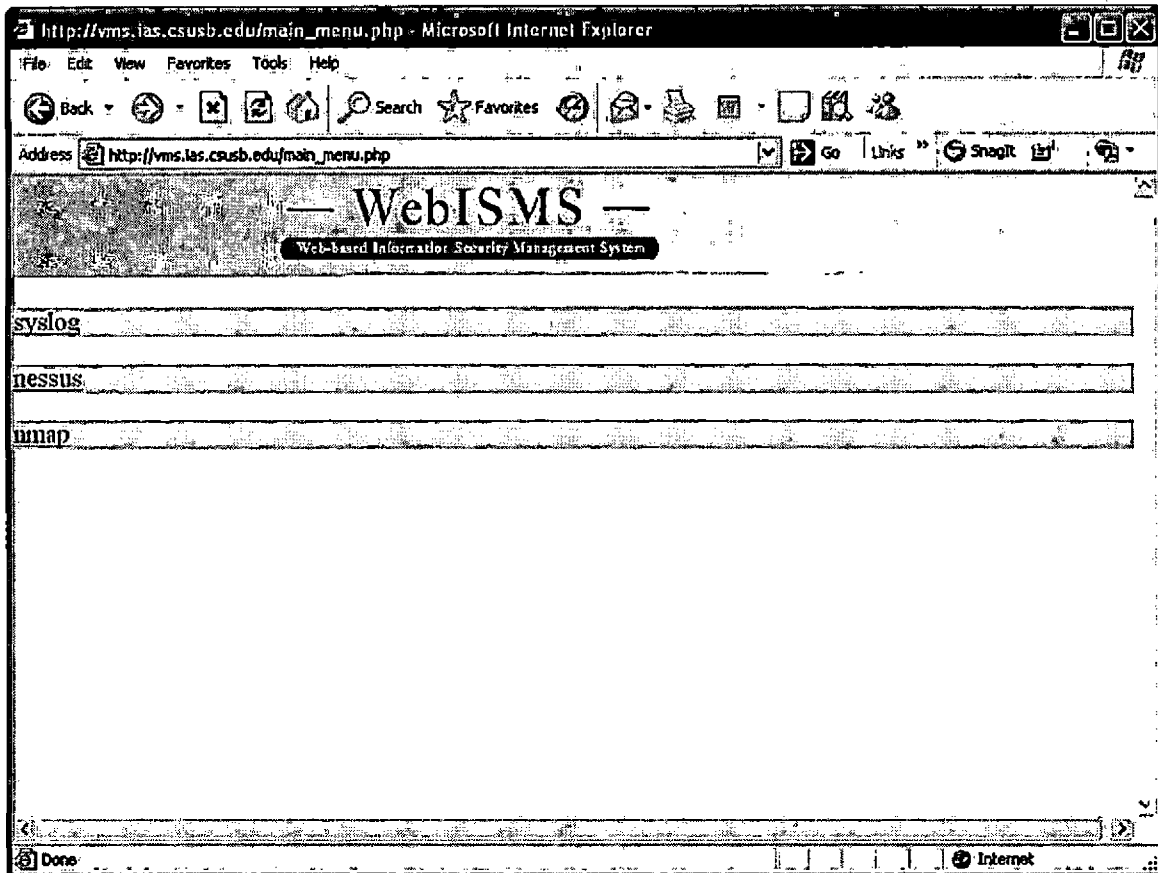


Figure 4.3. Module Menu Page

4.3.2 Syslog Module

Syslog module is demonstrated as for integration testing. Figure 4.1 shows a main page of syslog module. Bar graphs shows status of syslog clients' logs within 24 hours. Each client machine could have up to 4 bars depending on the priority of log. For example, in Figure 4.4, syslog client machine, zzyzx has 2 alert logs with a

red bar, and none of critical, error, or warning log. Other client, gimme has 2 warning logs with a yellow bar, jethro has 33 error logs in a brown bar. Furthermore, each bar is also clickable. If it is clicked, it shows detailed log information as in Figure 4.2-2. In this case, alert bar of zzyzx is clicked, and detailed log page, Figure 4.5 is rendered. Detail log information is tabled as hostname, risk priority, date, time, application program logged, and detailed message about log. The main page also shows a current query time at the top of page and most frequent searched queries are in the menu list.

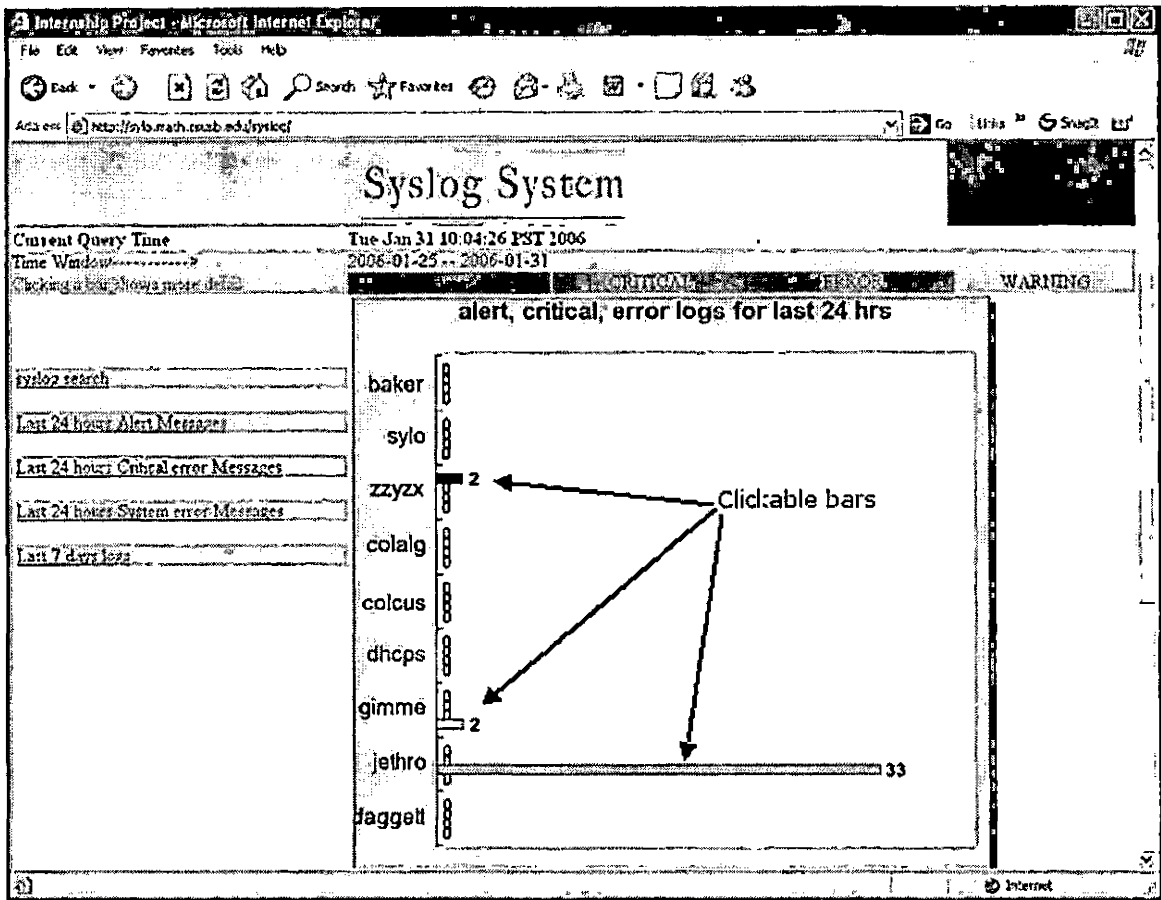


Figure 4.4. Syslog Main Page

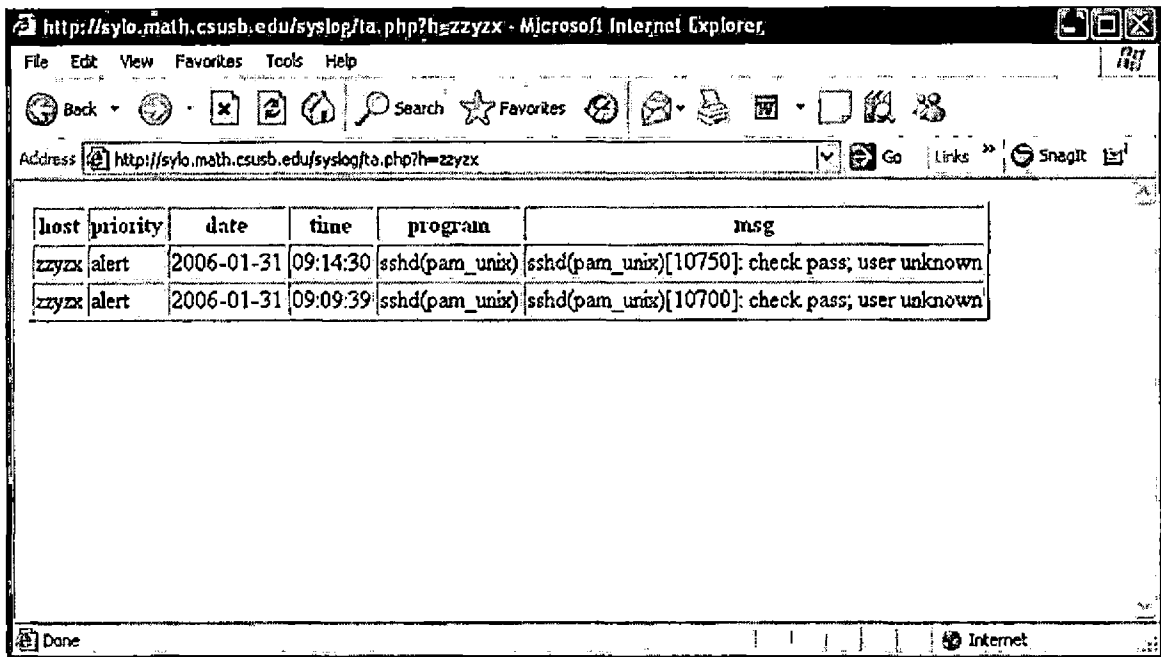


Figure 4.5. Demonstration of zzyzx Bar Click

4.3.2.1 Syslog Search. Syslog search is a sub function of Syslog module. Figure 4.3-1 shows a simple demonstration of syslog search menu. In host, hostname zzyzx is chosen, date is from 2006-01-31 to 2006-01-31, Time is open, Priority is alert. Result is in Figure 4.6 which is a same result in Figure 4.5.

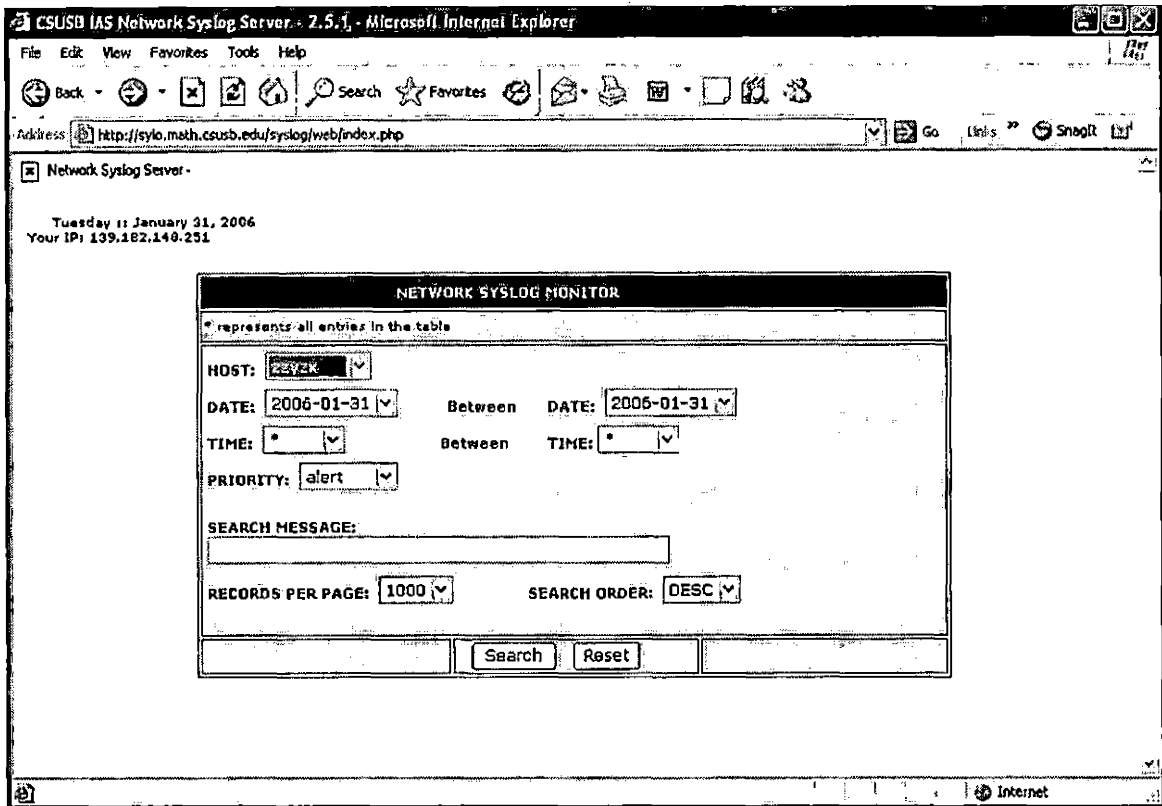


Figure 4.6. Demonstration of Syslog Search

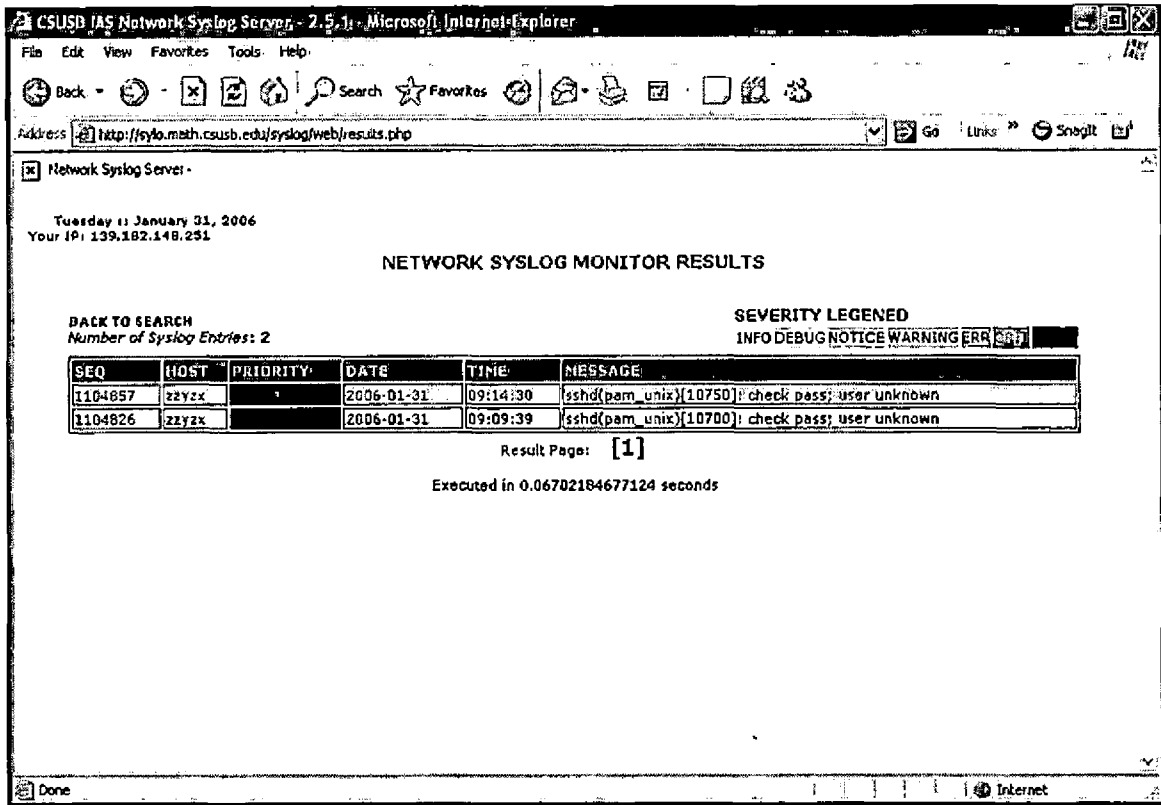


Figure 4.7. Result of Syslog Search Demonstration

4.3.2.2 Demonstration of menu click of Syslog Main page. Screen-Shots below are demonstration of clicking each list of main menu in syslog main page.

host	priority	date	time	program	msg
rzyzx	alert	2006-01-31	09:14:30	sshd(pam_unix)	sshd(pam_unix)[10750]: check pass; user unknown
rzyzx	alert	2006-01-31	09:09:39	sshd(pam_unix)	sshd(pam_unix)[10700]: check pass; user unknown

Figure 4.8. Last 24 Hours Alert Messages

host	priority	date	time	program	msg
daggett	crit	2006-01-31	10:50:17	sendmail	sendmail[29843]: k0VIQHhR029843: SYSERR(root): collect I/O error on connection from adsl-69-222-6-132.dsl.chcgil.ameritech.net, from=<paypal@email.paypal.com>

Figure 4.9. Last 24 Hours Critical Messages

host	priority	date	time	program	msg
jethro	err	2006-01-30	09:44:27	named	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied
jethro	err	2006-01-30	09:44:27	named	named[13918]: dumping master file: tmp-XXXXpT7YjK: open: permission denied
jethro	err	2006-01-30	09:01:01	named	named[13918]: client 139.182.155.47#1065: update '155.182.139.IN-ADDR.ARPA/IN' denied
jethro	err	2006-01-30	07:55:01	named	named[13918]: client 139.182.155.45#1068: update '155.182.139.IN-ADDR.ARPA/IN' denied
jethro	err	2006-01-30	07:32:24	named	named[13918]: transfer of 'csusb.edu/IN' from 139.182.2.1#53: failed while receiving responses: multiple RRs of singleton type
jethro	err	2006-01-30	03:57:12	named	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied
jethro	err	2006-01-30	03:57:12	named	named[13918]: dumping master file: tmp-XXXX0cH0mB: open: permission denied
jethro	err	2006-01-30	02:06:40	named	named[13918]: transfer of 'csusb.edu/IN' from 139.182.2.1#53: failed while receiving responses: multiple RRs of singleton type
jethro	err	2006-01-31	12:47:38	named	named[13918]: client 139.182.155.47#1740: update '155.182.139.IN-ADDR.ARPA/IN' denied
jethro	err	2006-01-31	12:00:45	named	named[13918]: transfer of '182.139.IN-ADDR.ARPA/IN' from 139.182.2.1#53: failed while receiving responses: permission denied

Figure 4.10. Last 24 Hours System Error Messages

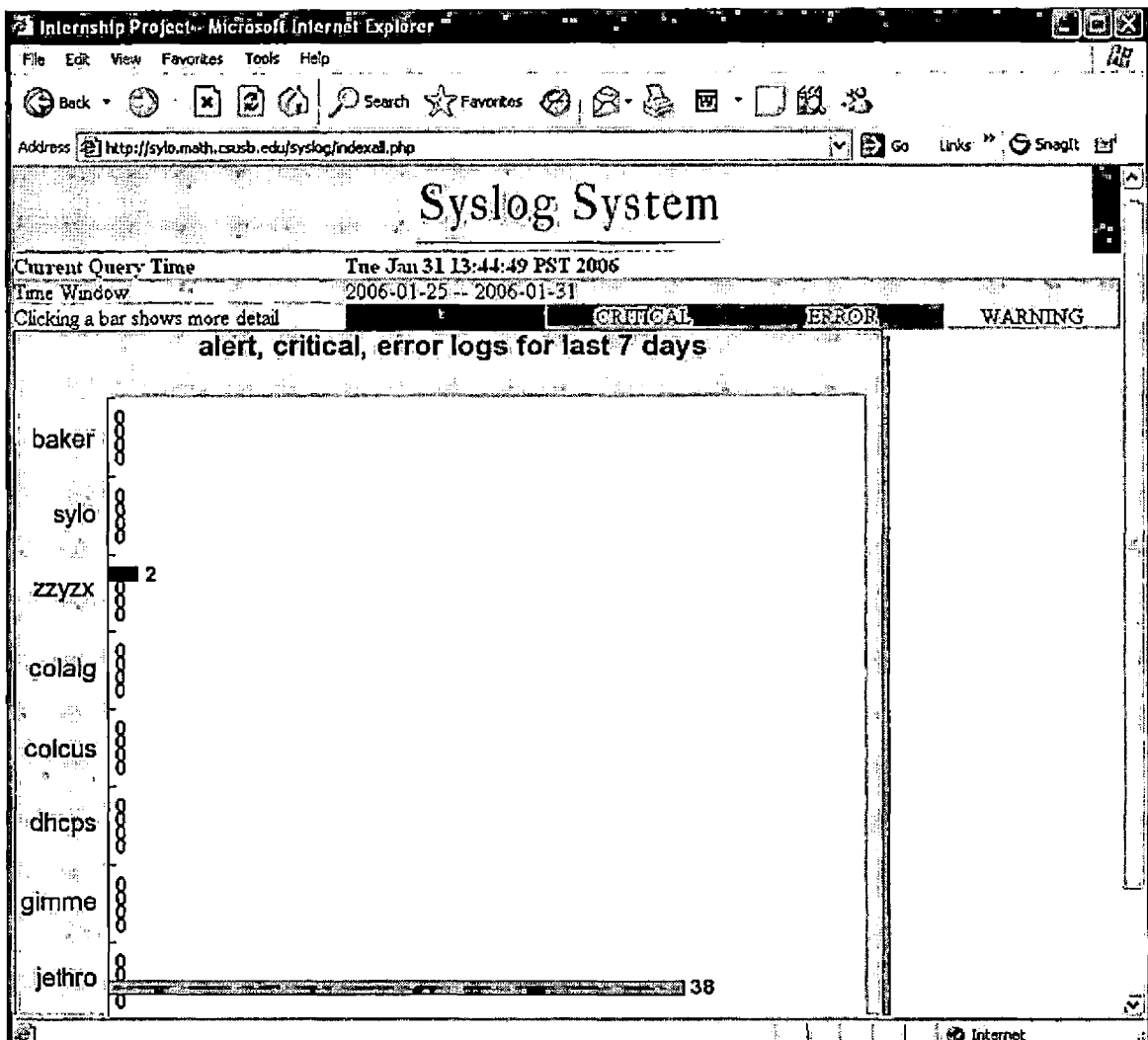


Figure 4.11. Last 7 Days Log

4.3.3 NESSUS Module

Figure 4.12 is main page of NESSUS module. It consists of Running NESSUS and Report parts.

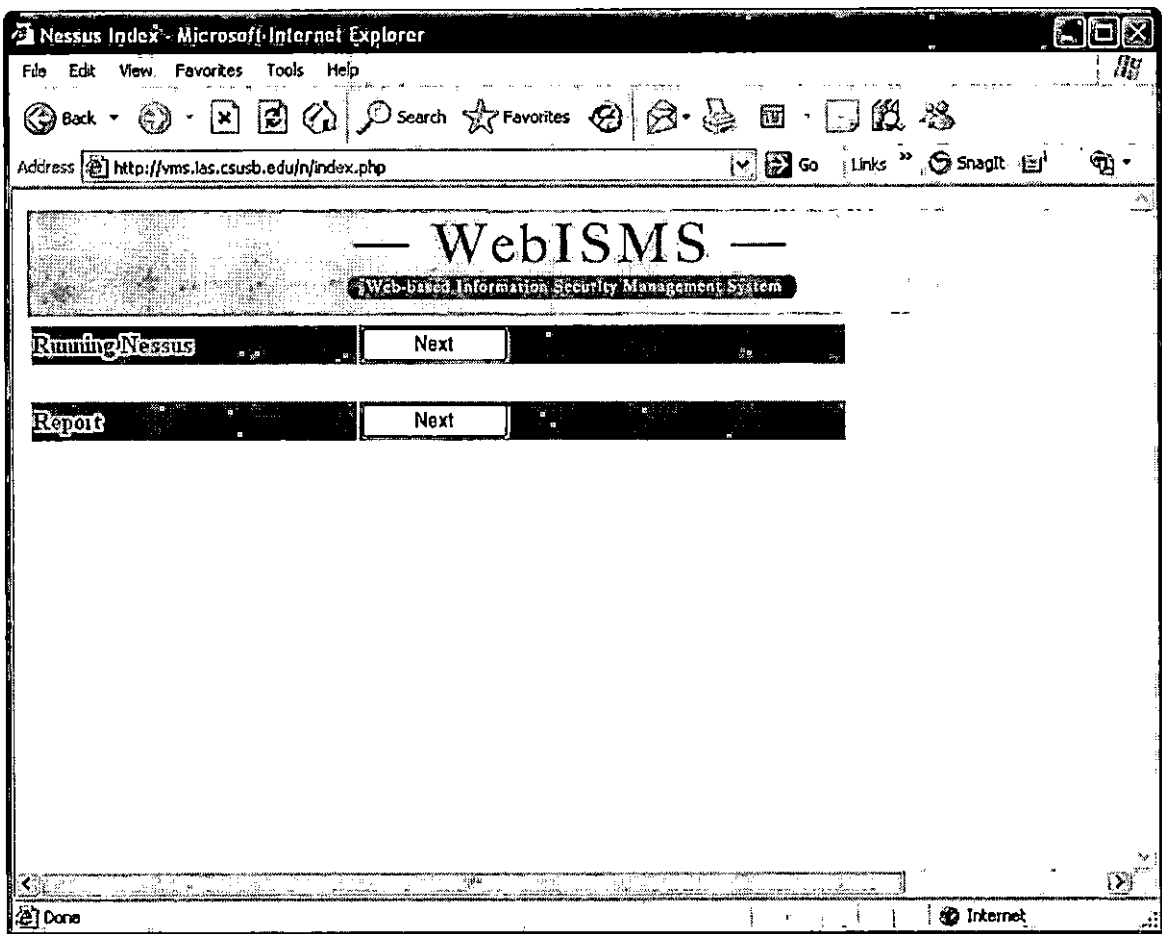
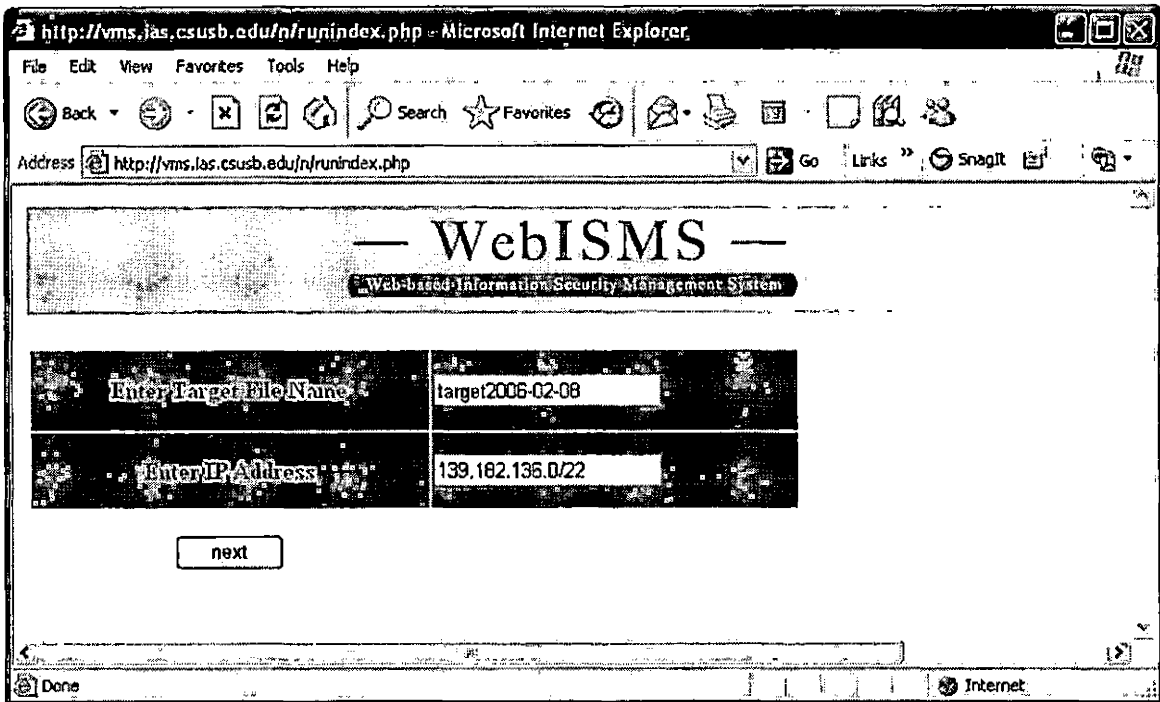


Figure 4.12. NESSUS Main

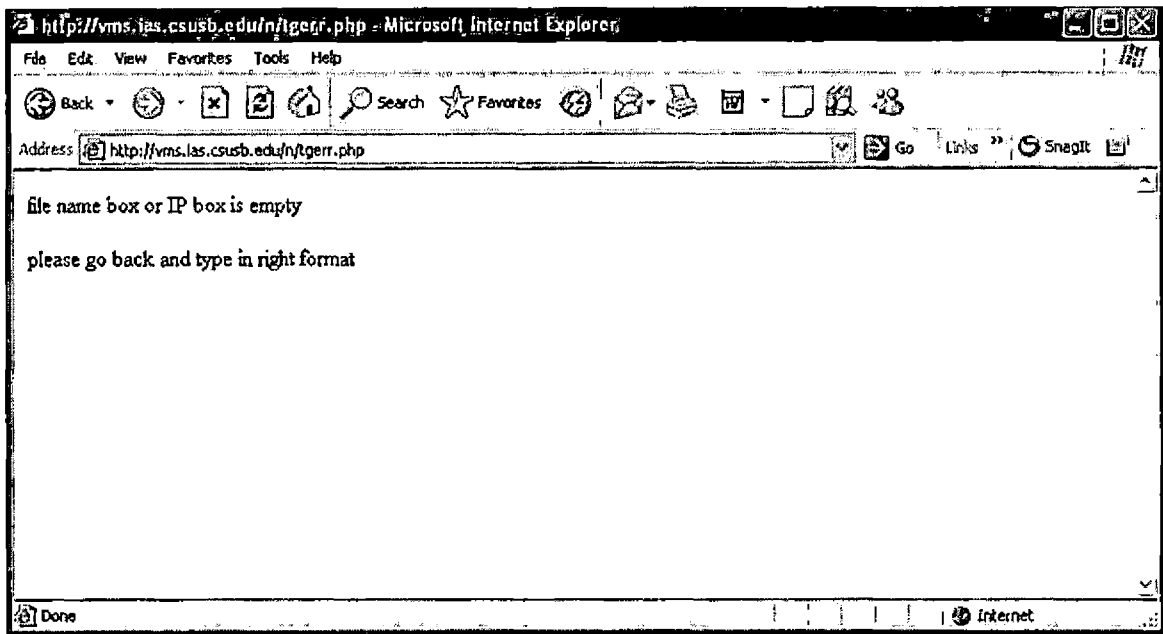
4.3.3.1 Running NESSUS. After clicking a next button of Running Nessus menu, NESSUS option page is appeared as in Figure 4.13. Recommended target file format is "target0000-00-00" as `_targetYear-Month-Day`. In this demonstration, "target2006-01-31" is typed. For IP address box, either single IP address or a network address with netmask is a legitimate format. In this module, three

exception handling cases are implemented. First, if either target file box or IP box is empty, it goes to the error page. Second, it checks whether a target file is created. Third, NESSUS process is independent to the Web page process. If the web page is closed or goes to the other site, NESSUS process is still alive and saves the result to the database. The main reason this exception handling is implemented is that NESSUS process usually takes 3 minutes to 7 days depending on the IP list. Lastly, When NESSUS process is over, the result page is created that create a link of instant report page.



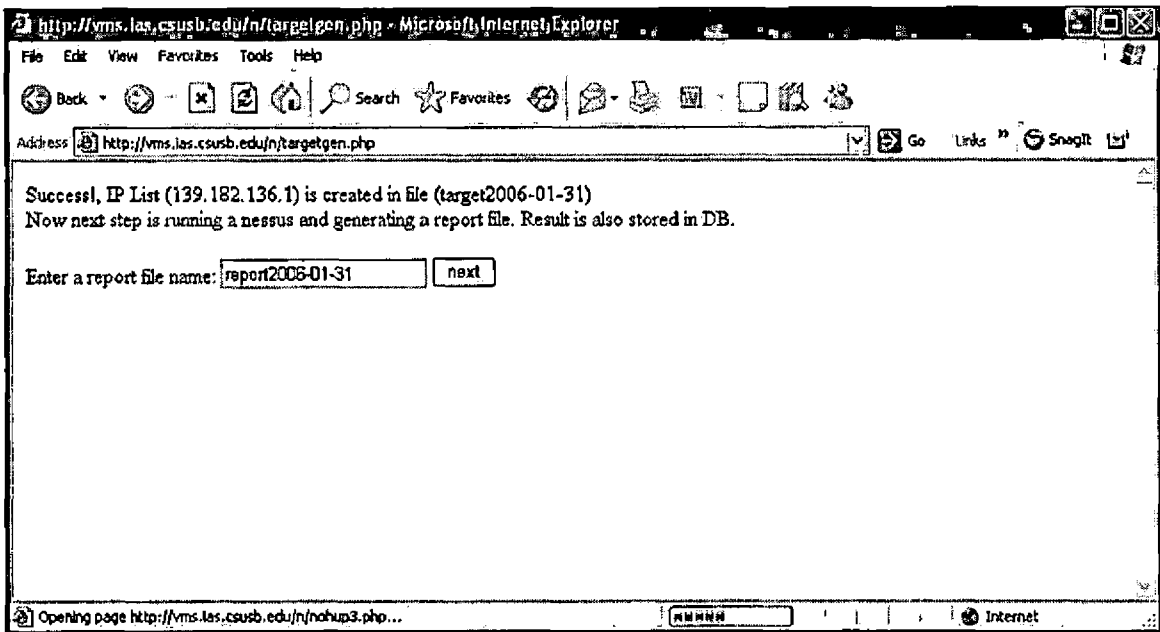
(1)

Figure 4.13. Demonstration of Running NESSUS



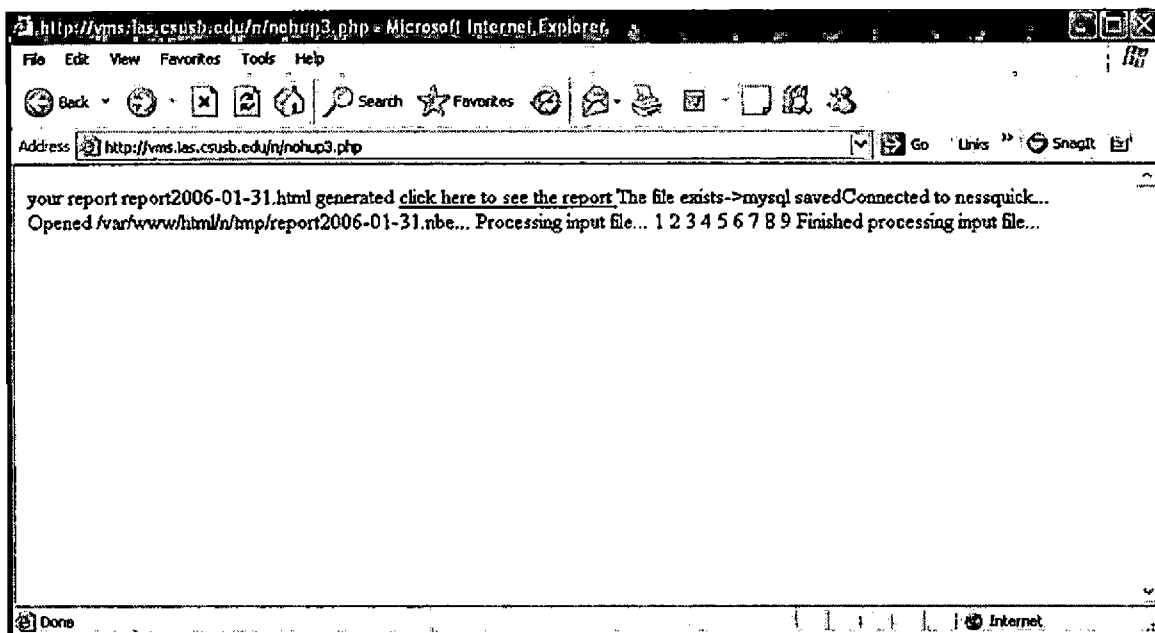
(2)

Figure 4.13. Demonstration of Running NNESSUS (continued)



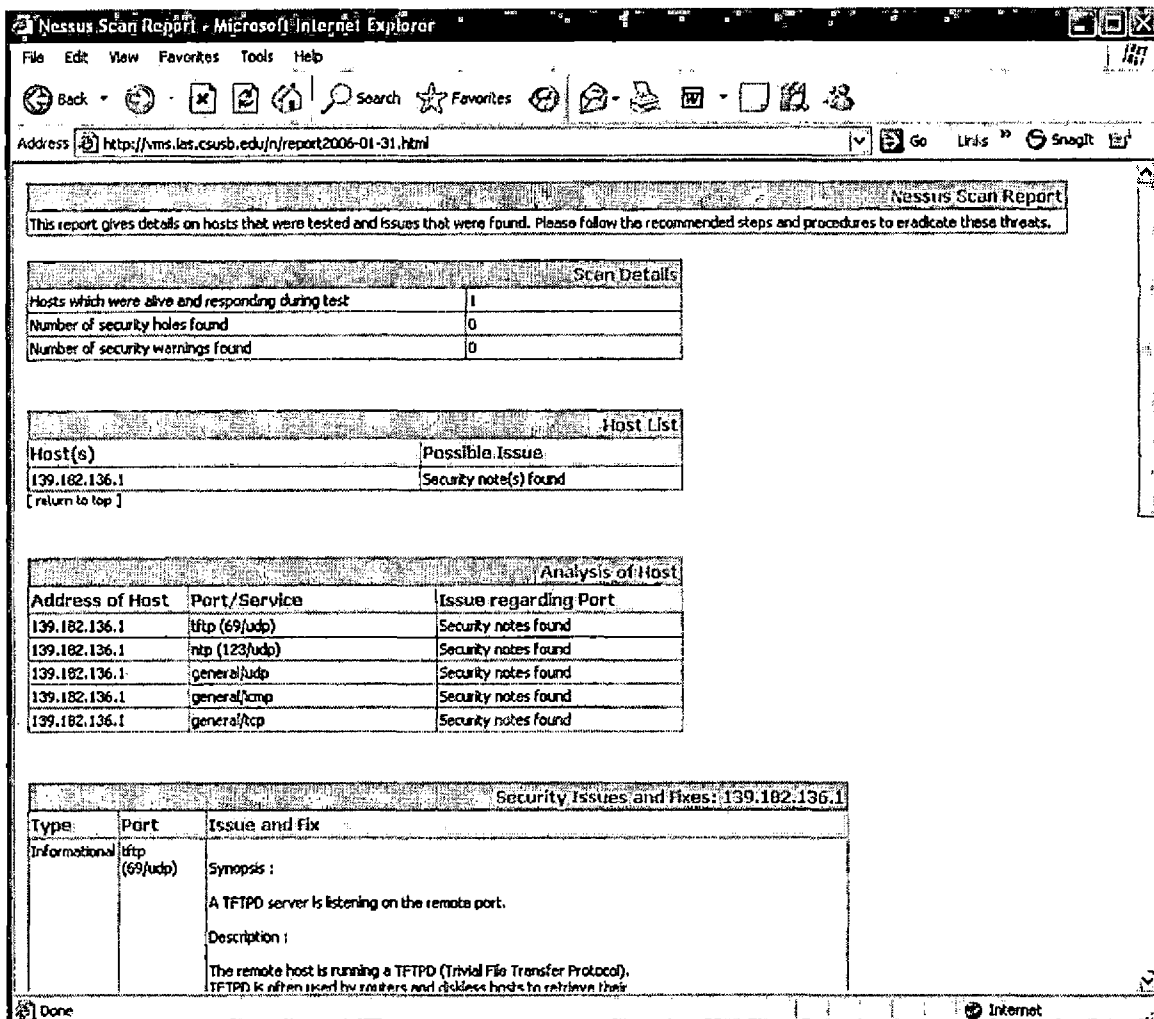
(3)

Figure 4.13. Demonstration of Running NESSUS (continued)



(4)

Figure 4.13. Demonstration of Running NNESSUS (continued)



(5)

Figure 4.13 Demonstration of Running NESSUS(continued)

4.3.3.2 NESSUS Report. Main menu of NESSUS Report

shows two options (Figure 4.14). First, "Show all of hosts having vulnerabilities" shows all the list of host with a clickable icon (Figure 4.15). If any icon is clicked, it

shows detailed information about vulnerabilities (Figure 4.16). Second, vulnerability report is generated based on the IP address (Figure 4.17). The report generated is shown in Figure 4.18.

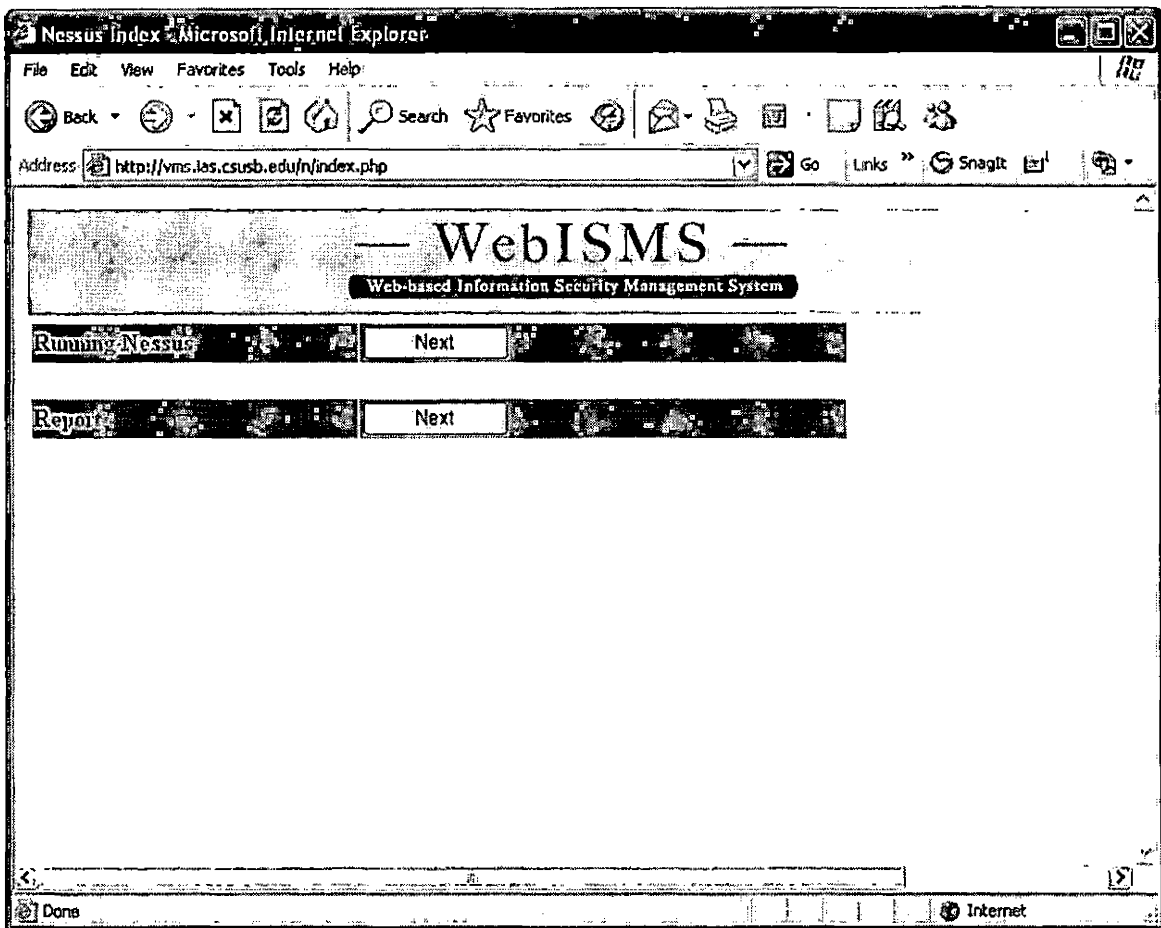


Figure 4.14. NESSUS Module Main Page

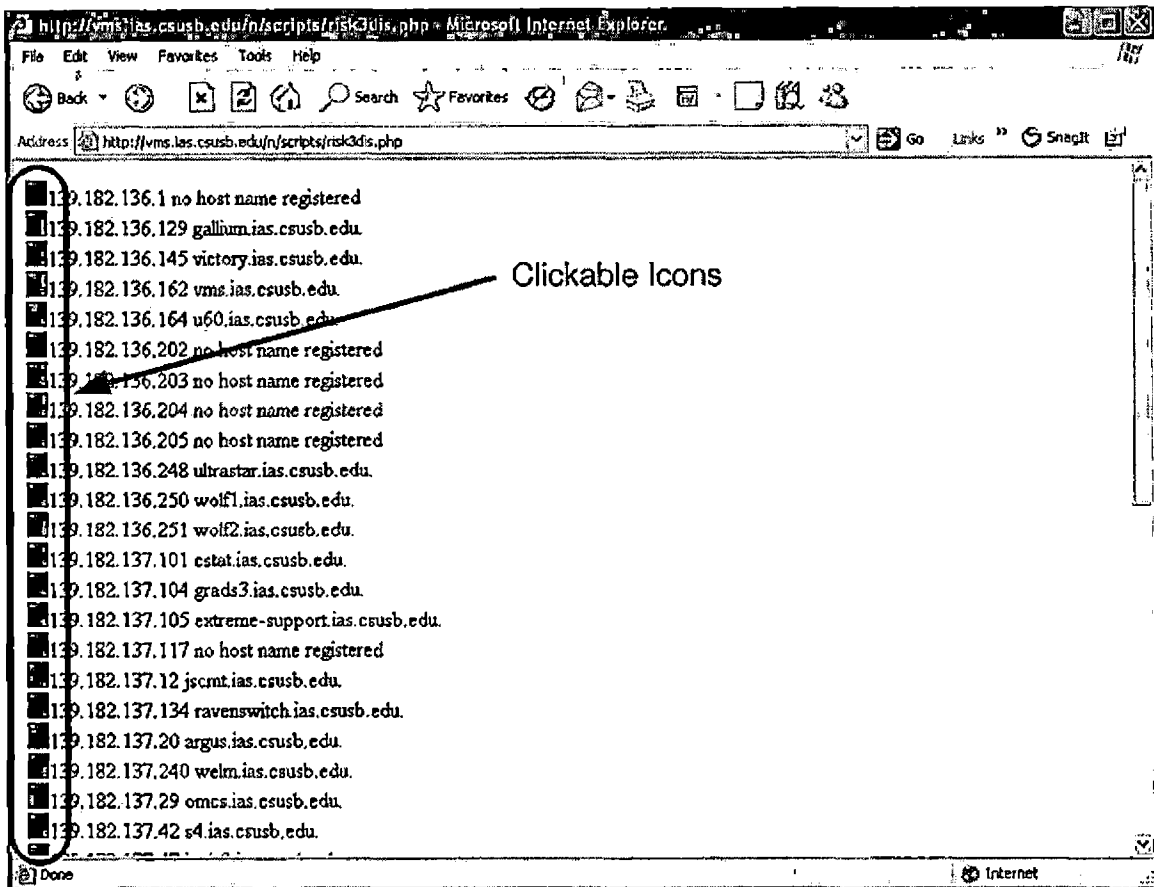


Figure 4.15. Demonstration of Show All of Hosts Having Vulnerabilities

http://vms.ias.csusb.edu/n/scripts/hout.php?ip=139.182.136.129 - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites Go Links SnagIt

Address http://vms.ias.csusb.edu/n/scripts/hout.php?ip=139.182.136.129

host	msg	timestamp
139.182.136.129	Synopsis : It is possible to determine the exact time set on the remote host. Description : The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : None / CVSS Base Score : 0 (AV:R/AC:L/Au:N/R:C/N/A:N/N:B/N) CVE : CVE-1999-0524	Mon Dec 5 08:57:13 2005
139.182.136.129	Synopsis : It is possible to determine the exact time set on the remote host. Description : The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : None / CVSS Base Score : 0 (AV:R/AC:L/Au:N/R:C/N/A:N/N:B/N) CVE : CVE-1999-0524	Mon Dec 5 08:57:13 2005
139.182.136.129	The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524	Mon Dec 5 08:57:13 2005
139.182.136.129	Information about this scan : Nessus version : 2.2.5 Plugin feed version : 200511171015 Type of plugin feed : Registered (7 days delay) Scanner IP : 139.182.136.162 Port scanner(s) : nessus_tcp_scanner Port range : default Thorough tests : no Experimental tests : no Paranoia level : 1 Report Verbosity : 1 Safe checks : yes Scan Start Date : 2005/12/5 8:56 Scan duration : 37 sec	Mon Dec 5 08:57:13 2005
139.182.136.129	139.182.136.129 resolves as gallium.ias.csusb.edu.	Mon Dec 5 08:57:13 2005
139.182.136.129	The remote host is running one of these operating systems : Linux Kernel 2.6 Linux Kernel 2.4	Mon Dec 5 08:57:13 2005

Done Internet

Figure 4.16. Demonstration of a Icon Clicked

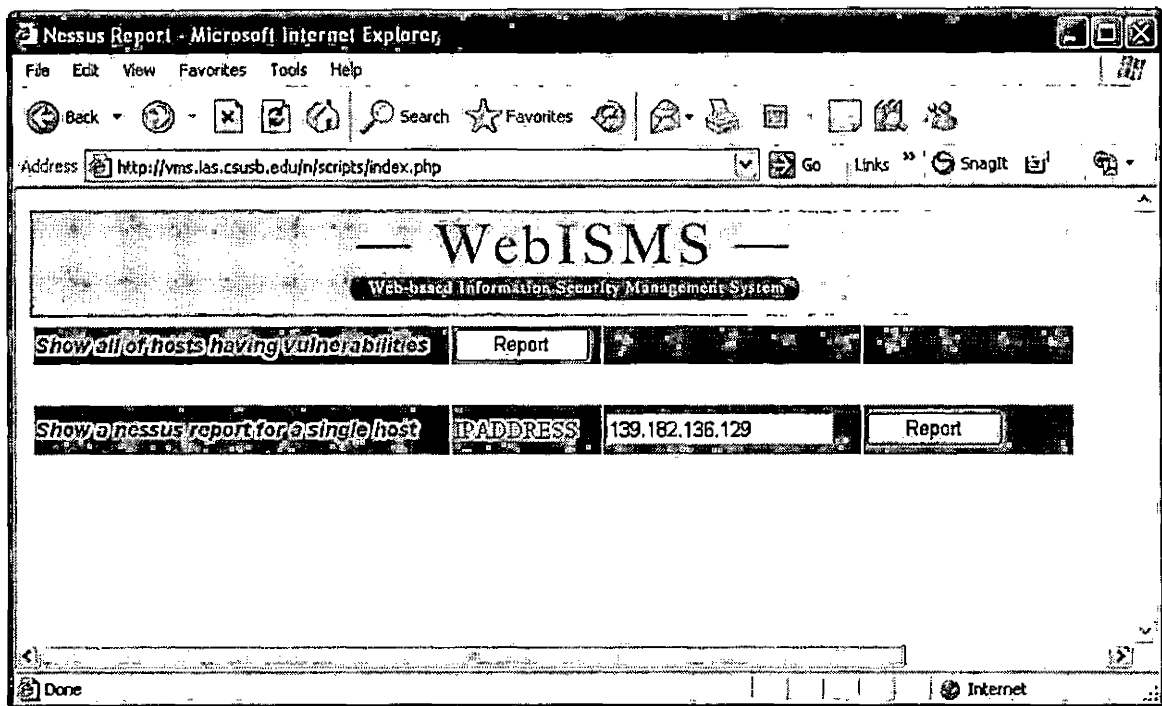


Figure 4.17. Demonstration of Generating a Report Based on Single Host

host	msg	timestamp
139.182.136.129	Synopsis : It is possible to determine the exact time set on the remote host. Description : The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : None / CVSS Base Score : 0 (AV/R/AC/L/Au/NR/C/N/A/N/LN/B/N) CVE : CVE-1999-0524	Mon Dec 5 08:57:13 2005
139.182.136.129	Synopsis : It is possible to determine the exact time set on the remote host. Description : The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : None / CVSS Base Score : 0 (AV/R/AC/L/Au/NR/C/N/A/N/LN/B/N) CVE : CVE-1999-0524	Mon Dec 5 08:57:13 2005
139.182.136.129	The remote host answers to an ICMP timestamp request. This allows an attacker to know the date which is set on your machine. This may help him to defeat all your time based authentication protocols. Solution : filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). Risk factor : Low CVE : CAN-1999-0524	Mon Dec 5 08:57:13 2005

Figure 4.18. Results from Show a NESSUS Report for Single Host

4.3.4 NMAP Module

Similar to NESSUS main page, NMAP main page has "running Nmap" and "Report" features (Figure 4.19). In "Running Nmap", there are two options to run NMAP. In Figure 4.20 shows single-host scan as typing in IP address of a single host and runid. Any random number is accepted, but date format is suggested. Figure 4.20 shows a example of runid as 20060201 (year month day). Figure 4.21 shows instant result as they are saved in the database at the same time. Figure 4.22 shows a demonstration of running

NMAP for a whole network scan. Any multiple IP addresses separated by a space or a network address with netmask are accepted. If any other characters than numbers are typed in runid and IP address boxes, error message is shown as Figure 4.23 and Figure 4.24.

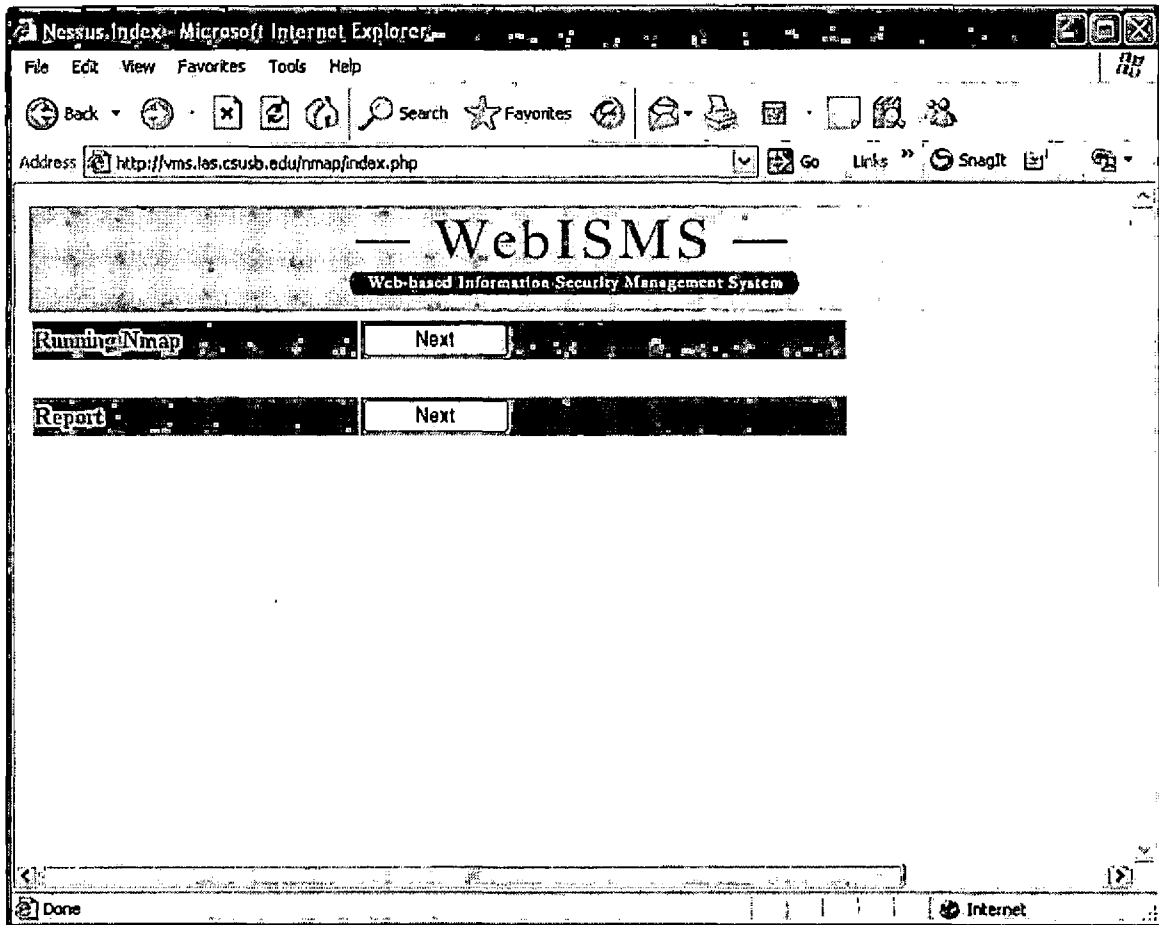


Figure 4.19. NMAP Main Page

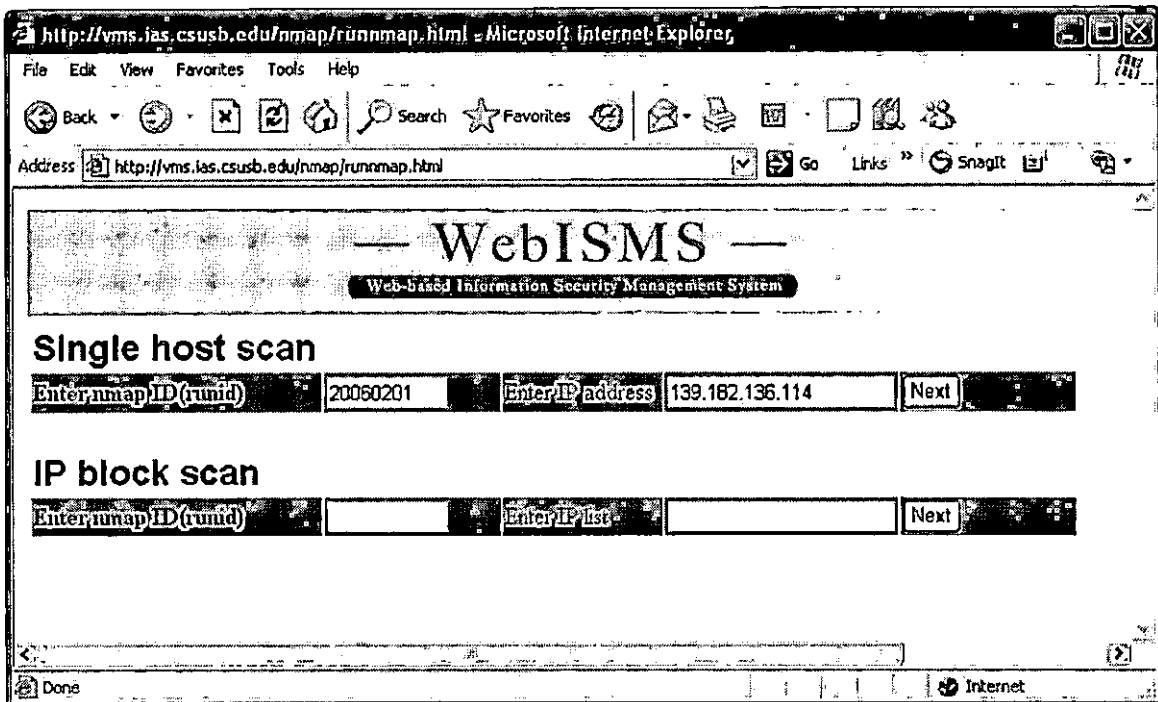


Figure 4.20. Demonstration of Single Host Scan

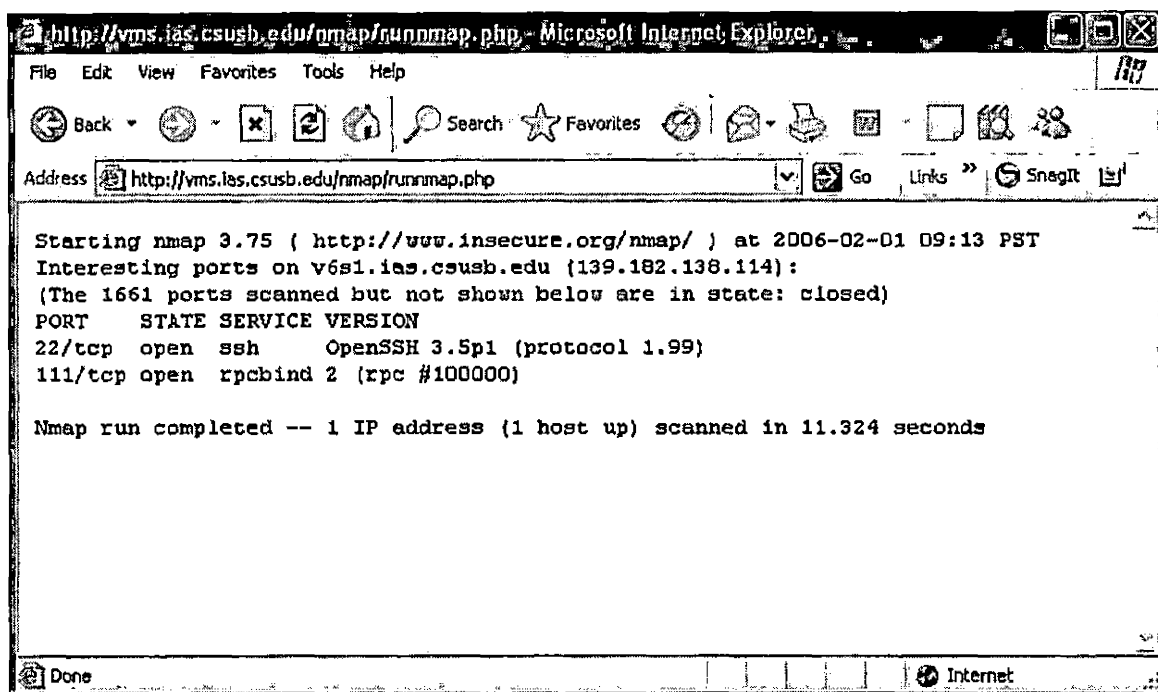


Figure 4.21. Demonstration of Instant Result

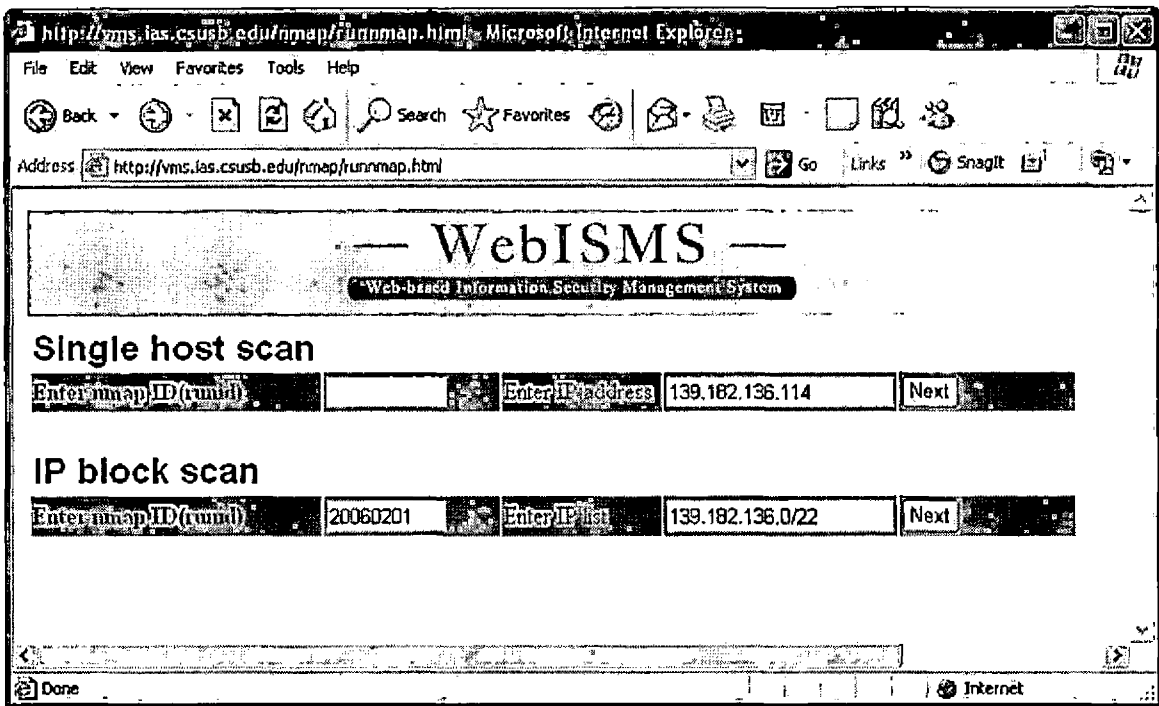


Figure 4.22. Demonstration of IP Block Scan

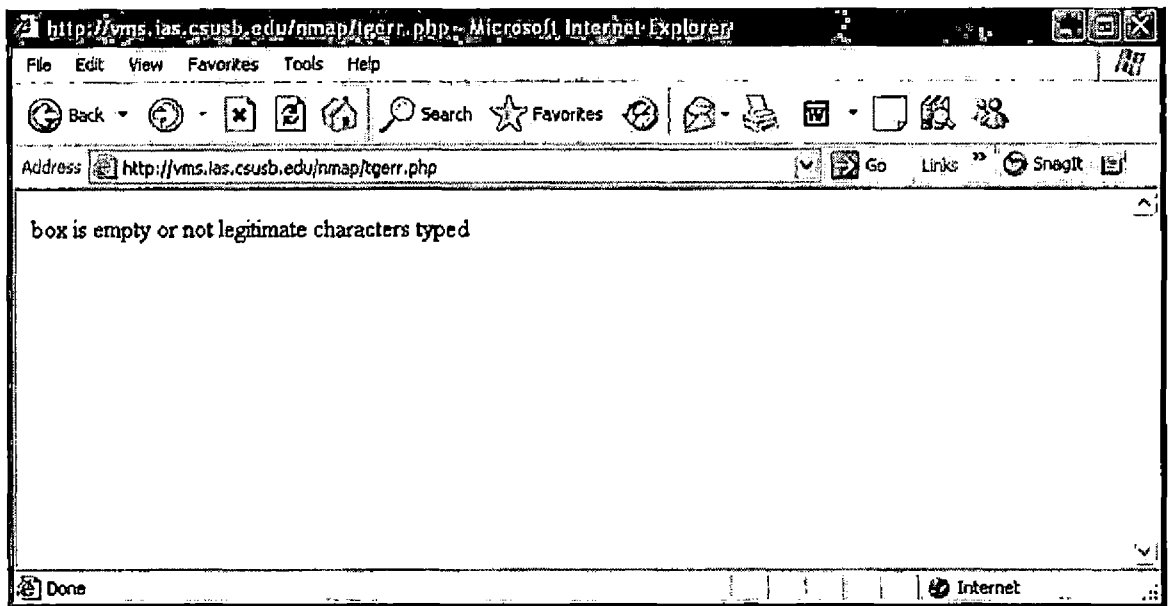


Figure 4.23. Error Message from Typed in Incorrect Runid

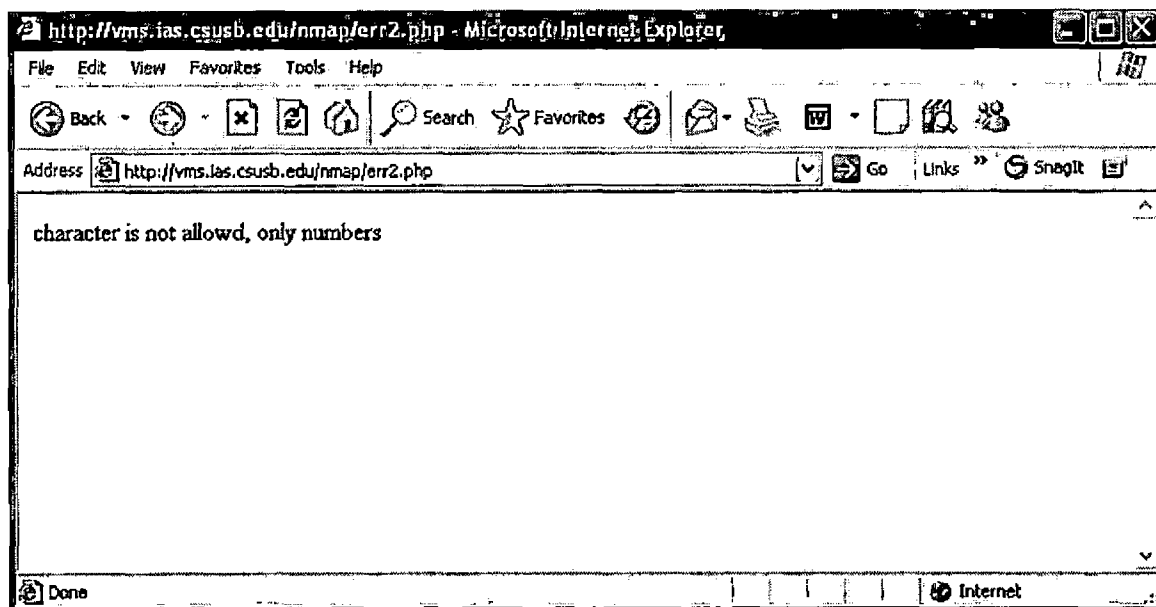


Figure 4.24. Error Message from Typed in Incorrect IP

4.4 System Acceptance Testing

System testing involves examination of the whole system of WebISMS. All the software and hardware components and any interfaces are tested in this process. Table 4.2 shows the result of system testing for WebISMS.

Table 4.2. System Testing Results

System Testing	Result
Install Operating System (Fedora core 4) in a X86 System box to make sure all of hardware components in functional with a Operating System	pass
Install and configure network components to make sure whether networking of the machine works correctly	pass
Install apache web server and start a daemon	pass
Install NMAP and test	Pass
Install NESSUS, start a daemon, and test	Pass
Install SYSLOG-NG and test with a client machine	Pass
Install MySQL database and start a daemon	Pass
Verify the pipe script program that parses logs and inserts them to the MySQL	pass
Install WebISMS program on Web server	pass

Specific test cases that when a wrong input is entered, or When Web browser is closed in the middle of running a NESSUS or a NMAP process, the system does not crash and handles the erroneous input will be described as following:

- This system is designed to run a module process and web page process separately using a UNIX nohup program and a background job controller so that in case of a web page process is killed, the module process is running until the outcome is generated. Figure 4.25 shows NESSUS process is just started from the web page, on the other hand, Figure 4.26 shows NESSUS process is still running as the driving Web page is closed.

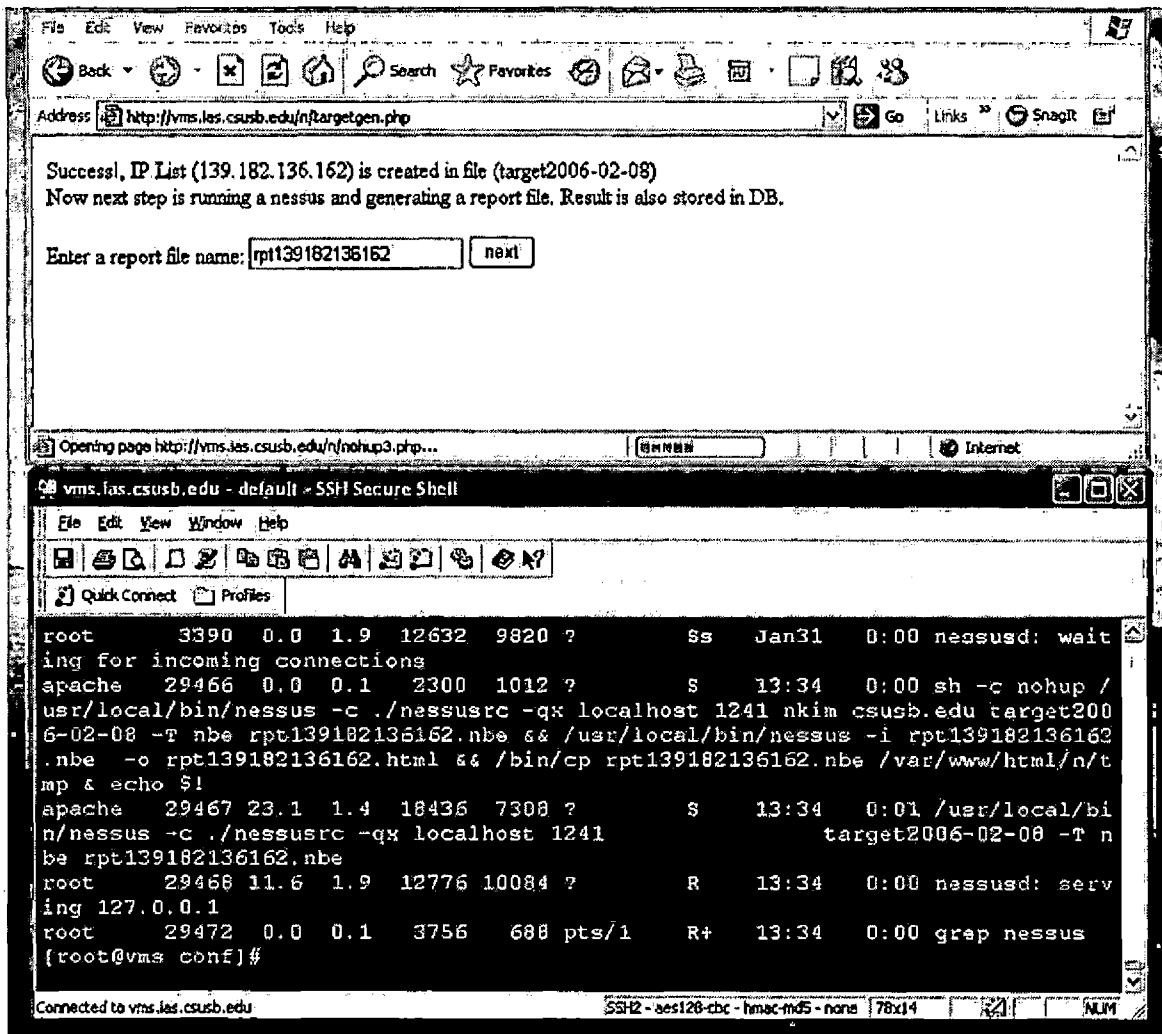


Figure 4.25. Screenshot of NESSUS Process Started

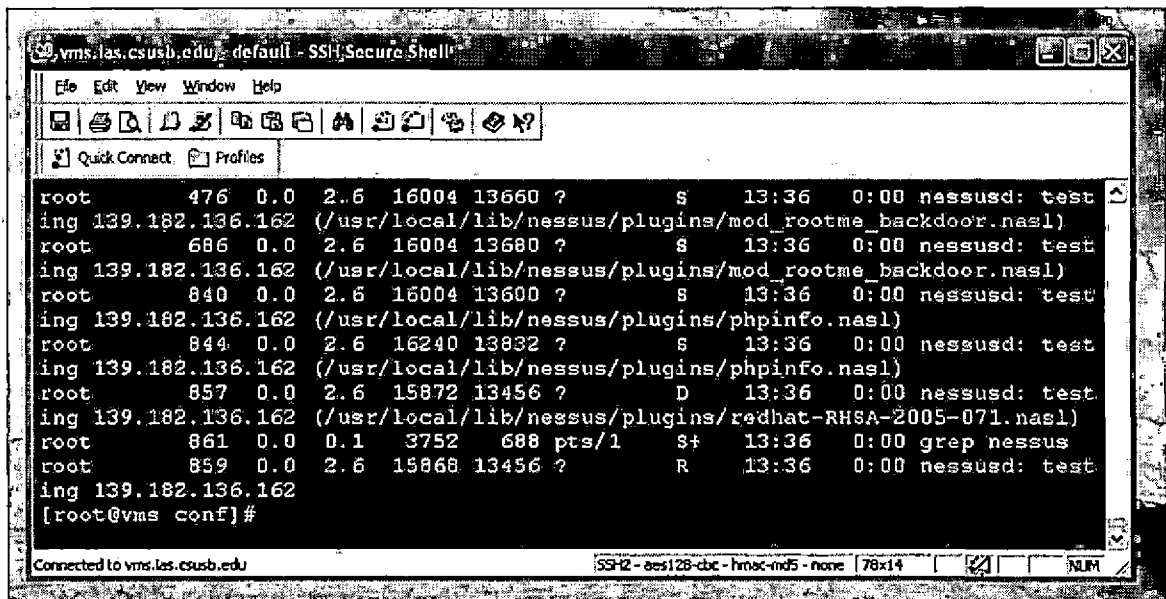


Figure 4.26. Screenshot of NESSUS process without Driving Web Page

- When incorrect login user name or password is entered, it goes to the page which has an error message.

Figure 4.27 and Figure 4.28 show the screenshot of the cases.

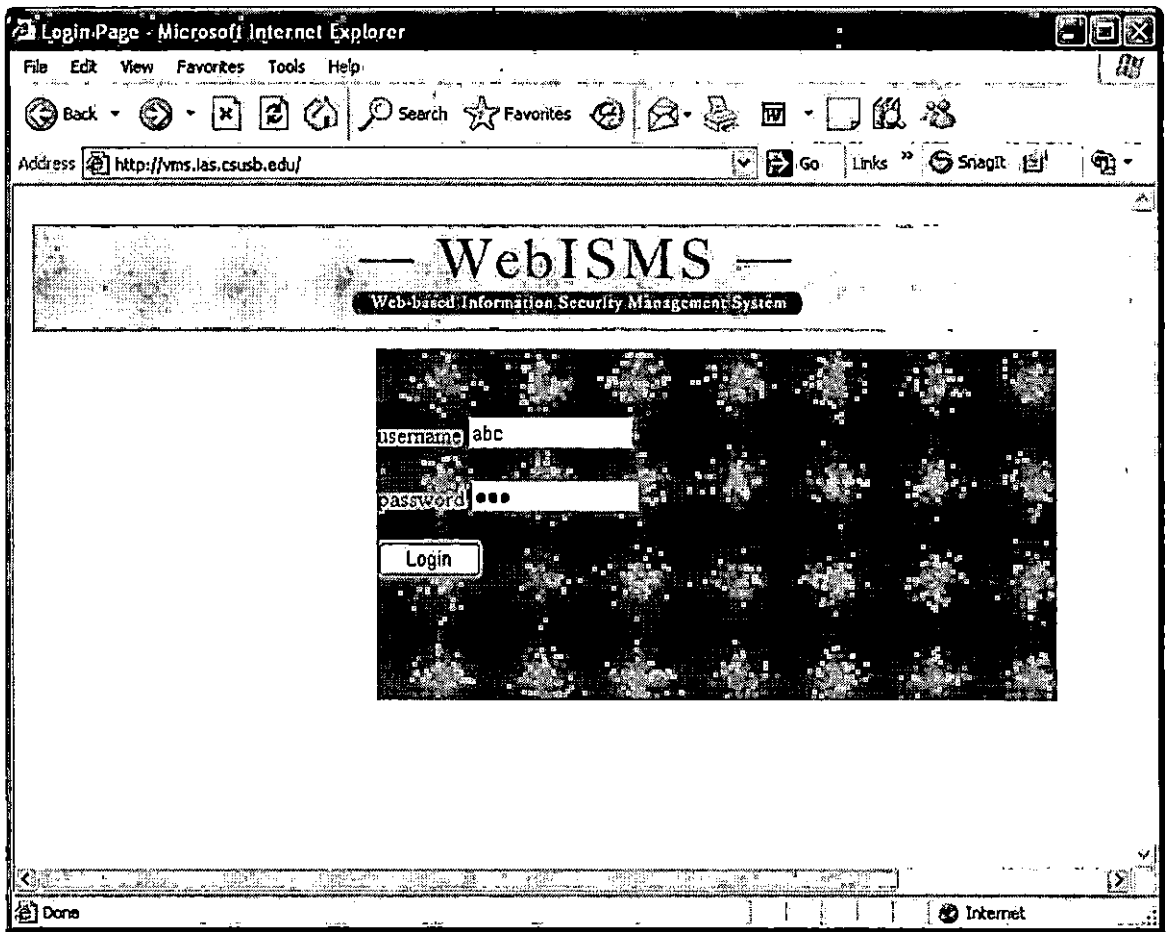


Figure 4.27. Screenshot of Wrong Username Entered

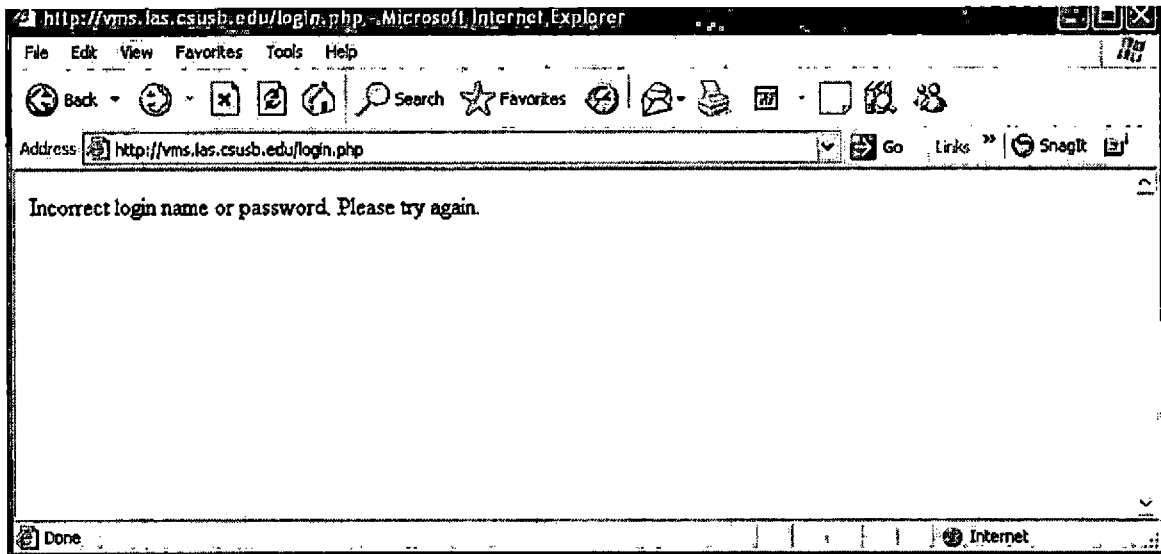


Figure 4.28. Screenshot of Wrong Username or Password Handled

- When a target file name is not entered in Running
NESSUS page, it goes to the error message page.
Figure 4.29 and Figure 4.30 shows screenshots.

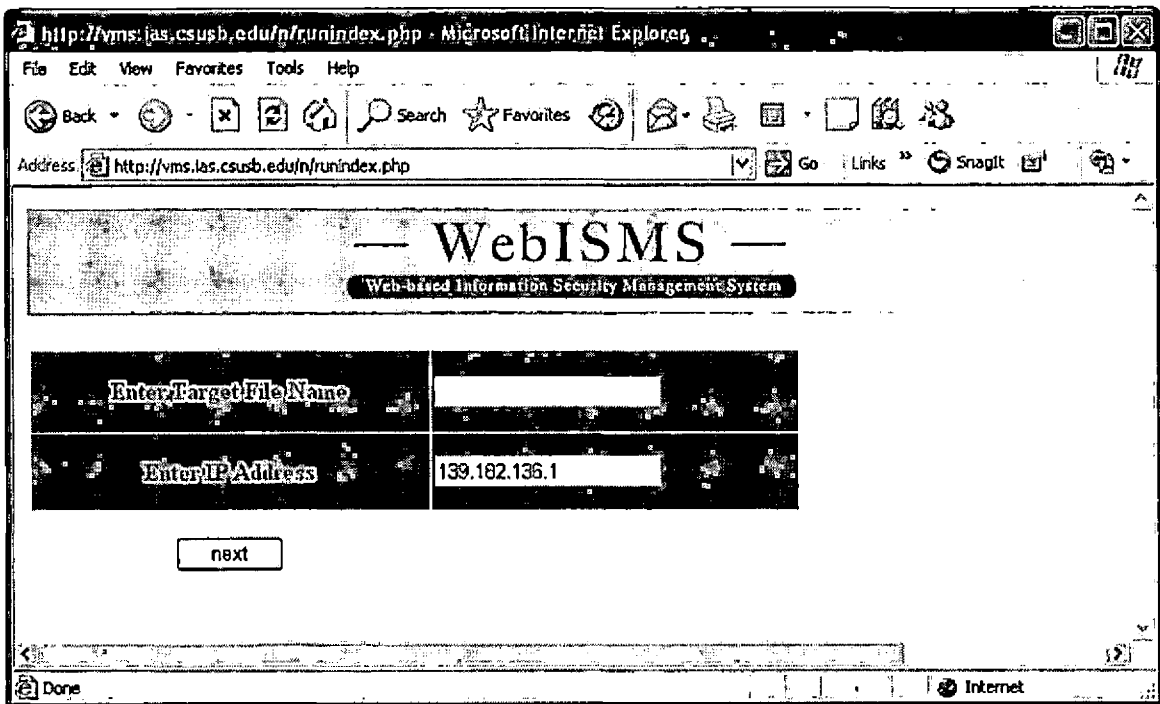


Figure 4.29. Screenshot of Empty Target File Box

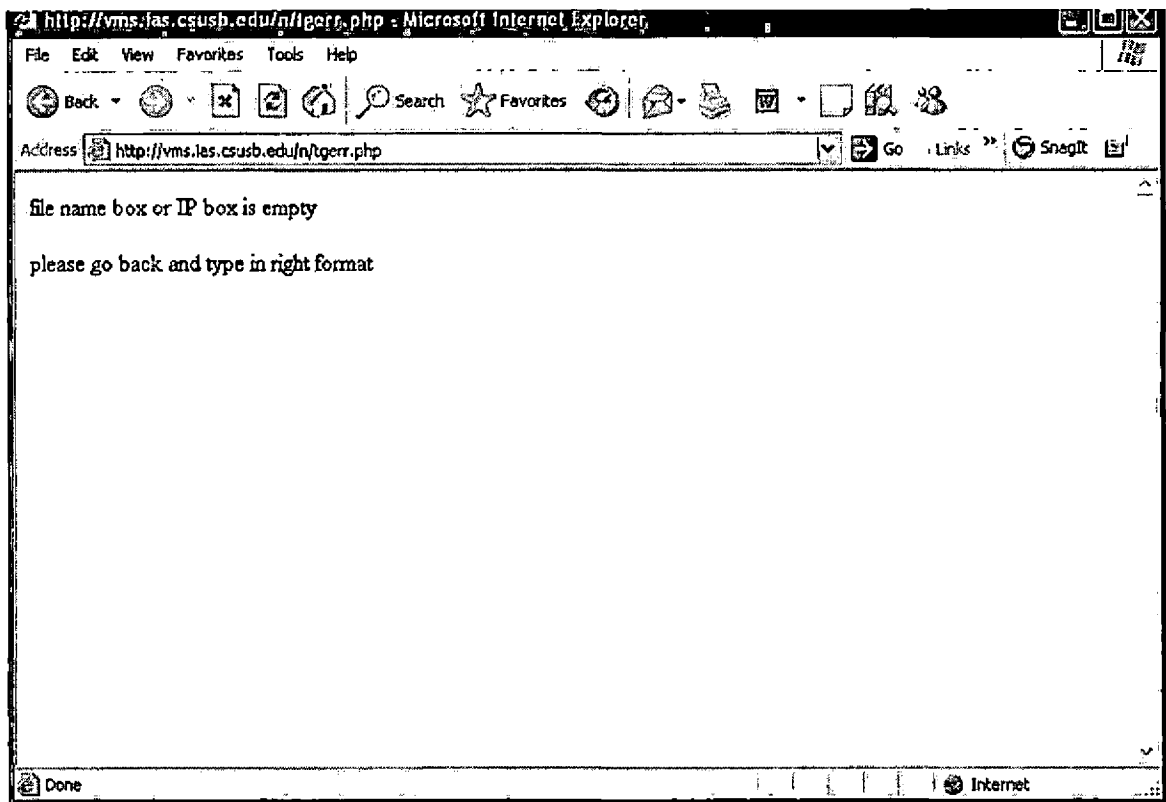


Figure 4.30. Screenshot of Empty Box Error Message

- When a user is trying to skip over the login page and directly access to any module page, Web server blocks and asks the user name and password. Figure 4.31 shows a screenshot that a user is attempting to access the syslog module, but a login window is popped up.

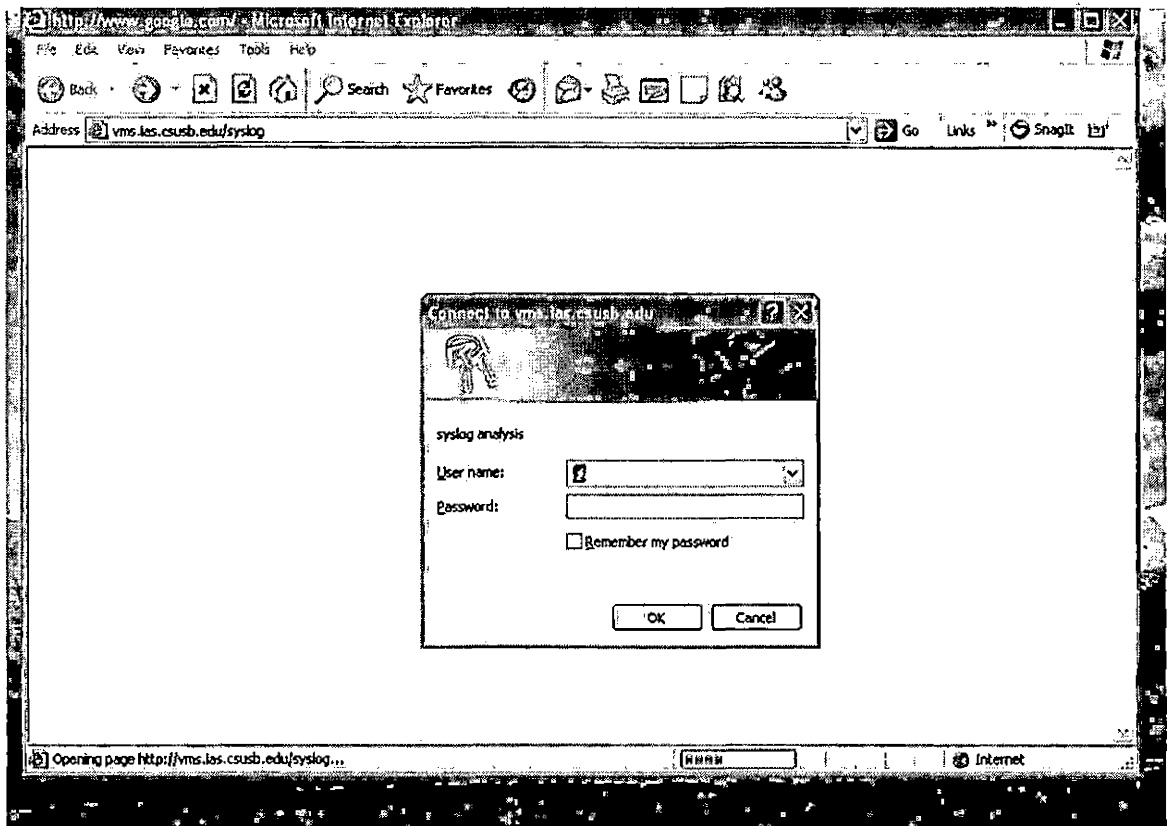


Figure 4.31. Screenshot of Block a Direct Access

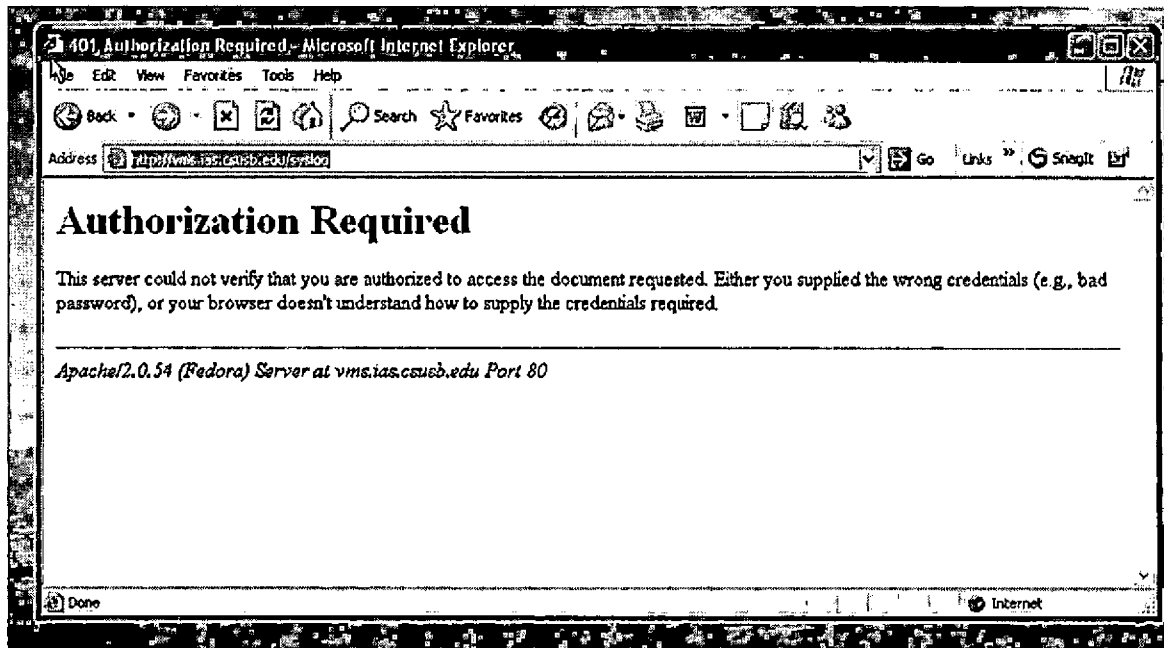


Figure 4.32. Screenshot of Authorization Required Message

- When a illegitimate format of runid or IP address is entered in a Running NMAP page, it goes to the error page. Figure 4.12-1 and Figure 4.33 shows the screenshots.

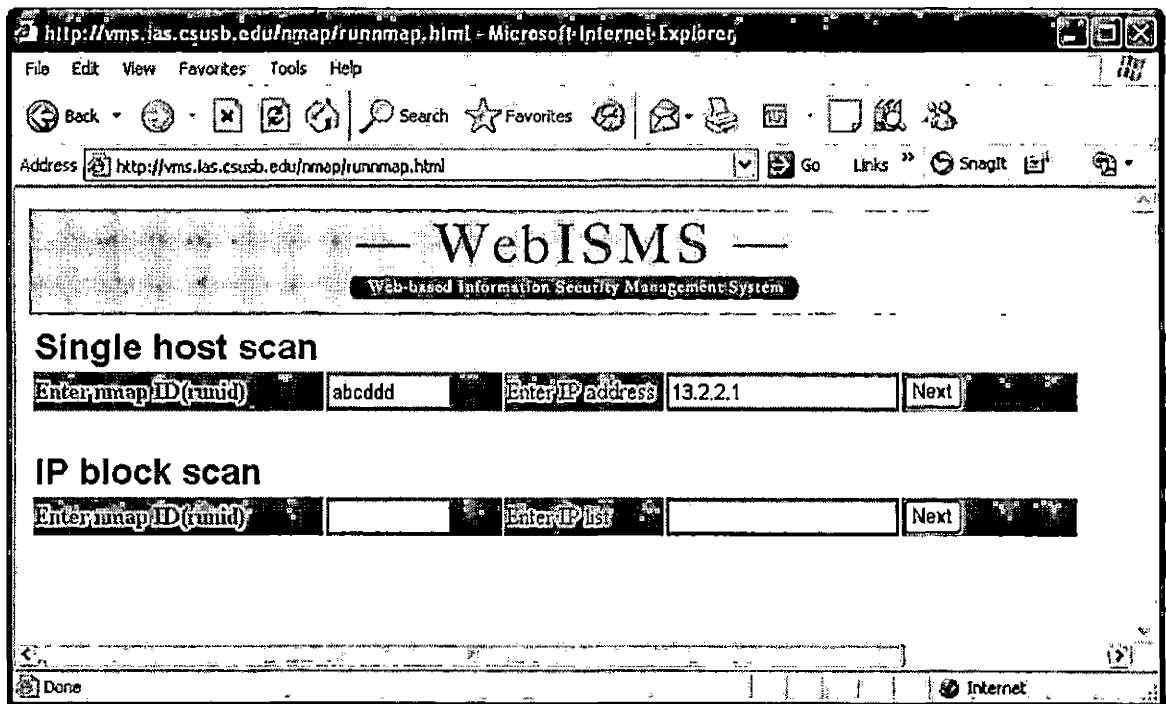


Figure 4.33. Screenshot of Wrong Runid Entered

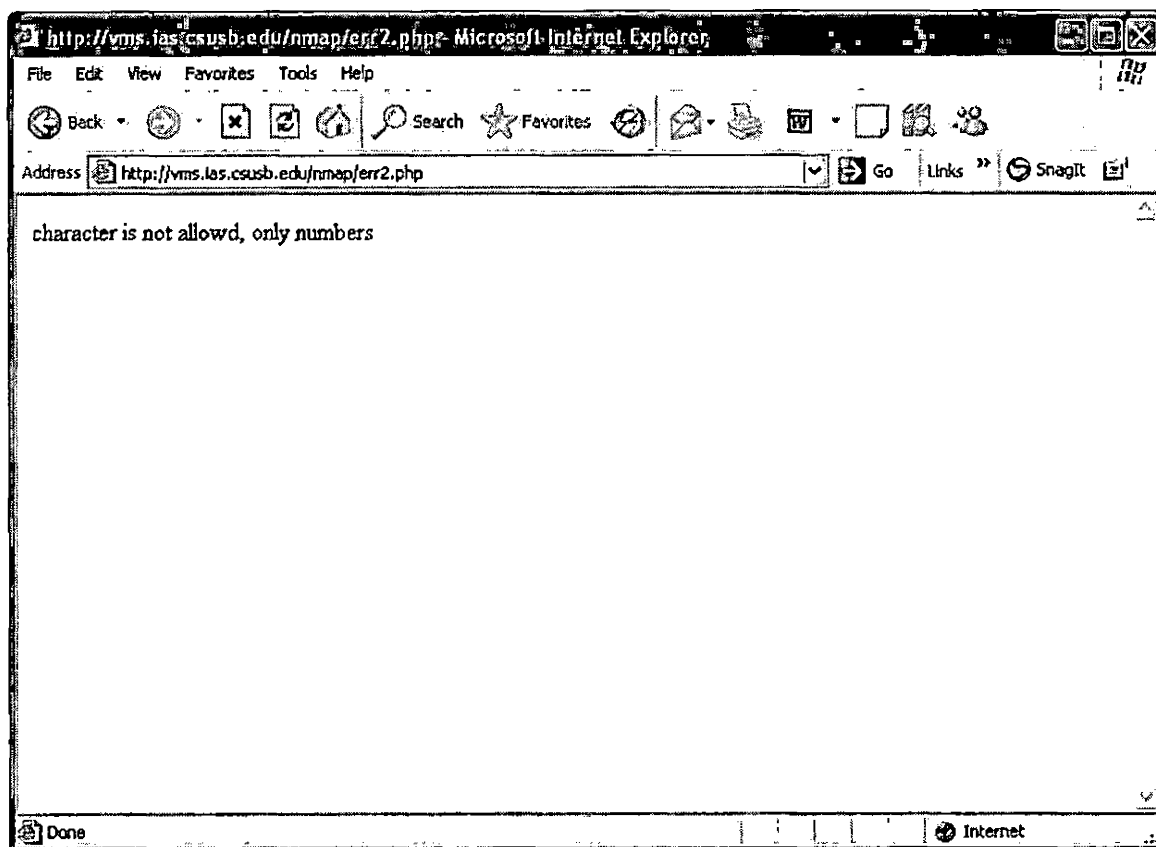


Figure 4.34. Screenshot of Character Error Message

- When a wrong format of IP address (including a blank box) is typed, it goes to the error message (Figure 4.34 and Figure 4.36).

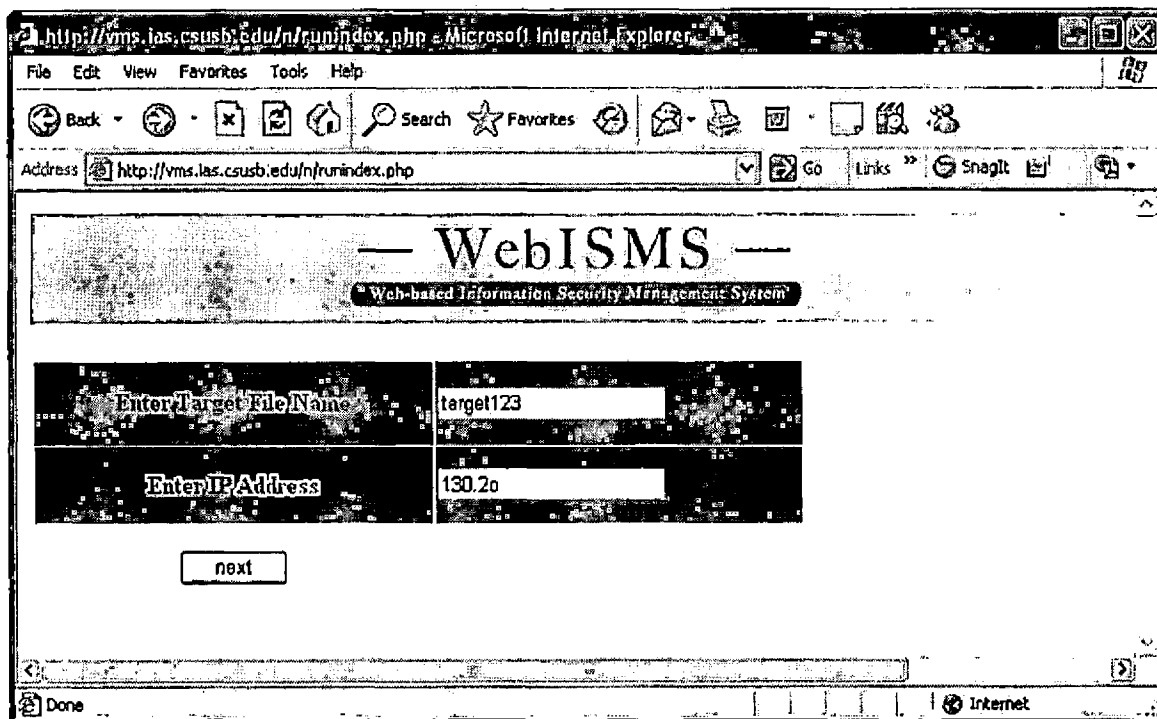


Figure 4.35. Screenshot of Wrong IP Entered

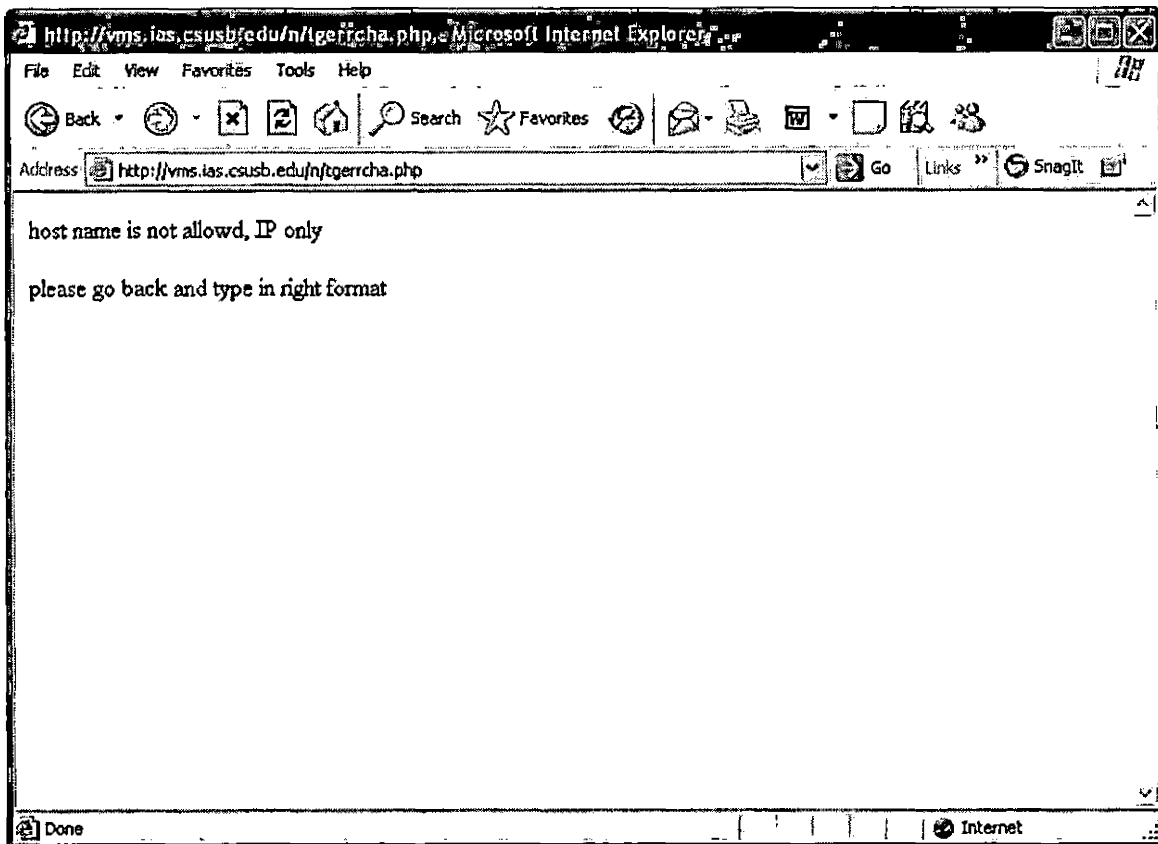


Figure 4.36. Screenshot of IP Error Message

CHAPTER FIVE
INSTALLATION AND MAINTENANCE

5.1 Directory Structure

WebISMS is based on Web technology that root directory of WebISMS is the Apache Web server documentation directory. The root directory and its sub directories are showed in Figure 5.1: Directory Structure Diagram. It is also suggested to install NMAP and NESSUS under Web root directory because the Web process should be owner of those directories and files.

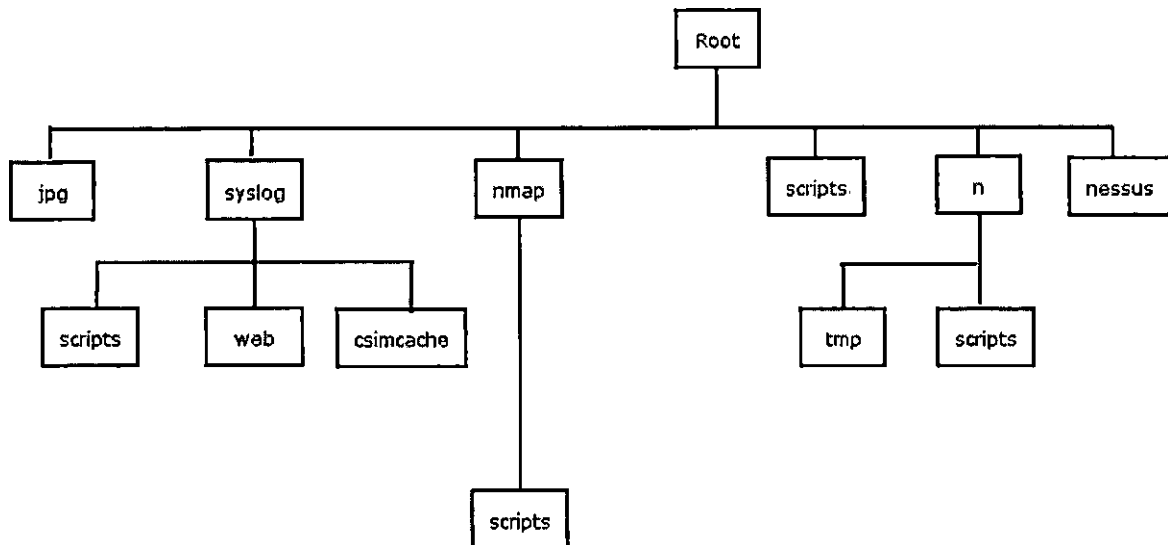


Figure 5.1. Directory Structure Diagram

- Root - root directory of WebISMS and its absolute directory is /var/www/html
- Jpg - graphic library directory for bar graphs
- Syslog - root directory of syslog module that has three sub directories, scripts, web and csimcache
 - scripts - there are PHP scripts for sub functions
 - web - syslog search scripts direcotory
 - csimcache - this is cache directory for JpGraph image files
- nmap - directory for nmap binary files and PHP scripts
 - scripts - nmap sub function scripts
- scripts - it keeps all PHP scripts for root sub functions
- n - NESSUS scripts root directory
 - tmp - temporary directory for generating temporary files
 - scripts - directory for sub functions of NESSUS
- nessus - NESSUS binary directory

5.2 Installation

This software is for the system administrator or information security administrator who is an expert in Linux, MySQL, PHP, and network assessment/audit tools. Therefore, this documentation skips the basic installation procedure for Linux, MySQL, Apache, and PHP.

5.2.1 NMAP and NESSUS Installation

The installation of NMAP and NESSUS is straight forward as following:

1. Get source code and unzip and untar them under root directory.
2. Compile them and install binaries under root directory.
3. Test NMAP and NESSUS in the command line to make sure they are functional.
4. Change ownership of NMAP and NESSUS to Web server account.

5.2.2 SYSLOG-NG Installation

Installation and configuration of SYSLOG-NG is difficult because it should be configured as a centralized syslog-ng and direct piping to MySQL database. Here are steps for server side:

1. Install syslog-ng from source code or rpm.
2. Disable the default syslog.

3. Edit syslog-ng.conf file. Here is syslog-ng.conf for WebISMS.

```
options {
    sync (0);
    time_reopen (10);
    log_fifo_size (1000);
    long_hostnames (off);
    use_dns (yes);
    use_fqdn (no);
    create_dirs (no);
    keep_hostname (yes);
};

source s_sys {
    file ("/proc/kmsg" log_prefix("kernel: "));
    unix-stream ("/dev/log");
    internal();
    udp(ip(0.0.0.0) port(514));
};

destination d_cons { file("/dev/console"); };
destination d_mesg { file("/var/log/messages"); };
destination d_auth { file("/var/log/secure"); };
destination d_mail { file("/var/log/maillog" sync(10)); };
destination d_spol { file("/var/log/spooler"); };
destination d_boot { file("/var/log/boot.log"); };
destination d_cron { file("/var/log/cron"); };
destination d_mlal { usertty("*"); };

#filter f_filter1 { facility(kern); };
filter f_filter2 { level(info..emerg) and
    not facility(mail,authpriv,cron); };
filter f_filter3 { facility(authpriv); };
filter f_filter4 { facility(mail); };
filter f_filter5 { level(emerg); };
filter f_filter6 { facility(uucp) or
    (facility(news) and level(crit..emerg)); };
filter f_filter7 { facility(local7); };
filter f_filter8 { facility(cron); };
```

(1)

Figure 5.2 Configuration File of syslog-ng.conf

```

destination d_mysql {

pipe("/tmp/mysql.pipe"

template("INSERT INTO logs (host, facility, priority, level, tag, date,
time, program, msg) VALUES ( '$HOST', '$FACILITY', '$PRIORITY',
'$LEVEL',
'$TAG',

'$YEAR-$MONTH-$DAY', '$HOUR:$MIN:$SEC', '$PROGRAM', '$MSG' );\n")
template-escape(yes));

};

log { source(s_sys); destination(d_mysql);

};

#log { source(s_sys); filter(f_filter1); destination(d_cons); };
log { source(s_sys); filter(f_filter2); destination(d_mesg); };
log { source(s_sys); filter(f_filter3); destination(d_auth); };
log { source(s_sys); filter(f_filter4); destination(d_mail); };
log { source(s_sys); filter(f_filter5); destination(d_mlal); };
log { source(s_sys); filter(f_filter6); destination(d_spol); };
log { source(s_sys); filter(f_filter7); destination(d_boot); };
log { source(s_sys); filter(f_filter8); destination(d_cron); };

```

(2)

Figure 5.2 Configuration File of syslog-ng.conf(continued)

4. Configuring MySQL template is an important part to be done by the system administrator. The recommended configuration is to use the above syslog-ng.conf following the database design.
5. Next step is to create a fifo pipe file which will insert logs to MySQL in real-time. Here are steps.

- 1) `Mkfifo /tmp/mysql.pipe.`
- 2) Restart `syslog-ng` daemon.
- 3) `Mysql -u root -pxxxx syslog < /tmp/mysql.pipe.`
- 4) Check the database whether logs are inserted or not.
If not, check the error logs in `/var/log` directory to trouble-shoot.

5.2.3 Database Installation

Once NESSUS, NMAP, and SYSLOG-NG are functional, next step is the installation of the database for the modules. The databases are already backed up and saved in CD, so just restore them in the MySQL. All databases are backed up in 'WebISMS_DB.sql'. The command line, "`mysql -u root -p < WebISMS_DB.sql`" will install the databases.

5.2.4 WebISMS Installation

`WebISMS.tar.gz` is the file holding all script files and directory structures. The steps to install are:

1. Go to the Web Server documentation root directory.
2. Copy `WebISMS.tar.gz` from CD to root directory.
3. Extract `WebISMS.tar.gz` by "`tar -zxvf WebISMS.tar.gz`".
4. Delete the tarball.

5.3 Maintenance

5.3.1 Syslog Module

syslog system is a very noisy system that logs grow so fast that the database size could be over 1 gigabyte within one week depending on the number of syslog clients.

Therefore, maintaining syslog database is important. It is suggested that when the database space is designed, the system administrator should plan how much database space should be reserved, how often the database should be backup and deleted to manage the database space. If the system administrator maintains more than 10 client servers as a syslog client, the database space of WebISMS should be at least 10 gigabyte with a separate partition, and using cron jobs to backup and delete useless data periodically on a weekly or monthly basis.

5.3.2 NESSUS PLUG-INS Update

NESSUS plug-ins should be updated daily to get an updated vulnerability information. NESSUS has a tool, 'nessus-update-plugins' to get an update, so it should also be in cron jobs to run daily.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

As the number of computers deployed in the College of Natural Science at CSUSB increases, there will be a greater chance that they will be compromised or infected. Then, it becomes difficult for the system administrator to repair each individual system one by one. This situation requires the need for an information security management system to prevent any kinds of security breaches. To meet this demand, WebISMS is developed with the following features:

- NESSUS (Network Assessment) - scans a target machine to find vulnerability in the application level and stores the event in a database to generate a report.
- SYSLOG (monitoring log) - receives the client machine's log in real-time and stores the logs for monitoring system's network activities and application software's status.
- NMAP (Network Audit) - scans a target machine's ports to check any vulnerable service running from virus/worm/rootkits.

- Web-based GUI - WebISMS is accessible from any Web browser as long as the system is up and running
- Backend Database - results from NESSUS, NMAP, and SYSLOG are stored in a database and generate a report accessible on the Web.

WebISMS is deployed in the Institute of Applied Supercomputing Research lab and Math VLANs. There are several benefits to running WebISMS in these two networks:

- System monitoring time reduced - SYSLOG module is able to monitor multiple machines simultaneously and quickly search for high priority logs. Chart below shows the number time spent for system log monitoring. Blue columns is the time spent for each individual system log as logging in the machine and viewing the system log file. Time spent for each machine is about 5 to 6 minutes so that monitoring 6 machines takes about 30 minutes. On the other hand, Red bars show the time spent for monitoring with SYSLOG-NG. With centralized SYSLOG-NG, it takes 5 to 10 machine. As a result, Monitoring time will be greatly reduced as the number of syslog client machines increases.

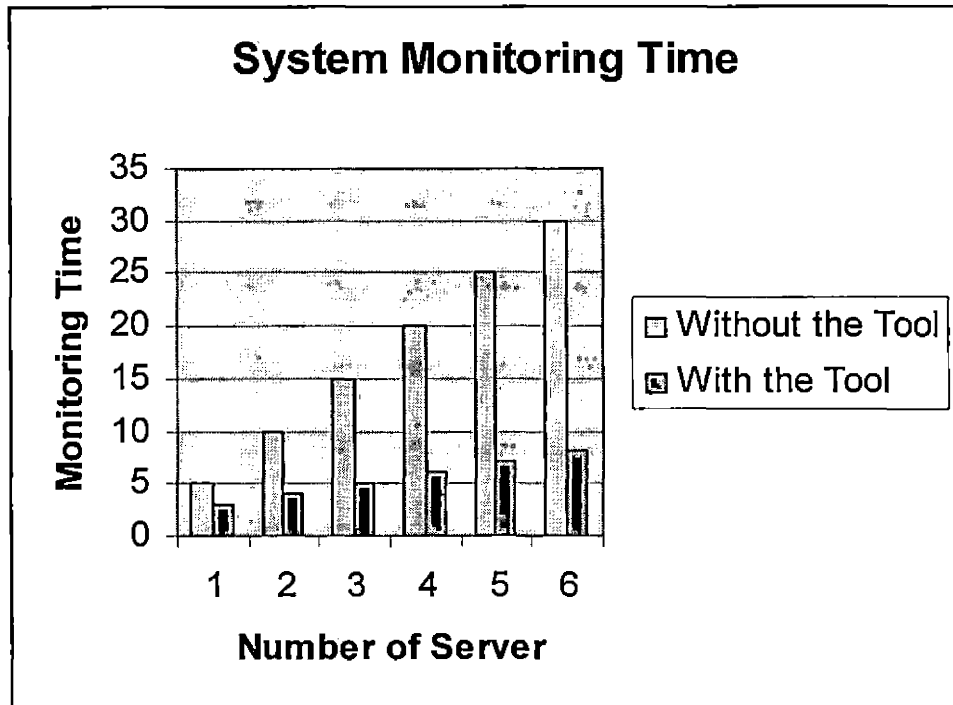


Figure 6.1. Time Spent for Syslog Monitoring

- Preventing vulnerabilities - NNESSUS and NMAP help to find most vulnerability of a freshly installed machine.
- Assisting network security plan - Results from WebISMS help the system administrator or information security office for planning or

decision making in regards to information security

The primary purpose of WebISMS is for information security management as an assessment, audit, and monitoring tool. Results that are stored at the database of WebISMS will be valuable data not only for system administrator or information security officer but also for any researcher/student studying information security.

6.2 Future Work

WebISMS is module based that supports any system runs on the Web could also be a part of WebISMS. Therefore, future work on following features of WebISMS will enhance information security system:

- Adding a network traffic monitoring system like MRTG to be able to monitor network bandwidth.
- Adding a Intrusion Detection System like SNORT to watch who is trying to break in the system.
- Adding a network transaction auditing tool like ARGUS to track and report on the status and performance of all network transactions seen in a data network traffic stream.

- Adding a honeypot system that watches hacker's behaviors and what tools are being used.
- If it is possible, adding firewall's or router's log to SYSLOG-NG to monitor Wide Area Network traffic.
- Deploying multiple NESSUS, NMAP, and SYSLOG servers to gain a better performance and stability.
- The utilization of data mining to determine trends and common patterns that can be used to improve security.

REFERENCES

- [1] Elizabeth Castro, "HTML for the World Wide Web," Peachpit press, 2000.
- [2] Rob Flickenger, "Linux Server Hacks," O'Reilly, 2003.
- [3] Jason Gilmore, "PHP and MySQL," Apress, 2004.
- [4] The Institute of Electrical and Electronics Engineers, "IEEE Recommended Practice for Software Requirements Specifications Std.," Std 830-1998.
- [5] Andrew Lockhart, "Network Security Hacks," O'Reilly, 2004.
- [6] Chris MacNab, "Network Security Assessment," O'Reilly, 2004.
- [7] Cyrus Peikari, Anton Chuvakin, "Security Warrior," O'Reilly, 2004.
- [8] David Sklar, Adam Trachtenberg, "PHP Cookbook," O'Reilly, 2003.