AN EXTENDED ACCESS CONTROL LIST FOR LOCAL NETWORK PROTECTION FROM INSIDER ATTACKS

MUAMER N. MOHAMMED

MASTER OF COMPUTER SCIENCE
UNIVERSITY MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG
CENTER FOR GRADUATE STUDIES

We certify that the thesis entitled "An Extended Access Control List for Local Network Protection from Insider Attacks" is written by MUAMER N. MOHAMMED. We have examined the final copy of this thesis and that in our opinion; it is fully adequate in terms of scope and quality for the awarding the degree of Masters of Computer Science. We herewith recommend that it be accepted in fulfillment of the requirements for the degree of Masters of Computer Science.

Signature

Name of External Examiner
Prof. Madya Dr. Asri bin Ngadi
Institution: Faculty of Computer Systems
& Information Systems, University Technology Malaysia

UMP

Rohani
Signature

Name of Internal Examiner
Dr. Rohani binti Abo Bakar
Institution: Faculty of Computer Systems
& Software Engineering, University Malaysia Pahang

AN EXTENDED ACCESS CONTROL LIST FOR LOCAL NETWORK PROTECTION FROM INSIDER ATTACKS

MUAMER N. MOHAMMED

Thesis submitted in fulfillment of the requirements
for the award of the degree of Master of Computer Science

Faculty of Computer Systems and Software Engineering
UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2011

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion this thesis is satisfactory in terms of scope and quality for the award of the degree of Master of Computer Science.

Signature

Name of Supervisor: DR. NORROZILA BINT SULAIMAN

Position: SENIOR LECTURE

Date: 28 FEBRUARY 2011

Signature:

Name of Co-supervisor: MR. ABDULAH MAT SAFRI

Position: LECTURE

Date: 28 FEBRUARY 2011

UMP

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature

Name: MUAMER N. MOHAMMED

ID Number: MCC09002

Date: 28 FEBRUARY 2011

UMP

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratitude to my supervisor Dr. Norrozila Sulaiman for her ideas, invaluable guidance, continuous encouragement and constant support to making this research possible. I appreciate her consistent support from the first day I applied to graduate program to these concluding moments. I am truly grateful for her progressive vision about my training in science, her tolerance of my mistakes, and her commitment to my future career. I also would like to express thanks to my co-supervisor Mr. Abdulah Mat Safri.

A special thank to University Malaysia Pahang (UMP) for the real support and to Faculty of Computer Systems & Software Engineering for the Laboratory facility.

Special thank to Prof. Dr. Adel H. Salih for his encouragement, support, advice and care about my work. He is consistently encouraged me to further studies in Malaysia.

To all my friends in UMP, I would also like to thank them for their support and provide me a comfortable environment to study, and for the nice and useful moments I spent with them. To all my Malaysian friends, thank you for enlightening my life while I am far from my home, and create a new environment for me.

I would like to express greatest appreciation to my mother, father, brothers and sisters for their enormous love. They always respect my decision and give me their full support and encouragement over the years.

Finally, I am grateful to my wife, daughters and son for their sacrifice, patience, and understanding that were inevitable to make this work possible. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals.

ABSTRACT

The security of Local Area Network (LAN) has become one of the most important interesting areas for researches and this connection is prone to vulnerability caused by the attackers in steeling information from the network and possibly makes damages. Protecting the network can be done through many mechanisms among the most effective one is the network firewall. While the firewall focusing on protecting the network from the external attacks, it only limits the internal users accessing the network. Insider attacks can be unauthorized host, application, and/or user backdoor connected to the LAN and reveal information to the outside. These types of attacks can be very dangerous. This thesis proposes solutions for these problems by creating two programs one at each client and the other at the server. At client, the program will provide each outgoing packet destined outside the network with Host Identifier, Application Identifier and User Identifier responsible for sending the current outgoing packet. It also authenticates these Identifiers in order to ensure that it is trustworthy and valid for the second program. The server will receive the authenticated packets and verifies them before passing them to the external network, while dropping and track the unauthorized one. This work based on TCP/IP protocol suite because it is the leading and important current communication protocols. Both programs operate under Microsoft Windows operating system environment. The performance of the new system is computed and the results show that the security aspects have been enhanced with respect to a slight impact in speed (decreased by 1.96 % in download, 2.35% in uploading). Finally, the proposed system implementation was developed using Visual Basic.NET language.

UMP

ABSTRAK

Keselamatan rangkaian setempat (LAN) adalah salah satu bidang yang menarik dan penting untuk kajian. Sambungan komputer kepada rangkaian akan terdedah kepada keselamatan data yang mungkin disebabkan oleh pengodam maklumat dari rangkaian yang mungkin merosakkan data. Keselamatan rangkaian setempat boleh dilindungi melalui pelbagai mekanisma, antara yang paling berkesan adalah dinding api. Walau bagaimanapun, dinding api menumpukan kepada melindungi rangkaian dari serangan luaran. Ia hanya menyekat pengguna dalaman mengakses rangkaian. Serangan dalaman boleh terjadi melalui penyambungan rangkaian, aplikasi atau pengguna yang terhubung ke LAN secara tidak sah dan mendedahkan maklumat ke luar. Jenis-jenis serangan ini sangat berbahaya. Tesis ini mencadangkan penyelesaian untuk masalah ini dengan membangunkan dua jenis program, satu pada komputer pelanggan dan yang lagi satu pada komputer pelayan. Pada komputer pelanggan, program berfungsi untuk memastikan setiap paket yang keluar dari rangkaian direkodkan dengan identiti komputer, aplikasi dan pengguna. Ia juga digunakan untuk memastikan bahawa maklumat itu adalah boleh dipercayai sebelum dihantar ke program di komputer pelayan. Komputer pelayan akan menerima paket dan mengesahkannya sebelum dibenarkan ke rangkaian luaran, sedangkan paket yang tidak sah akan disekat untuk ke rangkaian luar dan maklumatnya akan direkodkan. Pengkajian ini berdasarkan TCP / IP protokol kerana ia adalah protokol komunikasi yang banyak digunakan. Kedua-dua program beroperasi dalam persekitaran sistem pengendalian Microsoft Windows. Prestasi dari sistem baru dinilai dan hasilnya menunjukkan bahawa aspek keselamatan telah ditingkatkan. Namun demikian, terdapat sedikit kesan dalam prestasi (mengalami penurunan sebanyak 1,96% di download, 2,35% di upload). Akhirnya, pelaksanaan sistem yang dibangunkan ini menggunakan bahasa Visual Basic.NET.

TABLE OF CONTENTS

UMP

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACL	Access Control List
CPU	Central Processing Unit
CSI/FBI	Computer Crime and Security Survey
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DMZ	demilitarised zone
DoS	Denial of Services
FRG	Filtering-Rule generalization
FTP	File Transfer Protocol
FTP	File Transfer Protocol
HIDS	host-based intrusion –detection system
HTTP	Hypertext Transfer Protocol
IDS	Intrusion detection system
IMAP	Internet Message Access Protocol
IP	Internetworking Protocol
IPX	Internetwork Packet Exchange
ISO	International Standards Organization
ISP	Internet service providers
Kbps	kilobits per second
LAN	Local Area Network
MAC	Media Access Control

Mbps	Mega bit per second
MLS LAN	Multi Level Secure LAN
NetBIOS	Network Basic Input/Output System
NIDSs	network intrusion detection systems
OSI	Open Systems Interconnect
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
UMAC	Universal Message Authentication Code
VPNs	Virtual Private Networks
XOR	exclusive-or

CHAPTER 1

INTRODUCTION

The Internet has arguably become the most diverse virtual entity ever developed by human kind. The Internet is the virtual common place where everyone is welcome to do business, communicate, research information, or simply enjoy surfing the net. The vastness of the Internet, along with the differences among its visitors, creates a most unique melting pot. However it also contains a great potential for misuse, abuse, and criminal activity (Strassberg, 2002).

The most famous computer network is the Local Area Network (LAN). This is true depending on the fact that most institutions are using this type of computer networks. As the nations without controlled borders will not assure the security and safety of their citizens, nor can they prevent piracy and theft. In addition, the security or privacy of stored data cannot be guaranteed and there is tendency that network resources will be exploited.

Securing a host-by-host basis grows more difficult, so it is not enough for securing information. A network security model enables the administrator to control the access to various hosts of network and services that hosts offer by using a network firewall at the server (Stallings, 1999). Many network firewalls are available. However, these conventional firewalls are focusing on attacks which come from the outside, e.g. Denial Of Services (DOS) attack.

Another type of attacks that come from internal user, known as internal intruding or internal attacks (Lakshmi and Agrawal, 2001). These attacks include attaching an unauthorized host to the network to deal with the other hosts and to disclose information to the outside. In addition, the attack can be a use of an unauthorized application, which is used either to disclose information to the outside network, or to control a host or even to control the whole network in an unauthorized manner. The use of such application can be done by a user, either the person intends it or not. This is based on a fact that not every person using the network is a specialist in a network security. Another possible attack is the use of an internal host by a person, which is not authorized. These internal intruding attacks are important but partially considered in the conventional firewall or even not taken into account at all. Insider attacks can lead to a big corruption.

1.1 BACKGROUND OF THE PROBLEM

Computer security is the process of preventing and detecting unauthorized use of the computer. Prevention measures help to stop unauthorized users (also known as "intruders") from accessing computer systems. Detection system helps to determine intruders and record their harmful activities. Computers are used in banking and investing, to shopping and communicating with others through email or chat programs. Although some users may not consider their communications "top secret," they probably do not want strangers reading their email, using their computer to attack other systems, sending forged email from their computer, or examining personal information stored on their computer (such as financial statements). Intruders (also referred to as hackers, attackers, or crackers) may not care about their identity. Often they want to gain control of their computer so they can use it to launch attacks to other computer systems.

Having control of the users' computer gives them the ability to hide their real location as they launch the attacks, often against high-profile computer systems such as government or financial systems. Even if the users have a computer connected to the Internet, to play the latest games or to send email to friends and family, their computer may be a target (Lowery, 2002).

Intruders are able to observe all the users activity on the computer, or cause damage to their computer by reformatting their hard drive or changing their data.

In the last few years, defending against the security threats which came from the internal network has become an important area of research. The internal intruding attacks are crucial but partially considered in the conventional firewall or even not taken into account at all. Internal attacks can lead to a big disaster. According to the annual CSI/FBI report, Computer Crime and Security Survey, the number of successful attacks from inside is roughly equal to numbers from outside. Therefore, protecting from outside attacks will cover only half the threats (Gordon et al., 2004; Strand, 2004).

Furthermore, Gartner Group estimates that over 70% of attacks that resulted in an economic loss to companies involved a company unauthorized insider. These statistics suggest that many companies may need to re-examine their security policies with a focus on internal security threats (Marks, 2004).

1.2 STATEMENTS OF PROBLEM

Protecting the network can be accomplished by many mechanisms. One of the most effective is the network firewall. However, firewall protects the network from external intrusion not from internal intrusion. Internal intruding or internal attacks can lead to disaster.   An attack launched from the internal means an attack launched from one of the hosts protected by the firewall. Internal intruding can be unauthorized host, application, and/or user connected hiddenly to the LAN and disclosed information to the outside. Most (LAN) firewalls perform an excellent job at protecting their networks from attacks launched from the Internet, while nothing much is done to protect their networks from internal intruding attacks which, launched from the inside (Strand, 2004).

1.3 RESEARCH OBJECTIVES

The aim of this study is to develop a new technique that can secure the network communication from internal intruders. The objectives of the study are as follow:

1. To protect local area network from internal attack.
2. To propose a new algorithm for protection against internal intrusion.
3. To compare and evaluate the performance of the new algorithm in a local area network environment.

1.4 SCOPE OF THE STUDY

The scope of this study focusing on enhances the security of the firewall with respect to internal attacks. This study is limited by several factors that have to be taken into consideration.

1. The algorithm was developed using Visual Basic.Net language and can only run in Windows environment.
2. Only the authorized computer can use the applications, the unauthorized computer will not be allowed.
3. The computers need to be on the same network in order to access the applications.

1.5 THESIS ORGANISATION

This thesis is organized in six chapters, including introduction in chapter one which represents a background, problem statement, scope of study and objectives. Chapter two discusses about all literature review on intrusion detection system (IDS), brief history of IDS and Networking. Chapter three explains in detail the algorithm that was implemented on a client/server model for the LAN. Chapter four includes system design and shared procedures, which describe the proposed system and shared procedures for both sides (client and server). Chapter five explains clients and server's implementation, discussions and results. Finally, conclusions and future work was described in chapter six.

UMP

CHAPTER 2

LITERATURE REVIEW

Intrusion is one of the popular areas in the information security landscape. Although the promise of technology that automatically detect alerts hostile intruder is extremely attractive, the technology is growing. Initially, intrusion detection systems were described as techniques used to decide whether or not to raise an alarm. As the number of systems grew, various classifications and taxonomies were produced to group these systems.

2.1 COMPUTER SECURITY

In early 1980s, individual workgroups in organizations have begun to use (LAN) technology to communicate and share resources. The first LANs were introduced into the academic world in the mid-1970s as the technology developed and major computer manufacturers adopted them. LANs provide means for meeting the requirements for high-speed, relatively short-distance communication among intelligent device (Tangney, 1988).

Most networks are organized as a series of layers or levels, each one built upon the one below it. The number of layers, the name of each layer, the contents of each layer, and the function of each layer differ from network to network. However, in all networks, the purpose of each layer is to offer certain services to the higher layers, shielding those layers from the details of how the offered services are implemented (Tangney, 1988, Shipley, 2001).

(Anderson, 1980; Denning, 1987) wrote in a technical report for a classified customer that audit records could be used to identify computer misuse. Taxonomy of threat classification was built and a suggestion to improve upon audit subsystems was offered so that they can be used to detect misuse.

(Balmer, 1999) described the physical interfaces needed to make a trusted path between clients and server. A detailed analysis of methods of a client PC in the Multilevel secure LAN with high assurance of properly controlled object reuse and operating system integrity was presented. Similarly, Susan and Scott (Bryer and Heller, 1999) provided the initial design and proof of concept implementation for a secure LAN that supported the extension of the trusted computing base to commercial grade personal computers.

(Rossetti, 2000) developed a trusted processes tool that allows the administrator to easily set up Internet Message Access Protocol (IMAP) mail boxes for each LAN user and group account at multiple security levels.

(Bora, 2000) described the hardware and software design for a custom plug-in board that can both successfully complete the trusted path connection and control the client PC. This enabling the Multilevel secure LAN to extend the trusted computing base from the high assurance server to a commercial Personal Computer (PC).

(Wilson, 2000) presented a framework of communications protocols that will enable the components of the Multi Level Secure LAN (MLS LAN) to security interact. The framework first presents a communications channel protocol that protects all data transmitted on the network. Following that, three other protocols are described that enable MLS LAN users to safely login and negotiate a secure session, access application protocol servers that provide services such as E-mail or World Wide Web (WWW) services, and to use typical LAN based office automation service.

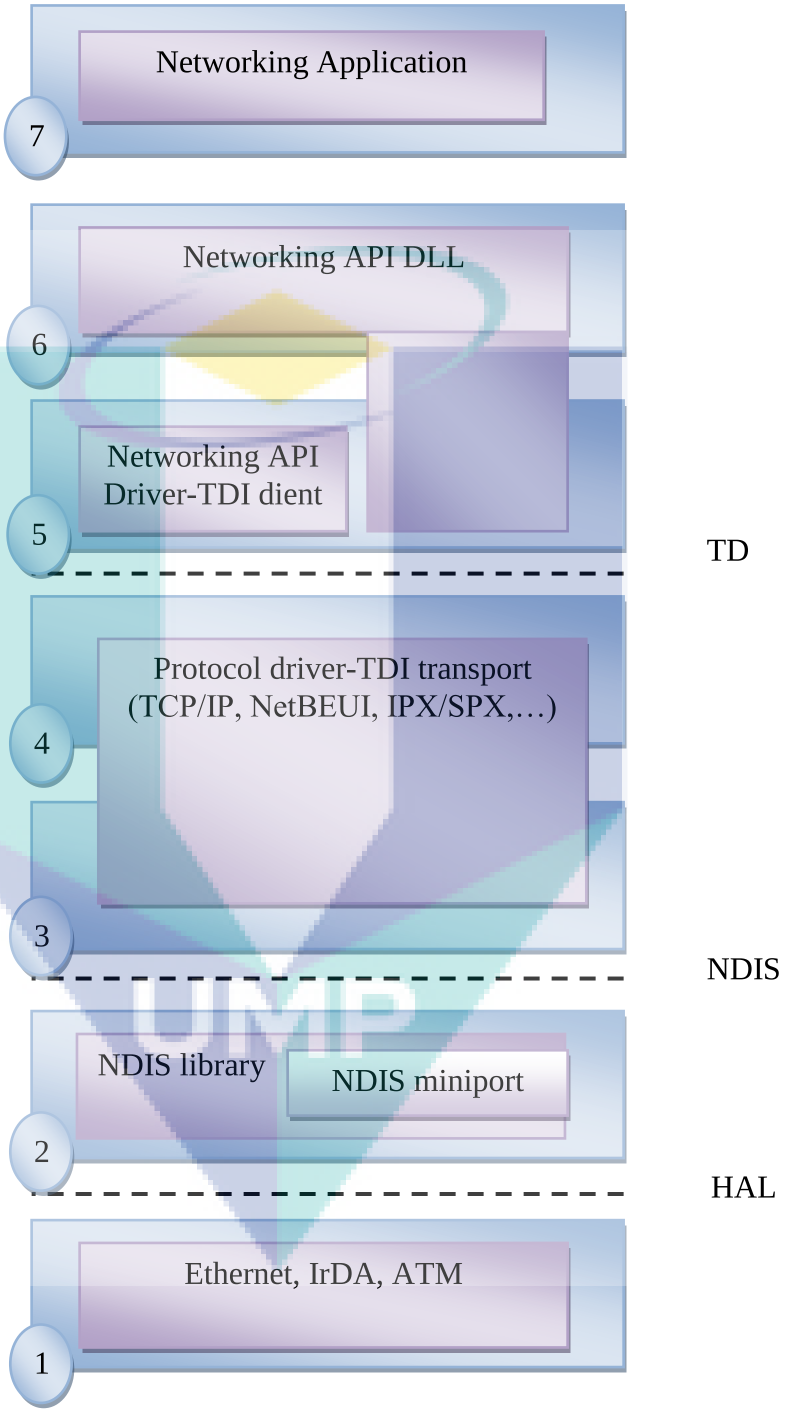
(Moller and Donbaek, 2001) introduced a modification to the open source Linux kernel operating system in order to send application identifier and a host identifier and user identifier (read from text file) and send these information to the verification server.

(Irvine et al., 2004) described a high assurance architecture system, named Monterey security architecture (MYSEA). MYSEA provides a trusted distributed operating environment for enforcing multilevel security policy. MYSEA introduces a trusted path between the clients and the server and a multilevel security for LAN clients.

Windows Networking Architecture is made of number of layers with well-defined interfaces between them. This makes it possible for modules from different vendors to work together and lets the user install new modules that provide added functionality (Microsoft Corporation, 2001).

Figure 2.1 shows the most important parts of the windows networking architecture and how they are related; it also shows how each component fits into the OSI reference model. The mapping between OSI layers and networking components are not precise which the reason that some components cross layers is (Forouzan, 2003).

UMP



UMP

Figure 2.1: Windows networking component and OSI layers

Networking communication begins when a network application program attempts to access resources on another computer. The data cross many components modules (layers) until they reached wire.

2.1.1 Basic Security Issues

The security of network has four fundamental designed objectives to protect the data and the network's resources. These objectives are (Fisch and white, 2000):

i. ***Confidentiality:*** ensuring that authorized individual does not gain access to data contained on a source of the network.

ii. ***Availability:*** ensuring that authorized users are not unduly denied access or use of any network access for which they are normally allowed.

iii. ***Integrity:*** ensuring that unauthorized individuals related to this do not alter data authenticity, that it is concerned with the unauthorized modification.

iv. ***Usage:*** ensuring that the resources of the network are reserved for use only by authorized users in appropriate manner.

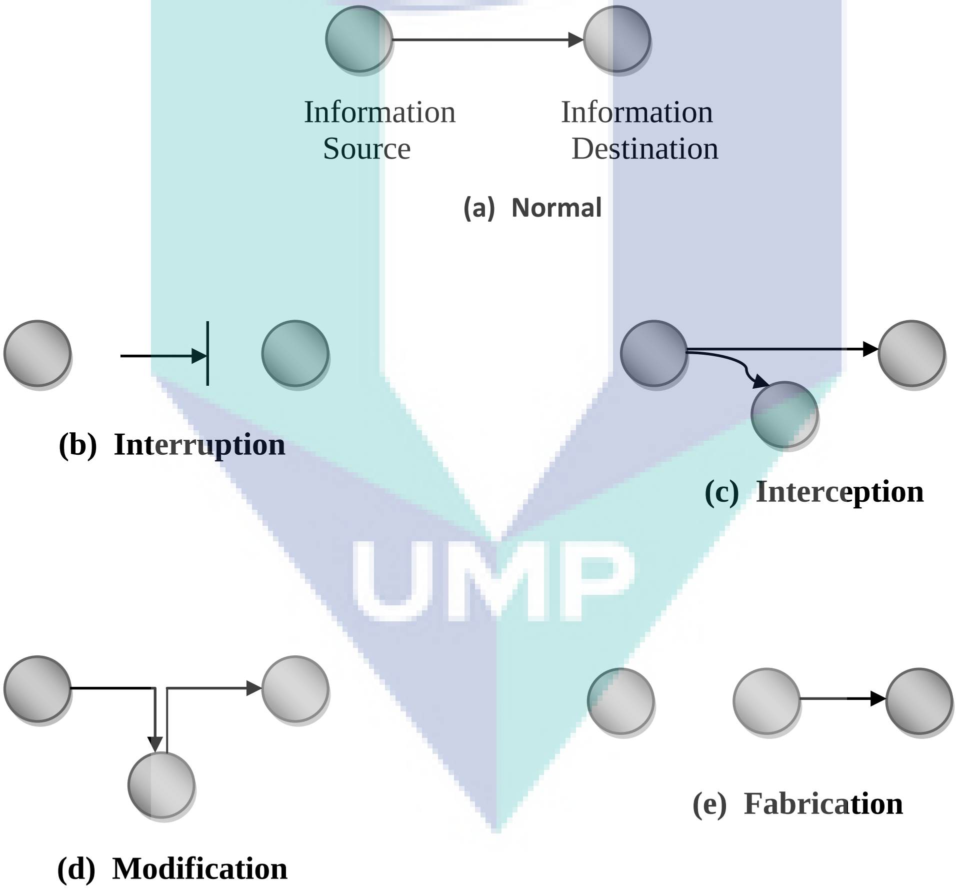
Attacks on the security of a computer system or network are best characterized by viewing the function of the computer system as providing information. The threats can be described in terms of how they affect the normal flow of information in the network. There are four basic patterns of attack for these threats as depicted in Figure 2.2: and the following are categories of attack (Khazal, 2004):

i. ***Interruption:*** An asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability, which include destruction of a piece of hardware, such as a hard disk, the cutting of a communication line, or the disabling of the file management system.

ii. ***Interception:*** An unauthorized person gains access to an asset. This is an attack on confidentially the unauthorized party could be a person, or a program, or a computer. For examples, wiretapping to capture data in a network and unauthorized copying of files or programs.

iii. ***Modification:*** An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. For examples, changing values in a data file, alerting a program so that it performs differently, and modifying the content of messages being transmitted in a network.

iv. ***Fabrication :*** An unauthorized party inserts counterfeit object into the system. This is an attack on authenticity. Examples include the insertion of spurious messages in a network or the addition of records to a file.



UMP

Figure 2.2: Security threats

2.1.2 Firewall

A firewall is a security guard placed between a private network and the outside Internet that monitors all incoming and outgoing packets. The function of a firewall is to examine every packet and decide whether to accept or discard it based upon the firewall's policy. This policy is specified as a sequence of (possibly conflicting) rules. When a packet comes to a firewall, the firewall searches for the first rule that the packet matches, and executes the decision of that rule (Alex, and Eric, 2007).

Modern firewalls are able to work in conjunction with tools such as intrusion detection monitors and email/web content scanners for viruses and harmful application code. But firewalls alone do not provide complete protection from Internet-borne problems. As a result, they are just one part of a total information security program. The work of firewall shown in Figure 2.3: (Tihomir and Predrag, 2007).

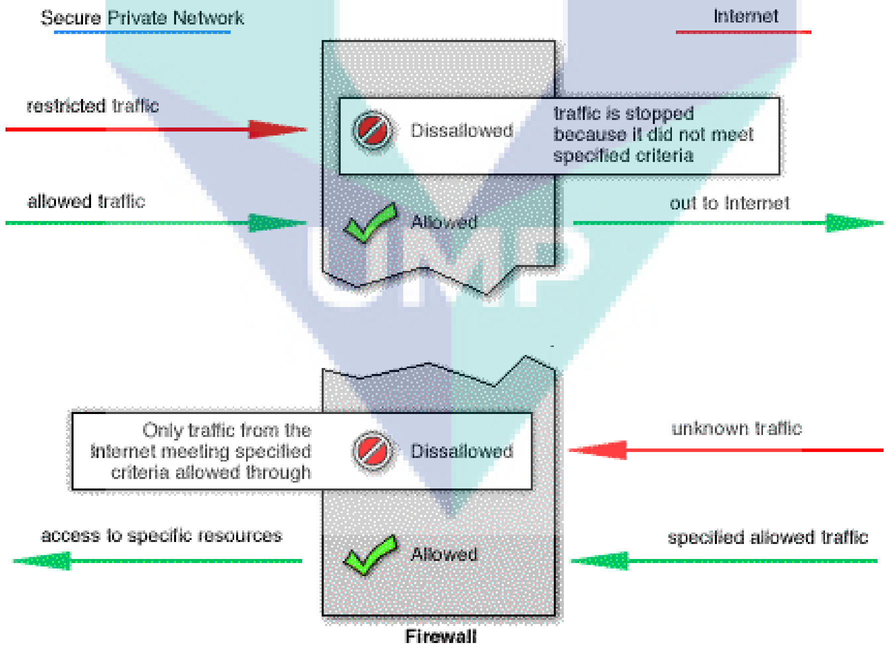

Secure Private Network
Internet
restricted traffic
Dissallowed
traffic is stopped because it did not meet specified criteria
allowed traffic
Allowed
out to Internet
Only traffic from the Internet meeting specified criteria allowed through
Dissallowed
unknown traffic
access to specific resources
Allowed
specified allowed traffic
Firewall

Figure 2.3: Basic Firewall Operation

UMP

Generally, firewalls are viewed as the first line of defense. However it may be better to view them as the last line of defense for an organization. Organizations should still make the security of their internal systems a high priority. Internal servers, personal computers, and other systems should be kept up-to-date with security patches and anti-virus software (Cutler and Pole, 2002).

Firewall is a computer, router or other communication device that filters access to the protected network. (Tihomir and Predrag, 2007) define a firewall as a collection of components or a system that is placed between two networks.

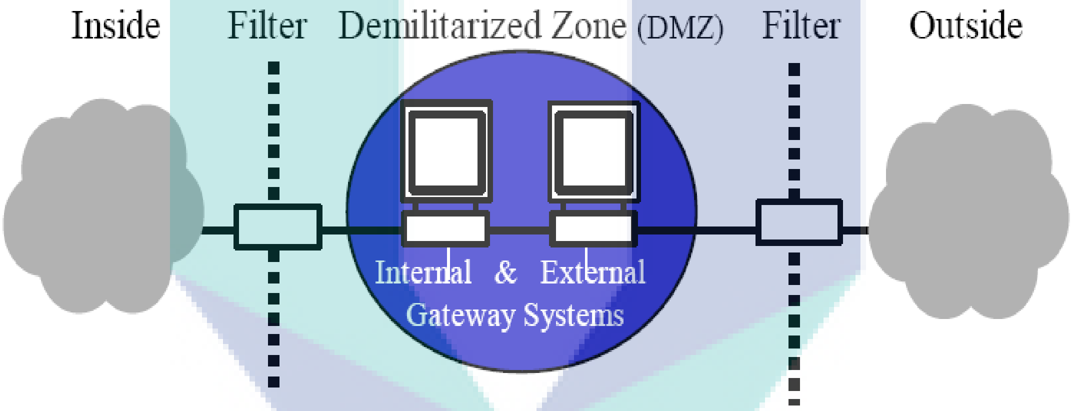


Figure 2.4: Firewall Schematics

UMP

Such traditional network firewalls prevent unauthorized access and attacks by protecting the points of entry into the network. As Figure 2.4: shows (Tihomir and Predrag, 2007). A firewall may consist of a variety of components including host (called bastion host), router filters (or screens), and services. A gateway is a machine or set of machines that provides relay services complementing the filters. Another term illustrated in the diagram is "demilitarised zone or DMZ". This is an area or sub-network between the inside and outside networks that is partially protected. One or more gateway machines may be located in the DMZ.

Exemplifying a traditional security concept, defence-indepth, the outside filter protects the gateway from attack, while the inside gateway guards against the consequences of a compromised gateway. Depending on the situation of the network concerned, there may be multiple firewalls, multiple internal networks, Virtual Private Network(VPN)s, Extranets and perimeter networks. There may also be a variety of connection types, such as TCP and UDP, audio or video streaming, and downloading of applets (Habtamu, 2000).

Firewalls have the following characteristics (Stallings, 1999; Stallings, 2000):

1. All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall, where various configurations are possible.
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.
3. The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system.

Firewalls also have their limitations, including the following (Stallings, 1999; Stallings, 2000):

1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to the outside.
2. The firewall does not protect against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.

The firewall cannot protect against the transfer of virus-infected programs or files. Because of the variety of operating systems and applications supported inside the perimeter, it would be impractical and perhaps impossible for the firewall to scan all incoming files, e-mail, and messages for viruses.

Some of the most powerful firewall software on the market is designed to run on an ordinary computer probably a dedicated server if people are securing a large network. Other firewall software is designed to run on proprietary hardware that user have to buy along with the software, turning the bundle into a "security appliance." As a general rule, appliances are faster, easier to install and operate and also more expensive. But there is no guarantee that an appliance will do a better job than a software-only firewall. Software firewalls tend to be more flexible, and it is easier to upgrade the hardware it is running on (Tihomir and Predrag, 2007).

There are several types of firewall techniques:-

i. Packet filter

Packet filter looks at each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.

ii. Application gateway

Application gateway applies security mechanisms to specific applications, such as FTP and Telnet servers. This is very effective, but it can impose performance degradation. Circuit-level gateway applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.

iii. Proxy server

Proxy server intercepts all messages entering and leaving the network. The proxy server effectively hides the true network addresses.

2.1.3 Types of Networking Attacks

There are four major categories of networking attacks. Every attack on a network can be placed into one of these (Pachghare et al., 2009).

i. Denial of Service (DoS): A DoS attacks is a type of attack in which the hacker makes a memory resources too busy to serve legitimate networking requests and hence denying users access to a machine e.g. apache, smurf, Neptune, ping of death, back, mail bomb, UDP storm, etc.

ii. Remote to User attacks (RlL): A remote to user attack is an attack in which a user sends packets to a machine over the Internet, and the user does not have access to in order to expose the machines vulnerabilities and exploit privileges which a local user would have on the computer, e.g. xlock, guest, xnsnoop, phf, sendmail dictionary etc.

iii. User to Root Attacks (U2R): These attacks are exploitations in which the hacker starts off on the system with a normal user account and attempts to abuse vulnerabilities in the system in order to gain super user privileges, e.g. perl, xterm.

iv. Probing: Probing is an attack in which the hacker scans a machine or a networking device in order to determine weaknesses or vulnerabilities that may later be exploited so as to compromise the system. This technique is commonly used in data mining, e.g. satan, saint, portsweep, mscan and nmap.

There are different techniques for protecting networks from attacks such as:-

i) The Data Encryption Standard (DES)

In the 1970, Data Encryption Standard (DES) was developed by the National Bureau of Standards with the help of the National Security Agency. The purpose was to provide a standard method for protecting sensitive commercial and unclassified data. LUCIFER was the first draft of the algorithm which was created by IBM. In November of 1976, DES officially became a federal standard (Daley, 1999).

Generally, DES performs two operations on its input, bit shifting, and bit substitution. The key controls exactly how this process works. By doing these operations repeatedly and in a non-linear manner, it end up with a result which cannot be used to retrieve the original without the key. The most likely subject to cipher text only attacks is the users of Enigma. For this type of attack, the cryptographer has access only to encrypted documents. Therefore, under such conditions there is no known method of attack better than randomly guessing keys (Vimalathithan and Valarmathi, 2009). The algorithm was designed to encipher and decipher blocks of data consisting of 64 bits under the control of a 64-bit key. Deciphering must be accomplished by using the same key as for enciphering (Hombrebueno, et al., 2009).

Complicated logical functions, such as various types of permutations, XOR and SHIFT functions are used in DES algorithm. Since the key employed is transformed to mentioned function, by following the algorithm provided, the only way to decrypt the plain text is to apply the same key in decryption algorithm (Taherkhani et al., 2010).

ii) Universal Message Authentication Code (UMAC)

Message Authentication Codes (MACs) are widely used in communication networks. In this situation, the parties share a secret key and the channels are assumed to be insecure. Normally, the communicated messages are lengthy which in turn necessitates the existence of fast MACs. Moreover, in many networks, such as sensor networks, messages are sent very frequently. Therefore, there is a need for MACs that take multiple messages at the same time and generate a single tag for all the messages in efficient time (Shaw et al., 2010).

The UMAC algorithm specifies how the message, key, and nonce determine an authentication tag. The sender will need to provide the receiver with the message, nonce, and tag. The receiver can then compute what should be the tag for this particular message and nonce, and see if it matches the tag actually received. The receiver might also wish to verify that the nonce has not been used already; doing this is a way to avoid replay attacks.

iii) Access Control List (ACL)

Access Control Lists (ACLs) is a list of permissions attached to an object. An ACL specifies which users or system processes are granted access to objects which includes what operations are allowed on given objects. Each entry in a typical ACL specifies a subject and an operation. Among the various options for implementing Internet packet filters in the form of ACL is the intuitive, but it is potentially crude method of processing the ACL rules in sequential order. ACL is then a sequence of such rules designed to implement a given objective or set of objectives.

(Grout et al., 2007) used ACLs for security purposes, which is simply to pass or block packets, or as filters for more sophisticated policies such as traffic shaping, address translation, queuing or encryption. Moreover, ACL has been used to enhancement algorithm by reducing part of its complexity. Although the simplification involved leads to an instantaneous lack of accuracy, the long term trade-off between processing speed and performance can be seen, through experimentation, to be positive.

When a subject requests an operation on an object in an ACL-based security model the operating system first checks the ACL for an applicable entry to decide whether the requested operation is authorized. A key issue in the definition of any ACL-based security model is determining how access control lists are edited, namely which users and processes are granted ACL-modification access. ACL models may be applied to collections of objects as well as to individual entities within the system hierarchy (Chow et al., 2005.)

The ACL assignment, a rule may consist of up to five parts i.e. the permit or deny type, the protocol, a source address, destination address and a flag function (as in the echo-reply parameter above) for fine-tuning (Grout and Davies, 2010). Each parameter may be a single value or a range of allowable matches. Inefficiently implemented ACLs can add significantly to packet delay and even small ACLs will contribute to this latency simply by their aggregation across several routers.

2.2 PROTOCOLS MODELS

A protocol is an agreement or rules between the communicating parities on how communication is to proceed, there are some common protocols like IPX, TCP /IP, and Open Systems Interconnection model **(**OSI), most protocols actually consist of several protocols grouped together in a suite.

2.2.1 TCP/IP Model

Transmission Control Protocol / Internet Protocol (TCP/IP) network model defines a set of rules to enable computers to communicate over the network, specifying how data should be packaged, addressed, shipped, routed and delivered to the right destination. The TCP/IP family uses four layers while ISO/OSI uses seven layers. The TCP/IP and ISO/OSI systems differ from each other significantly, although they are very similar on the network and transport layers. In the TCP/IP model, each layer has its own functionality and services.

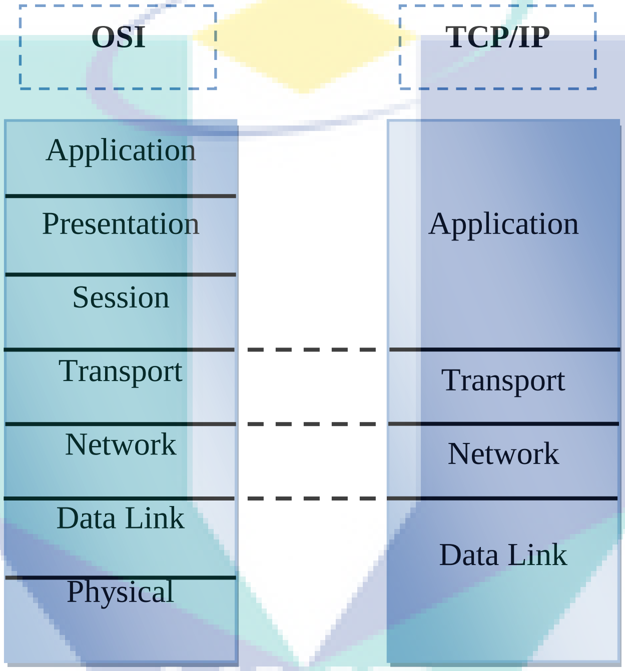
For example, the application layer is used by most programs for network communication. The transport is responsible for end-to-end message transfer capabilities independent of the underlying network, along with error control, fragmentation and flow control. Accordingly, each layer has its own attacks and challenges, which means each layer needs a specific protection process (Forouzan, 2003).

2.2.2 OSI Model

The International Standards Organization (ISO) is a multinational institute dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection (OSI) model. It was first introduced in the late 1970s. An open system is a set of protocols that allows any two different systems to communicate without requiring changes to the logic of the underlying hardware and software (Safaa and Fakhri, 2009).

The OSI model is not a practical protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable (Forouzan, 2003).

The OSI model is a layered framework. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network, Figure 2.5: describes the TCP/IP and OSI layers model.



UMP

Figure 2:5: TCP/IP and OSI layers model

2.3 TCP/IP PROTOCOL SUITE

TCP/IP (Transmission Control Protocol/Internet Protocol) communication protocol is assumed, so that all traffic in the network uses the IP protocol. Any IP-datagram sent over the network contains the sender and receiver IP-addresses, which uniquely identify the sending and the receiving hosts on that network.

Normally, the firewall knows which host sent any given IP-datagram based on the IP-address of the sender (Stallings, 1997). Unfortunately, the internal adversary is capable of faking the IP-address. This means that he can impersonate any host on the trusted network. Moreover, the IP protocol contains neither information about which application sent a given network packet, nor anything about the user that started the application.

Thus, the firewall is unable to distinguish between legal and illegal network packets based solely on the information available in the IP-protocol. So, it needs more information. TCP/IP is still a very capable communication protocol suite, and it is not likely to be replaced, at least not within the next few years.

The proposed firewall must ensure that no information leaves the trusted network, except when the communication link originates from a trusted source (a trusted host running a trusted application, under trusted user). This means that the protected computers by the extended firewall must provide the extended firewall with these extra information in every single packet destined to going out. The process of finding and adding these information to each packet and verifying of these information requires many changes to the operating systems of all hosts (clients and server), on the protected network, that need to communicate through the firewall.

The adversary is capable of hijacking an established communication link, due to his physical access to the trusted network (Moller and Donbaek, 2001). Also, providing each packet with just a User ID, Application ID and Host ID is a very naive approach. The internal adversary is capable of sniffing the trusted network and sending/modifying packets, so duplicating the credentials of a valid packet is easy for the attacker to do. So, strong authentication method is a vital process.

The TCP/IP protocol suite is made of five layers (Anderson and Karlsson, 2000) physical, data link, network, transport, and application. The first four layers provide physical standards, network interface, Internetworking, and transport functions that correspond to the first four layers of the OSI model. The three top most layers in the OSI model, however, are represented in TCP/IP by a single layer called the application layer, as shown in Figure 2.6:

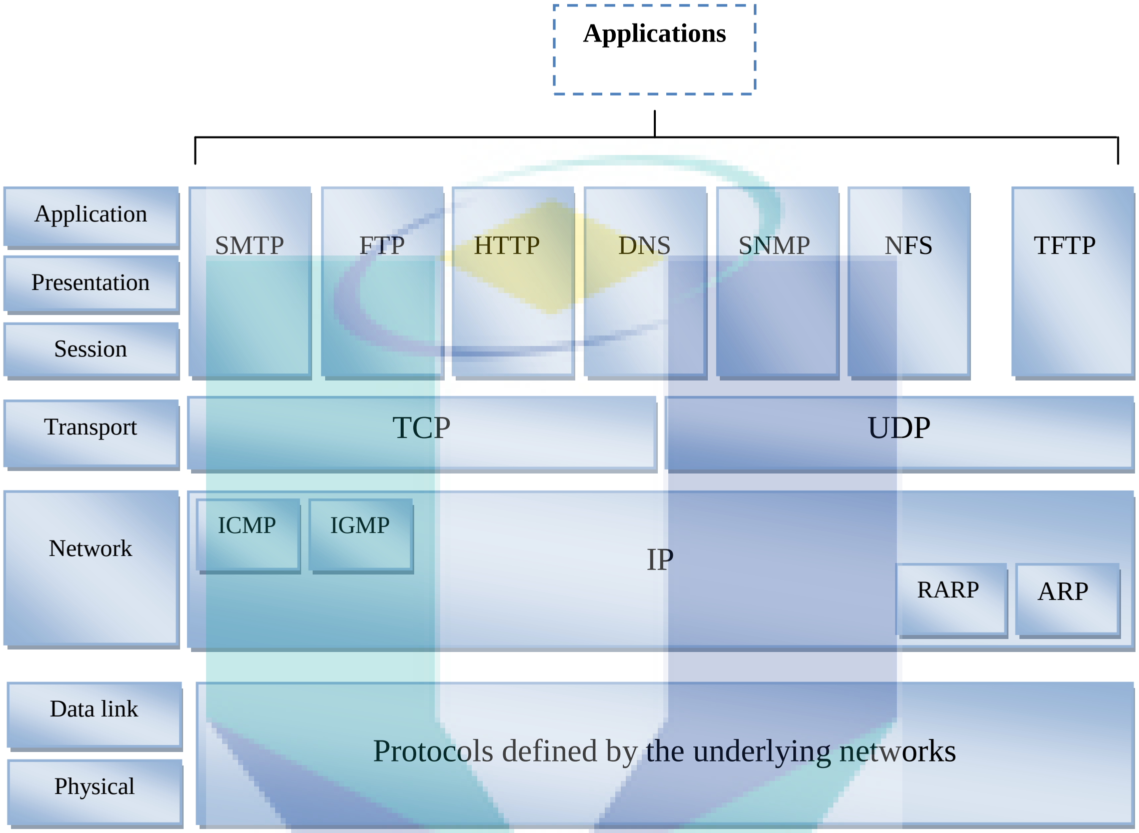


Figure 2.6: TCP/IP and OSI model

UMP

To transmit data across a layered network the data should pass from the application to a top layer on a protocol stack. After that the layer finishes its operation on that data, it passes the data to the next lower layer on the stack (Russinovich and Solomon, 2004). As shown in the Figure 2.7: the layers in the stack encapsulate the data for the next lower level in the stack, as the data passes through' each layer. Encapsulation, therefore, is the process of storing data in the format required by the lower level protocol in the stack (Forouzan, 2003).

TCP/IP is a hierarchical protocol made up of interactive module. The term hierarchical means that each upper level protocol is supported by one or more lower level protocols (Forouzan, 2003). Each of which provides a specific functionality, but the modules are not necessarily interdependent (Forouzan, 2003). Whereas the OSI model specifies which functions belong to each of its layers, the layers of the TCP/IP protocol suite contains relatively independent protocols that can be mixed and matched depending on the needs of the system (Anderson and Karlsson, 2000).

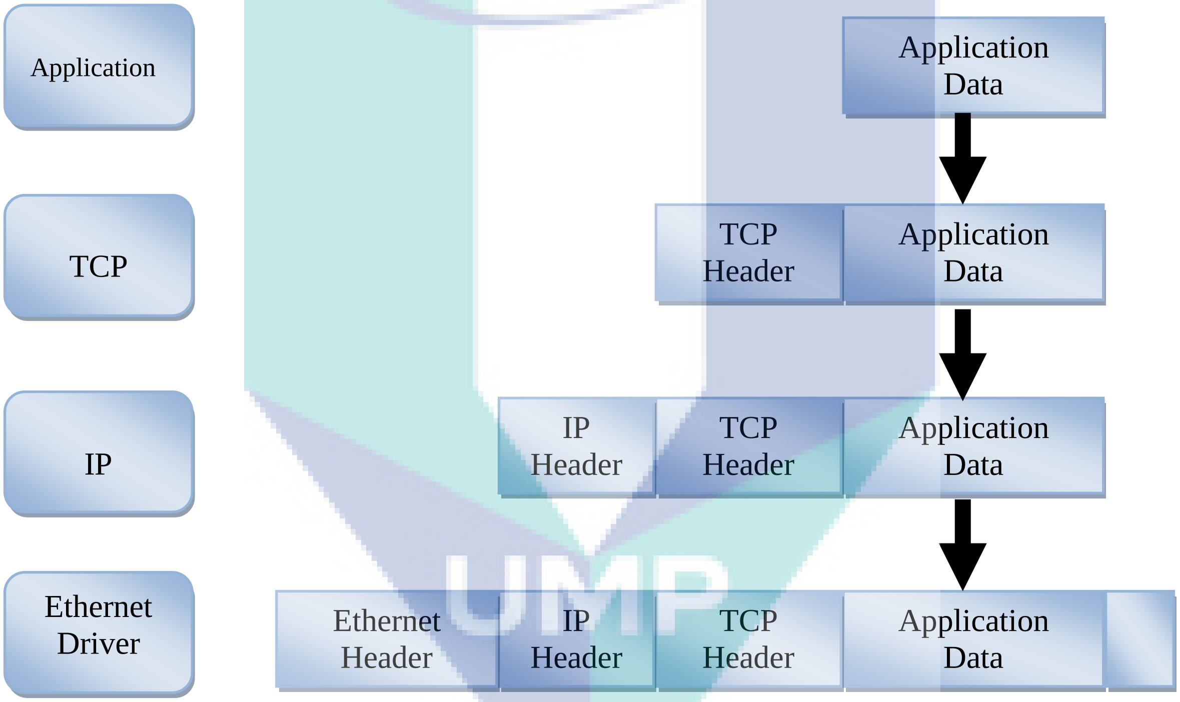


Figure 2.7: Data Encapsulation in to the protocol layers

The application module at the sender host will, therefore, encapsulate data from the user in an application message. TCP module encapsulates the application data and attaches the TCP header and sends it to the next layer. As the data passes through IP module in the network layer, it formats the TCP segment into an IP datagram or packet. The Ethernet driver formats the data from the IP module and places the data into an Ethernet frame.

UMP

This explains how a frame encapsulates an IP datagram and further how the IP packet encapsulates TCP/UDP data. A reverse process will be happened at the receiver host in order to make two applications communicate between them.

2.3.1 Internetworking Protocol (IP)

The Internetworking Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless datagram protocol -a best-effort-delivery service. The term "best-effort" means that IP provides no error checking or tracking (Forouzan, 2003) IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees (Zaman and Karray, 2009).

A packet in the data link layer is called Ethernet packet. In order to identify an IP packet, the Ethernet frame structure must be known. Looking at the Figure 2.8: the frame data field will contain an IP packet when the frame type field has a value 0800 (Bentham, 2000).

The limited functionality of IP should not be considered a weakness. IP provides back-bone transmission functions that enables the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency (Forouzan, 2003).

If reliability is important, IP must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would be recipient to discover the loss and rectify the problem. The post office itself does not- keep track of every letter and cannot notify a sender of loss or damage (Forouzan, 2003).

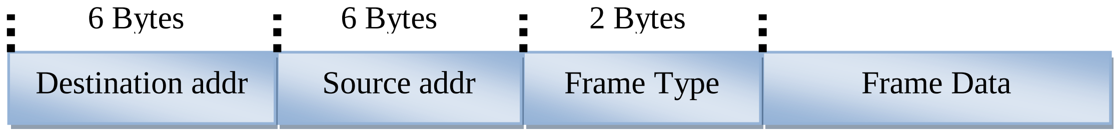


Figure 2.8: Format of an Ethernet Frame

Where the destination address is a six bytes destination Media Access Control (MAC) address, and the source address is a 6 bytes source MAC address.

2.3.2 Types of Attacks on TCP:

Designing a protocol that is resistant to attacks is a challenging task because of traffics. TCP/IP was designed to be an open protocol. A node running TCP/IP or UDP/IP is listening for in bound network traffic from almost anywhere. Some initial trust design has led to some bothersome attacks. The most popular attack are (Anderson and Karlsson, 2000):

1. **Address Impersonation**

Address impersonation is a threat to applications running on the TCP Protocol. The TCP protocol is slightly more difficult to impersonate than UDP because TCP provides flow control reliable delivery and consequently contains facilities in the protocol to detect anomalous conditions. TCP packets contain a sequence number that makes impersonation a little harder.

2. **Sequence Number Guessing**

The protocol of TCP uses sequence number sends acknowledgments to reliably converse with other stations on the network. A clever hacker can exploit a TCP connection during the initial handshake for the protocol if sequence numbers can be guessed. The favored choice is for the hacker to spend some time gathering information about sequence numbers chosen by the target various connections. Network traffic sniffing is useful here but not a necessity, because if the victim is on a public network, the attacker can send it as many TCP connections attempts, as he likes.

3. Session Hijacking

If the socket addresses and sequence numbers are known, a node that is in between the endpoints of a TCP connection can hijack one or both halves of the session. Sometimes this attack also is referred to as the bucket brigade attack. All the impostor must act is ensuring that the two endpoints receive the appropriate protocol messages during the hijack. Because the attacking node is in the middle, intercepted packets can be easily altered, discarded or substituted.

2.4 INTRUSION DETECTIONS SYSTEM (IDS)

Intrusion detection is the process of identifying and responding to suspicious activities targeted at computing and communication resources, and it has become the mainstream of information assurance as the dramatic increase in the number of attacks (Ying L., et al. 2010).

The concept of intrusion detection appears in two basic approaches. The first approach called anomaly detection, it aims to define and characterize the correct static form or acceptable dynamic behavior of the system, and then to detect wrongful changes or wrongful behavior (Anderson, 1980; Pachghare et al., 2009).

The second approach called misuse detection, involves characterizing known ways to penetrate a system. Each one is described as a pattern. Pattern takes a variety of forms, it may be a static bit string like virus bit string insertion, or may describe a suspect set or sequence of actions (Denning, 1987; Krsul, 1997).

(Yurcik, 2002) introduced a new class of attack against a network signature-based Intrusion detection system (IDS) which was tested using SNORT and it was called “Squealing”. This vulnerability has significant implications since it can be generalized to any IDS, while signature-based IDSs have implementation problem with high false positive rates that require tuning. It was shown that a more serious general vulnerability in that packets can be crafted to match attack signatures such that alarms can be selectively triggered allowing a target IDS to be externally controlled by malicious attacker.

(Uribe and Cheung, 2004) had given a network that deploys multiple firewalls and network intrusion detection systems (NIDSs), ensuring that these security components are correctly configured which was a challenging problem. This paper presented an integrated, constraint-based approach for modeling and reasoning about these configurations.

(Golnabi and Al-Shaer, 2006) presented a set of techniques and algorithms to analysis and manage firewall policy rules. The techniques are Data Mining technique to deduce efficient firewall policy rules by mining its network traffic log based on its frequency, Filtering-Rule generalization (FRG) to reduce the number of policy rules by generalization, and a technique to identify any decaying rule and a set of few dominant rules, to generate a new set of efficient firewall policy rules.

(Tihomir and Predrag, 2007) Predrag Pale presented one approach to rule optimization solutions for improving firewall performance. The new software solution has been developed based on relations between rules. Its main purpose is to remove anomalies in ordering of Linux firewall rules and to merge similar rules

Moses Garuba, Duane Fraites (Garuba et al., 2008) analyzed several organizational require security systems that are flexible and adaptable in order to combat increasing threats from software vulnerabilities, virus attacks and other malicious code, in addition to internal attacks, in order to determine the network intrusion detection system that effectively meets these objectives. Through conclusive analysis of the study, heuristic based systems were better served to meet the organizational objectives than signature based systems. Intrusion detection systems have been built to explore both approaches, anomaly detection and misuse detection.

(Richardson, 2008), CSI Computer Crime and Security Survey conducted on 522 computer security practitioners concluded that the average financial cost of fraud to a company was US $500,000 per year.

2.4.1 Anomaly Detection

The anomaly detection or behavior-based intrusion detection must be able to distinguish between the anomaly and normal. Anomaly detection divide into static and dynamic (Jones and Sieken, 2000).

A static anomaly detector is based on the assumption that there is a portion of the system being monitored that should remain constant. Static portion of the system can be represented as a binary bit string or set of such strings (files). If the static portion of the system ever deviates from its original form, either an error occurs or the intruder alters the static portion of the system. Static anomaly detector are said to check for data integrity (Shipley, 2001; Shanbhag and Wolf, 2008) as shown in Figure 2.9:

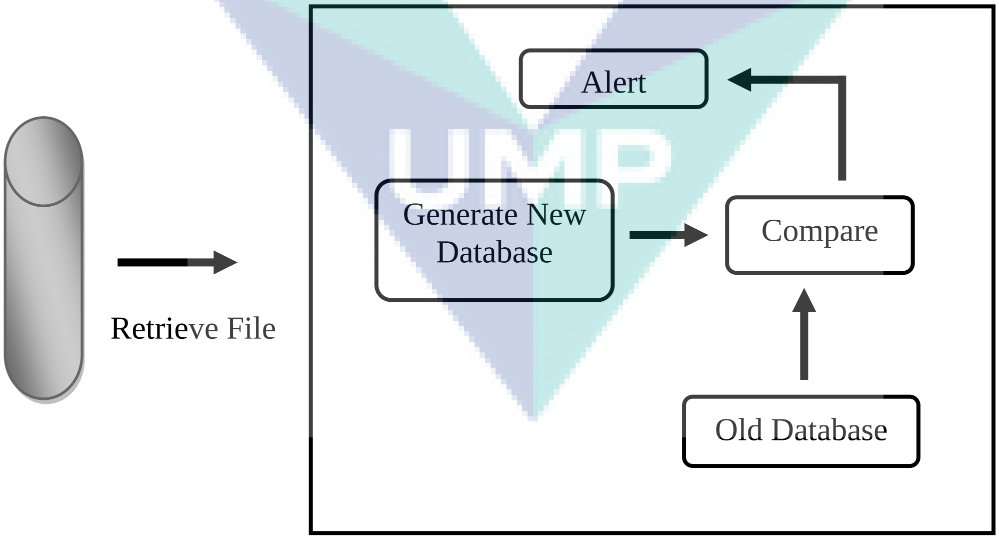


Figure 2.9: Operations of Static Anomaly Detector

UMP

Static anomaly detector archive a representation of system state, perhaps compressed. Periodically, the static anomaly detector compares the archived state representation to a similar representation computed based on the current state of the same static bit strings. The compressed representation is called a signature, it is a "summary" value computed from a base bit string. The computation is designed so that a signature is computed from a different base string, with high probability have a different value. Signature includes checksums, message digest algorithms and hash functions (Krsul, 1997; Shanbhag and Wolf, 2008).

Dynamic anomaly detection requires distinguishing between normal and anomalous activity. Dynamic anomaly detection system typically creates a base profile to characterize normal, acceptable behavior as shown in Figure 2.10. A profile consists of a set of observed measures of behavior for each of a set of dimensions. Frequently used dimension includes prefererred choices, (e.g log-in-time, login-location), resource consumed or cumulatively per unit time (e.g length of interactive session), or representative sequences of actions. Dimension may be specific to the type of the entity which behavior is associated.

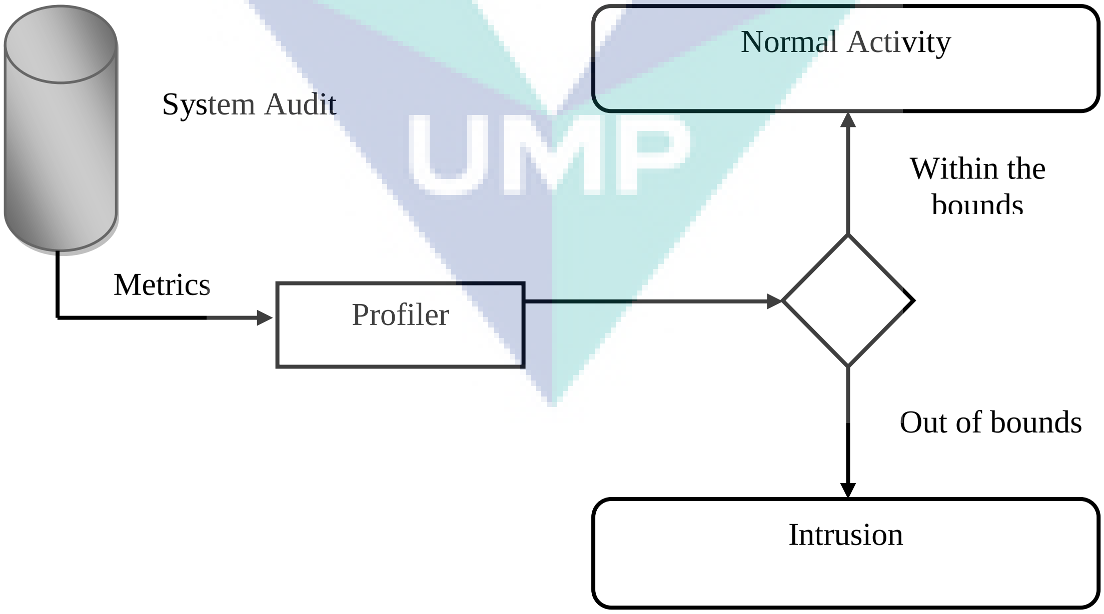


Figure 2.10: Operations of Dynamic Anomaly Detector

Typical entity is users, workstations as in NIDES (Lunt, 1993) or even application as in SRI Safeguard (Petrovic and Bakke, 2008). An intrusion detection system develops a unique base profile (typically based on observed behavior) for each individual entity that it recognizes. Dynamic detectors are similar to the static detectors in that they monitor behavior by comparing current characterization of behavior to the initial characterization of expected behavior (base profile) which seek diverge. To capture events or actions of interest, dynamic anomaly detector either depends on audit records, or record a database specific for intrusion detection.

2.4.2 Misuse Detection

Misuse detection or knowledge-based intrusion detection is concerned with catching intruders who attempt to break into the system using some known techniques. Misuse detection is based on the premise that all intrusions have a distinct signature that can be detected. It maintains a collection of attack signatures and monitors the system for an attack. If the activity matches a signature, then the system reports an intrusion (Petrovic and Bakke, 2008; Pachghare et al., 2009) as shown in Figure 2.11:

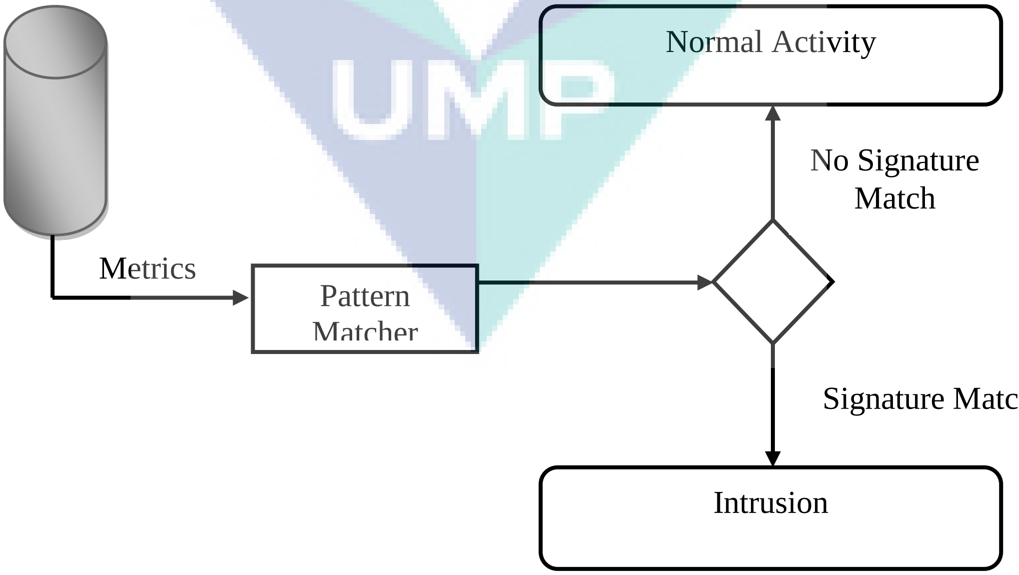


Figure 2.11: Operations of Misuse Detection

UMP

2.4.3 Comparison between Anomaly Detection and misuse detection

In general, anomaly detection describes normal usage, whereas misuse detection defines intrusive usage (Lunt, 1993). Figure 2.12: describes the two methods and the differences between them. In anomaly detection, usage outside the defined normal usage is reported. As normal usage is hard to define, some of the reported events are allowed and therefore considered as being false alarms. Intrusive behavior might be defined as normal if the methods used are inaccurate. Intrusion with such behavior will remain undetected with anomaly detection. Misuse detection has few, if any, false alarms as the patterns describe misuse. Only known intrusion can be defined which implicates those new types of intrusion will not be detected.

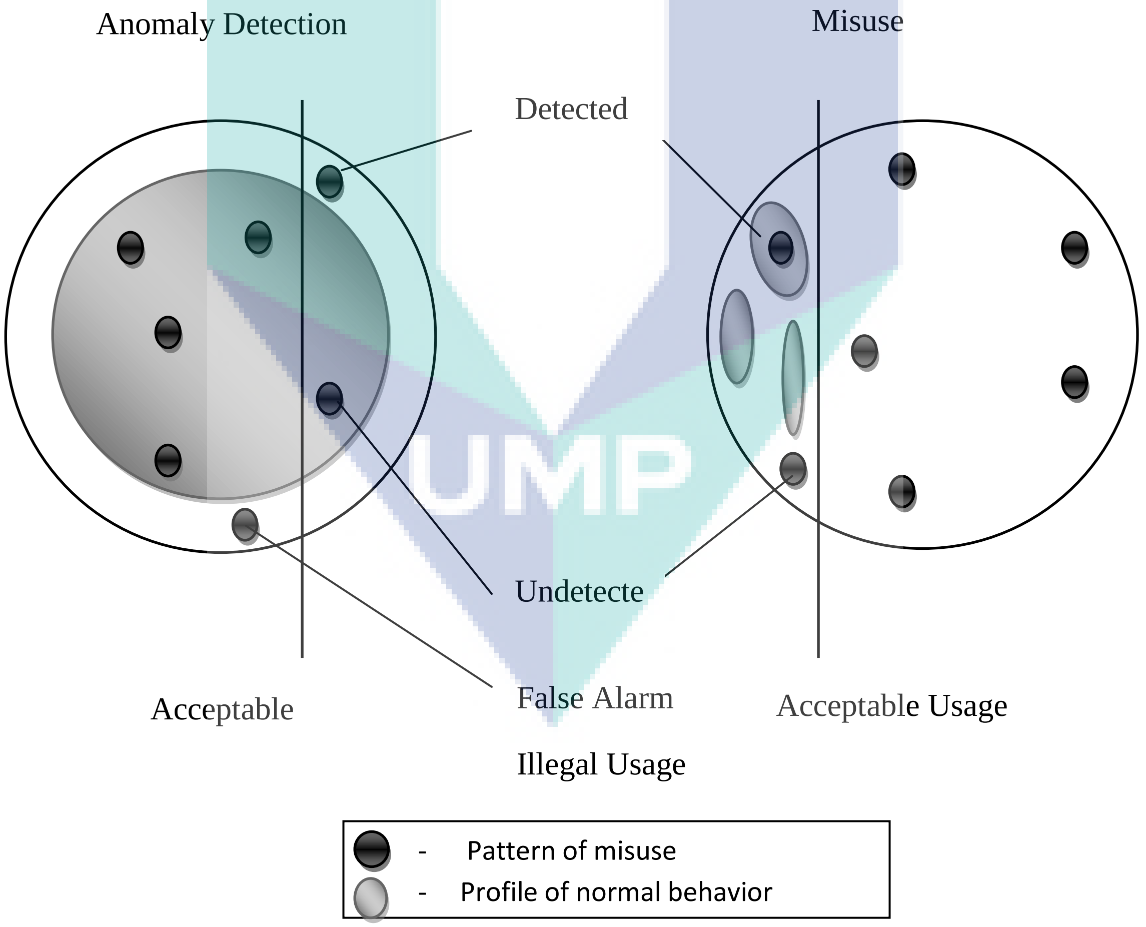


Figure 2.12: Comparison of Anomaly and Misuse Detection

Table 2.1 summarizes the advantages and disadvantages of each intrusion detection approach. In order to balance the advantages and disadvantages of each approach, an implementation of the combined approaches is preferred. The developed models in this work adopt this concept (Mark and Benjamenn, 2001).

Table 2.1: Advantages and disadvantages of detection techniques

Approach	Advantages	Disadvantages
Anomaly	• Effective against novel and unknown attacks. • Can detect abuse of privileges.	• High numbers of false alarm. • Behavior of the profile can become intrusive, which will not be detected as anomalous.
Misuse	• Few False alarms.	• Cannot detect new attacks. • Detection of abuse of privileges is difficult.

2.5 TYPES OF INTRUSION DETECTION CATEGORIES

Intrusion Detection systems fall into one of three categories that are host-based intrusion detection system (HIDS), network-based intrusion-detection system (NIDS), and hybrids of the two (Bace and Mell, 2002; Hairui and Hua, 2008).

2.5.1 Host-Based Intrusion Detection

Host- based ID analyzes events occurring on a particular computer and identifies activities and users performing malicious activities, on the operating system. It makes use of host operating system and it trails as the main source of input to detect intrusion activity. It also checks key system files and executable files through signature at regular intervals for unexpected changes. Host-based ID was the first area explored in intrusion detection. When the first intrusion detection was designed, the target environment was mainframe computer, and all users were local to the system considered (Bace and Mell, 2002).

As the focus of computing shifting from mainframe environments to distributed networks of workstations, several prototypes of intrusion-detection systems were developed to accommodate network issues. Here the first step was to get host - based IDS to communicate (Bace and Mell, 2002). Host-based ID has a number of advantages and suffers from a number of disadvantages Table 2.2.

Table 2.2: Advantages and disadvantages of host-based ID

Advantages	Disadvantages
Work in encrypted environments	Active target for attacker
Operate on switch-based network	Limited attack visibility
Easy to implement	Incur costs
Effective for insider attacks	

2.5.2 Network-Based Intrusion Detection

With the wide spread use of Internet, intrusion detection systems become focused on attacks to the network itself. Network attacks will not be detected by examining the host audit trail, or at least not easily (Hairui and Hua, 2008).

Network-based ID has been developed to capture and analyze network activities. These detectors can be placed in routers and other network components. This detector works by sniffing or capturing network traffic from various parts of a network and report attacks to centralized management console (Brown et al., 2001). It also has number of advantages and disadvantages Table 2.3. Hybrid approaches have also been developed that use both types in a multi-host environment (Steven et al., 1991).

UMP

Table 2.3: Network-based IDS advantages and disadvantages

Advantages	Disadvantages
Passive target for attackers	Cannot examine encrypted traffic
Greater attack visibility	Low monitoring range in switch-based networks.
Performance cost limited to dedicated host	More complex to implement
	Cannot monitor activity inside , the computer

2.6 INCIDENT RESPONSE OPTIONS FOR IDS

Intrusion detection system generates responses after collecting and analyzing the events and activities. Some of these responses are active, which themselves takes actions on the intrusions. Other responses are passive which report to a proper authority (Amorso, 1999). Active responses provide the IDS with the ability to take action against an attack when it is detected. Figure 2.13: illustrates the active intrusion detection process.

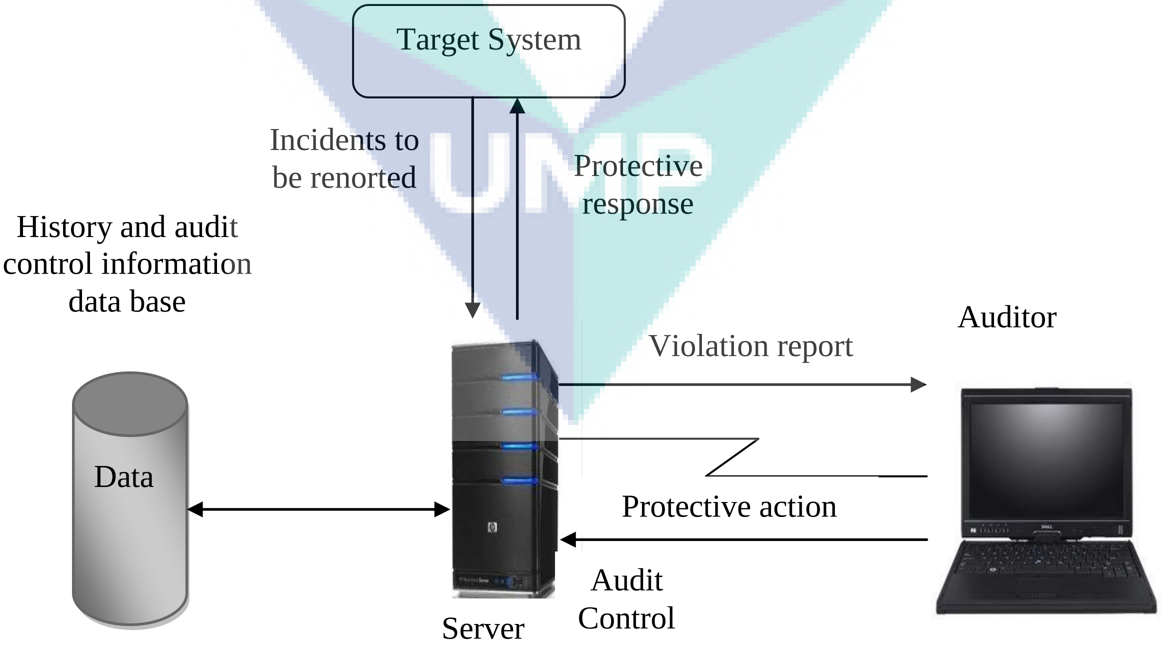


Figure 2.13: Active Intrusion Detection

UMP

These active responses employ two approaches. The first approach deals with gathering more information. These types of responses are constructed assuming that it is safe and useful to gather additional information about the attack. This additional information can help the security officers or the company in taking legal actions against the attackers. The second approach is used to thwart the progressing attack. These systems may take an immediate proactive response. The proactive response can be executed after the violation has occurred, or pre-emptively, to avoid the violation being perpetrated to completion. These include, killing the suspected activity, disabling privileges or user accounts, blocking IP address, etc (Amorso, 1999).

Most intrusion detection systems are passive. It means that when they detect an attack, they generate an alarm, but no countermeasure is actively applied to foil the attack. Passive response systems generally operate offline as shown in Figure 2.14.

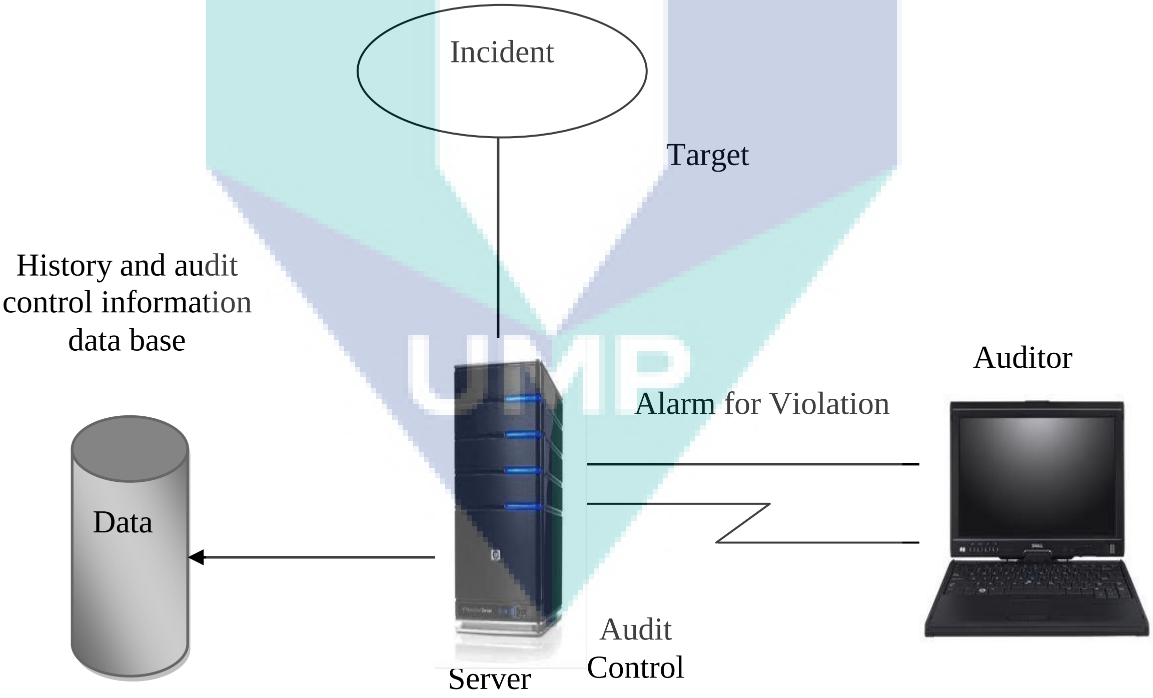


Figure 2.14: Passive Intrusion Detection

It analyses the audit data and brings possible intrusions or violations to the attention of the auditor (Amorso, 1999).

2.7 CLASSIFICATIONS OF INTRUDERS

Intruders can be classified into two types, one who has something to gain by the intrusion and the other a curious person trying to probe the security of the system. The first type is popularly termed as a “cracker”. Crackers attack web-sites or database servers in an attempt to gain critical information such as credit card or social security information. Some try to deface government web-sites or deny normal service and may be backed by political motive.

The second type is the “hacker” who can be further broken down into two types that are an extremely intelligent computer knowledgeable person or a “script kiddie”. An intelligent hacker is one who studies protocols and algorithms and tries to detect vulnerabilities in them. There is nothing malicious about this type although his curiosity and intent is often criticized by many security analysts as irresponsible behavior.

The “script kiddie” is the intruder with limited skills but the one who uses automated computer programs or who exploits code downloaded from the Internet. Needless to say the “script kiddie” is the most common type of intruder. This “script kiddie” is one of the reasons why “security by means of obscurity” will not work.

All these intruders are dangerous to a network system; the “cracker” being potentially the most dangerous and the “script kiddie” the most common (Hu, 2004).

The characteristics of good intrusion detection should address the following issues, regardless of what mechanism it is based on (Turkia, 2002).

i. It must be fault tolerant in the sense that it must survive a system crash and not have its knowledge base rebuilt at restart.
ii. It must run continually without human supervision.
iii. On a similar note to the above, it must resist subversion. The system can monitor itself to ensure that it has not been subverted.

iv. It must impose minimal overhead on the system. A system that slows a computer to a crawl will simply not be used.

v. It must observe deviations from normal behavior.

vi. It must be easily tailored to the system in question.

vii. It must cope with changing system behavior over-time as new applications are being added.

viii. It must be difficult to fool.

2.8 ISSUES WITH NETWORK INTRUSION DETECTION

There are many issues related with the networked intrusion detection.

a) Speed of Data Processing

NIDS have to deal with large amounts of network traffic. To be able to detect intrusions a NIDS must be able to handle large volumes of data at a relatively high rate. NIDS must be able to capture and store network data and also perform analysis on it. Importantly this must be done in real time. If network load increases beyond the point where the system cannot handle it, then intrusions may be undetected or packets might be dropped. NIDS must be able to detect changes in network load and adjust to it. The adjustment that a NIDS could accomplish is to use some kind of filtering mechanism at the raw link level and sort packets based on their importance before analyzing them in more detail (Garuba et al., 2008).

b) Visibility

To ensure a high degree of security for a network a NIDS should have access to all the traffic in the network. Today switched networks are used to increase efficiency by virtually providing two communicating systems with a “point-to-point” i.e. eliminating the broadcast nature of communication. Traditional methods of setting a network interface to listen in promiscuous mode will no longer work in such environments since switches filter traffic based on the interface for which the packet is addressed. NIDS in switched environments have to be configured so that they have access to all the network traffic. Also, any such configurations shouldn’t adversely affect the efficiency of the switches (Endorf et al., 2004).

c) Maintaining States

TCP connections are state based. In order to effectively detect TCP attacks, a NIDS should maintain the different states of these connections. This adds to the memory usage and increases the complexity of the detection process. Evasion is another reason for which the NIDS will have to maintain state. There are many techniques that can be used to evade the scrutiny of an NIDS. TCP fragmentation is one such method where the intruder fragments the malicious packet and fools the NIDS. The other technique commonly used to evade detection is to modify an attack pattern slightly without changing the attack itself. If detection of all possible attacks is of importance to the network then the NIDS must maintain states and should be provided with enough memory. On the other hand, if performance is required then the NIDS may not maintain connection states (Endorf et al., 2004; Qu, 2009).

d) False Positives

The term false positive is a broad and somewhat vague term that describes a situation in which an NIDS device trigger an alarm in a when there is malicious activity or attack occurring .False positive occurs when a NIDS detects an attack when in reality there is none. A NIDS uses signatures (profiles of known attacks) and scan for these signature patterns in sequence of network packets. It is quite possible that the patterns might occur in legitimate packets as well.

Other common terms used to describe this condition are "false alarms" and "benign trigger". False alarm is the better term to describe this behavior since "false positive" gives the impression that IDS technology itself is fundamentally flawed and benign trigger gives the impression that there is no possibility for a true false positive to exist (Endorf et al., 2004; Mell, 2002).

e) *False Negatives*

A false negative can be defined as a case where the NIDS fails to detect the attack. A false negative is a more serious flaw in the NIDS because the administrator will probably never know about. False negatives are dependent on the implementation of the NIDS and how efficient it is in detecting new attacks. Also NIDS rules and attack definitions need to be kept updated on a regular basis (Mell, 2002).

f) *IP Spoofing*

Inherent deficiencies in IP version-4 protocol allow an attacker to easily spoof (fake) IP addresses. With proper knowledge and advanced tools it is possible to impersonate any IP address. IP spoofing affects an NIDS in many ways. Firstly it makes it impossible to trace back the attack to the source since packet routes are not preserved by intermediate routers. Often administrators upon receiving alerts are required to contact the source IP (or the ISP) or lodge a complaint. Secondly a NIDS that drops packets or reject connections based on perceived spoofed IP addresses can result in denial of service.

Under the current protocol version (TCP/IP version 4) it is not possible to completely eliminate IP spoofing. Spoofing can be prevented to a certain extent if network administrators or Internet service providers (ISP's) don't allow a network packet to go out on the Internet that has an IP address that does not belong to their network (Endorf et al., 2004; Duan et al., 2008).

g) *Attacks Against the NIDS*

A NIDS can be subjected to denial of service attacks. If attackers are able to detect it, they will try to flood it with unnecessary traffic causing the NIDS to ignore other traffic. The attacker can then use this situation to direct attack, against an important computer or server. Hiding an NIDS can protect it from attacks. There are many ways to achieve this to varying degrees of success. Using network interfaces without IP addresses and using a receive-only network cable are two such techniques (Vakili et al., 2006).

2.9 PROTECTING AGAINST ATTACKS

There are two different mechanisms for protecting against attacks:

a) Protecting Against External Attacks

Intrusion detection systems for detecting outsider attacks are deployed around organizations network perimeter or behind network firewalls. These systems are not very effective in detecting insider attacks. The problem with most organizations is that employees are given a lot more access than what they actually need to do their job (Hu and Panda, 2010).

Although most companies pay lots of attention to outsider threats and spend significant efforts in securing information systems, very few of them adopt a systematic strategy to mitigate insider threats. Disgruntled employees use legitimate access to the information systems and networks of an organization to commit illegal activities. These activities may include committing fraud, steal company sensitive data, and sabotage employer information systems. Motivating examples that demonstrate the type and nature of possible insider attacks were presented in reports (Yanzhi et al., 2010)

A conventional firewall can do protecting against attacks from the outside. For the sake of simplicity, considering a firewall that does not offer any external services. This means that from the outside it is not possible to reach any mail, web or other services on the inside.

If configured correctly, the firewall only allows a packet from the external network to the internal network if a matching packet was sent in the opposite direction. This means that any communication with the external network is initiated by a host on the internal network. Such a firewall is considered very safe, as the only way to compromise the security policy, with an external attack, is to find a flaw in the firewall, or in the communicating application. Furthermore, it is not enough for the external attacker to find a flaw in the communicating application. The attacker must make the internal application communicate with him in some way. This could be achieved by sending an email to the victim's mail-browser, or convincing the victim to visit a web-site or run an application. Any way the attacker does it; he must somehow make an internal application communicate to the outside, in order to be able to send packets in the opposite direction (Moller and Donbaek, 2001).

b) Protecting Against Internal Attacks

Conventional firewall cannot be protected against inside attack; the adversary might have direct access to an internal host, what is called a hostile user. When a conventional firewall is met by such an attack, it has no way of defending against it. It is simply not able to determine the intentions of the user (Moller and Donbaek, 2001).

Industrial surveys have indicated they have had attacks reported internally. Insider Attacks are an unusual type of threats which are also serious and very common. Unlike an external intruder, in the case of internal attacks, the intruder is someone who has been entrusted with authorized access to the network (Platos et al., 2009). There are two types of internal intruding attacks:

1. **Hostile Applications:** It is very common to users, to download and install applications from the Internet; and that users receive email with an application as an attachment. Those applications can be anything from simple tools to even complete application packages. There is no way for a user to determine what a potentially hostile application will do once it is started. The question here is what might happen if a hostile application is secretly connected to the Internet and disclosed information to the outside? This may be as innocent as checking for an updated version of the application itself, but it could also be sending information to some place on the Internet without the user's knowledge. In the worst case, it might be a backdoor to the trusted network, circumventing the firewall.

2. **Hostile Users:** In the case above, an innocent user is a victim to a hostile application. In another scenario, the user might be a hostile himself. This may be happened due to an institution's policy is too strict. The user might, for instance, want to be able to connect to his machine from the Internet when he is on a business trip, or he might even want someone on the Internet to gain access to his files or other services, without letting the firewall administrator knows, or even an unauthorized adversary user who want to disclose an important information into the outside. Such a malicious user can, with a little knowledge, download and install a backdoor on a machine inside the network and enable anyone to gain access to this machine, or the adversary user can use an authorized application to disclose information to the outside of the network.

In both cases, an application is running on a host inside the firewall that connects to the outside network and the application is circumventing the security policy of the organization. A common firewall knows nothing about which applications are creating connections through it. It only knows the source and destination addresses of the numerous network packets. Besides the source and destination ports, the firewall cannot really rely on the information anyway. The conventional firewall simply has no chance of determining whether a user is running a hostile application, or if the user is hostile himself, or if it is a legal connection.

Insider attacks are called high level insider threats, which will make great damage to system. The classification of insider threats helps us to dispose different insider threats in different ways, which can accelerate the process of them. A lot of research have been conducted on anomaly detection against insider attacks (Zhang et al., 2010; Moses et al., 2008), where the objective is to detect deviation from a predetermined model of normal system behavior. However, since an insider has authorized access and extensive knowledge of the victim system, it can be more difficult to separate normal system behavior from insider attacks than the external ones.

2.10 SUMMARY

This chapter introduces a brief introduction to network models, intrusion detection model, discussing the types of intrusion, classifications of intruders, issues with network intrusion detection and principles of operation of two mechanisms for protecting against attacks that are protecting against outside attacks and protecting against inside attacks. The subsequent chapters will be built upon these concepts in.

UMP

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter explains in detail the algorithm that was implemented on a client/server model for a LAN attempts to improve the security process in a communication network that employs TCP/IP. It focuses on defending against the internal attacks. First, client server model was identified and application scenario for each of the clients and server was suggested in order to authenticate and verify the extra security information to enhance the network security for LAN model computer networks. Finally, clients and server cooperate in such a way that each of the clients does authenticate each of its application, host and user, to the server. The server will verify these information in turn.

3.2 CLIENT-SERVER MODEL

The term client/server was first used in the 1980 in reference to Personal Computers (PCs) on a network. The actual client/server model started gaining an acceptance in the late 1980s (Zaw and Su, 2008). The client/server model has become one of the central ideas of networking. The Client-Server computing model is a popular concept and many Internet and database applications are based on this model (Varekova et al., 2010). Most networking applications are being written today using the client/server model. The same can be said about internet (Gordon Bell and Jim Gray, 2002).

Client/server is a network application architecture environment where the control of data is established at a server node and is available for access from the clients. Client/server describes the relationship between two computer programs in which one program, the client, which makes a service requested from another program and the server, which fulfills the request (Carr, 1998; Han et al., 2010).

These days, most computers are multi-tasking i.e can run many programs simultaneously. Therefore a single computer can run the server and client programs at the same time (Bagwill, 1994). Often, in such a case this computer is a server of the internal network and is a client of the outer network.

UMP

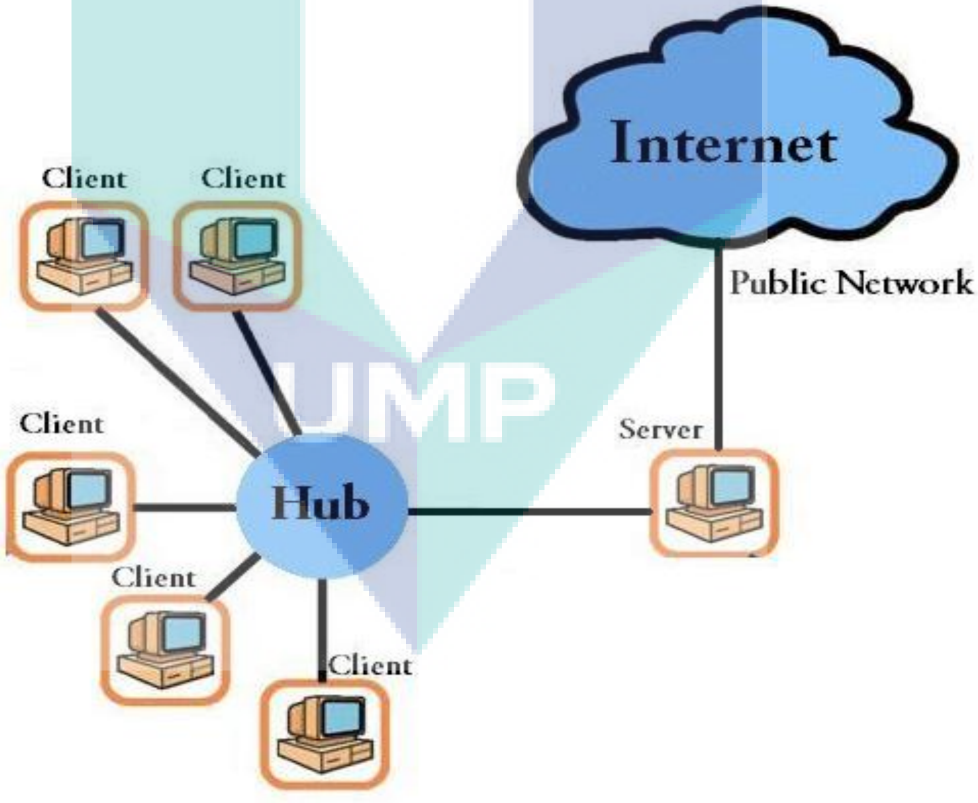


Figure 3.1: client/server model in a basic LAN topology

This work proposes a (LAN) topology based on a client/server model in which the server of the LAN is a client of a public network (Internet) and each LAN's client can connect to the public network through only the LAN's server. Figure 3.1 illustrates the relationship between the clients and server in a basic LAN topology connecting to an Internet.

3.3 AUTHENTICATION AND VERIFICATION MODEL

The information used for authorizing the application, host, and user is supplying to each outgoing packet to the extended firewall, in order to make a decision about whether to forward a packet or not. This is due to the fact that providing the extra information just in the first packet of a communication link is insufficient. Since, the adversary is capable of hijacking an established communication link, due to physical access to the trusted network (Moller and Donbaek, 2001).

Also, providing each packet with just a userID, applicationID, and hostID is a very naive approach. The adversary is capable of sniffing the trusted network and of sending/modifying packets, so duplicating the credentials of a valid packet is easy for him to do. The extended information i.e. the applicationID, hostID and userID are added to each outgoing packet produced from a client. Figure 3.2 shows communication scheme between client program and server program.

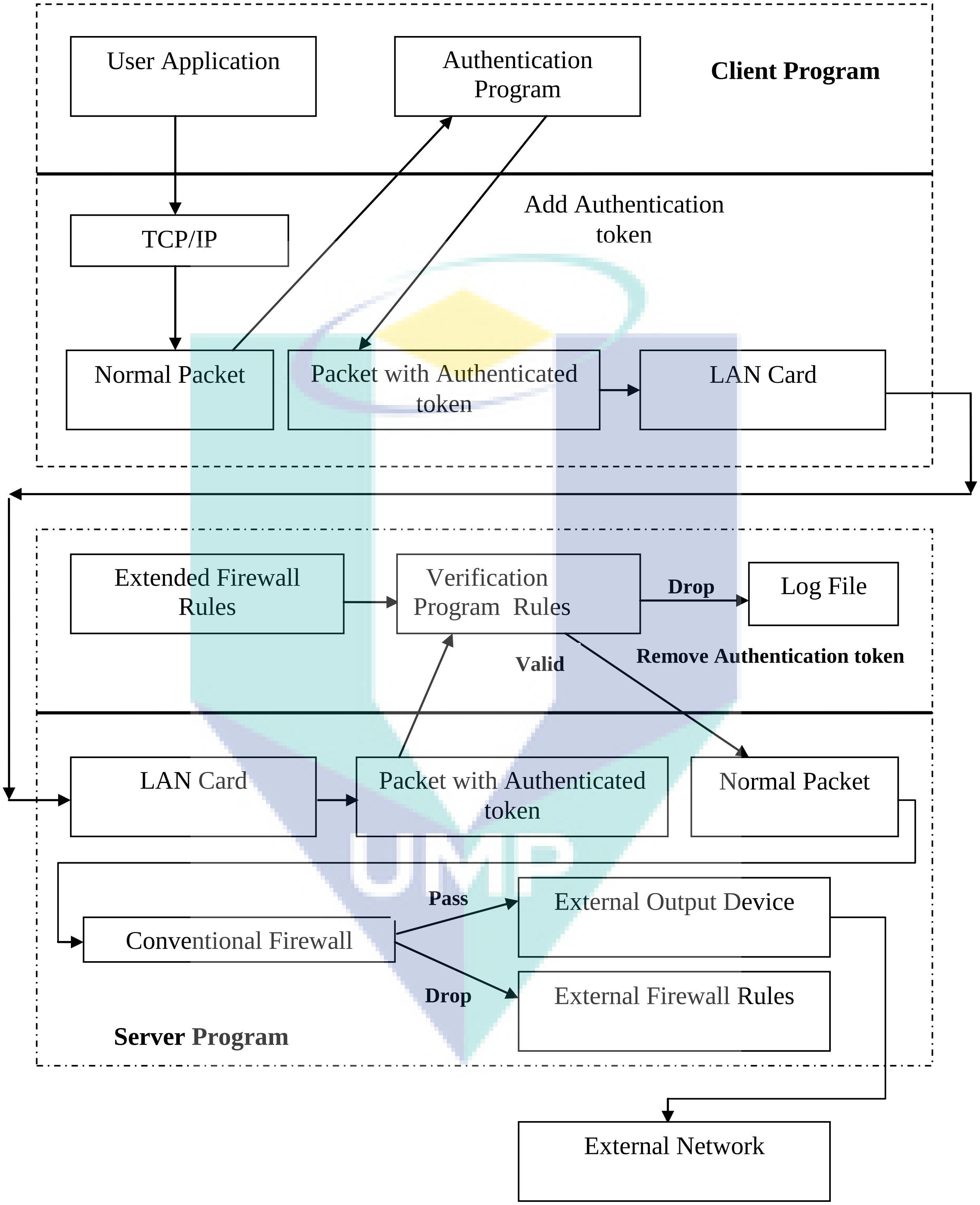


Figure 3.2: Communication scheme between client program and server program

3.3.1 Application Name Authentication(Application ID)

As stated in the problem statements, there are two types of inside attacks: Hostile users and Hostile applications.

Hostile user is the user that put an unauthorized application into the host in order to disclose some information to outside of the network or to overcome the restriction of the administrator security policy. However, hostile application is the application that discloses information to outside of the network without the user notice. The user is the victim. Hostile applications can be controlled by the personal firewalls as they display the application name that sent packets at that host and gives a control to the user to stop this application.

Unfortunately, this supposes that every user must have a good knowledge about the application authority and the security policy of the network administrator which is a rare case. This is a difficult condition especially that the network has tens or even hundreds of hosts and users. In both cases (hostile user and hostile application), an unauthorized application is running on a host inside the firewall that connects to the outside network. The conventional firewall at the server knows nothing about which applications are creating connections through it. Every packet that is destined to leave the network has to contain the application's name, which sent this packet. When this packet reached to the server, the extended firewall determines the action (Drop/Pass) to this packet. To implement this, the proposed application at every client has to:

i. Get every packet going out from this client before sending.
ii. Get the application name, which sent this packet.
iii. Add this application name to the packet.
iv. Transmit the packet.

3.3.2 Host Identifier(Host ID)

The second important problem in protecting the LAN network against the internal adversary is the ability to attach an extra host to the LAN network. In such a case, the adversary can install software and eavesdrop on the network. This new host would be a perfect place for the internal adversary to send information outside the network. In addition, it is possible to install software that enables the adversary to connect to this host from the outside. In both cases, the installed software may have been authorized.

To solve this problem, firewall has to know the source of the packet in order to distinguish between trusted and untrusted hosts. This enables the firewall to let only the packets that came from the trusted hosts to pass the network. The conventional firewall can distinguish the source of the packets by reading both fields, i.e. source Media Access Control (MAC) address and source IP address. This is not always valid due to two reasons. First, Dynamic Host Configuration Protocol (DHCP) is a protocol, which enables a host to obtain an IP address dynamically from a DHCP-server upon boot time. This means that the host can have a different IP address whenever it is rebooted. Second, the internal adversary can change the values of these two fields of his packets into the corresponding fields of an authorized host.

To solve this problem, management of an enumeration for all protected hosts behind the firewall and initialization of each host with its unique host Identifier (ID) are needed. Every packet that destiny to leave the network has to contain the ID that send this packet. When this packet reached to the server, the extended firewall has to retrieve the Host ID to determine the action (Drop/Pass) of this packet. In (Moller and Donbaek, 2001), the author used a normal string as the Host ID which read from a configuration file. However, this method has a few disadvantages. First, the string can be repeated by the user or the adversary (whether he/she meant or not) on other hosts. Second, the string can be stolen or even the whole configuration file by the adversary.

An alternative solution to these problems were proposed. First, the Host ID can be a serial number of one of the hardware components e.g. manufacturer hard disk serial number and CPU identification number (chosen in this case), or manufacturer Compact Disk (CD) serial number and so on, which are always unique. Second, the proposed application itself read the Host ID at the beginning of its execution and not from file, to prevent the adversary from copying or stealing the Host ID to use it in his application. Even if the adversary has stolen the hooking application, it will be useless.

3.3.3 User Identification (User ID)

Another important problem in protecting the internal network against the internal adversary is the ability of unauthorized (adversary) user to send information to the outside. To solve this problem, the firewall has to identify the person who sent this packet in order to distinguish between trusted and untrusted person as shown in Figure 3.3. This enables the firewall to let only the packets that sent from a trusted person to leave the network.

A user Identification (User ID) is unique information that added to every packet in order to determine the sender of a packet. Choosing the appropriate User ID is depending on the administrator security policy. In such a case, there are two possible situations:

i. If the administrator security policy decided that every authorized user worked on only his authorized host (not else) then the User ID can be the user password (read from a text box from the application) XOR with the Host ID.

ii. If the administrator security policy decided that any authorized user can work on any authorized host then a user name and a user password (read from a text box from the application) can both considered as a User ID.

One advantage of using this method is that no cracking can be done to the User ID because the User ID is not stored in the client at all, and the matching will be done at the server. The server will detect any false User ID immediately and record the error in a log file.

3.4 MESSAGE AUTHENTICATION

Message authentication is a procedure to verify that a received message comes from the alleged source and have not been altered (Bishop, 2002; Seberry and Pieprzyk, 1989). Authentication procedure required the following points:

i. *Data integrity:* If the message has been modified in transmission from the protected host to the extended firewall, this must be detected.

ii. *Identification of source:* An untrusted party must not be able to fake an authenticated message, even when it has unlimited access to the network traffic between the protected hosts and the firewall. This is also known as data origin authentication.

iii. *Speed*: Each LAN's host has to calculate authenticated messages especially the server. The calculation and verification, which involve hundreds or thousands of authenticated messages each second, need to be executed very fast.

Message authentication can be classified into three classes (Stallings, 1999; Stallings, 2000):

i. Message encryption: the cipher text of the entire message serves as its authenticator.

ii. Message Authentication Code (MAC): a public function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

iii. Hash function: a public function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

As stated earlier, unlike Hash function, MAC has a secret key that used to produce a fixed-length value that appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K. When A has a message for sending to B, it calculates the MAC as a function of the message and the key as show in Eq. (3.1).

$$MAC=C_k\ (M) \qquad (3.1)$$

The message plus MAC are transmitted to the intended recipient.

The recipient performs the same calculation on the received message, using the same secret key, to generate a new MAC. The received MAC is compared to the calculated MAC. If the receiver and the sender know the identity of the secret key, and if the received MAC matches the calculated MAC, then (Stallings, 1999; Stallings, 2000):

i. The receiver is assured that the message has not been altered. If an attacker alters the message but does not alter the MAC, then the receiver calculation of the MAC will differ from the received MAC. Since the attacker is assumed does not know the secret key, the attacker cannot alter the MAC to correspond to the alterations in the message.
ii. The receiver is assured that the message is from the alleged sender. Because no one else knows the secret key, no one else could prepare a message with a proper MAC.

The IP-datagram and the identification token pair can be considered as a message. So, in other terms, this means that if the protected host shares a secret key with the extended firewall, it possible, for every message sending to the firewall, the MAC value for the message will be calculated and sent along with the message to the firewall. If the extended firewall is able to reproduce the MAC value with the same· key on the received message, it identifies not only that the message originates from a source which has knowledge of the key, but also that the message has not been altered in transit (with negligible probability of failure). This includes the data integrity and identification of source requirements for the authentication scheme.

UMP

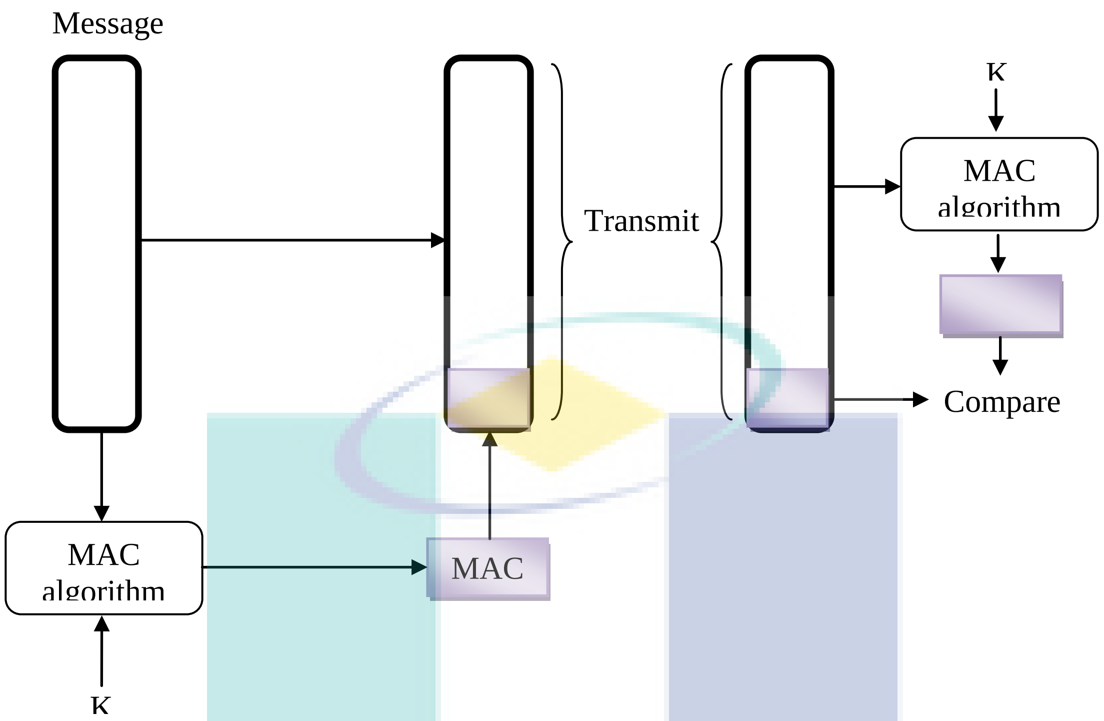


Figure 3.3: The use of MAC method

In order to increase the difficulty of finding the MAC key, this thesis proposes that the identification field and the next six bytes of the IP header is appended to the whole message before the MAC calculation. However, it will be removed from the sending message. This ensures that all messages MAC with the current key are unique. Moreover, the MAC key will be the chosen key and XOR with first 16 bytes of the IP header which change at each packet to increase the security. This is due to the chosen method (UMAC) key is 16 bytes length. For each message received, the extended firewall will append the identification field of the IP header at the end of the message and then recalculate the MAC value.

3.5 SUMMARY

In this chapter, a brief introduction to client-server model, identified of client-server model, authentication and verification model for User ID, Host ID and Application ID is described.

UMP

CHAPTER 4

SOFTWARE DESIGN

4.1 INTRODUCTION

This chapter explains in detail the software design algorithm that was implemented on a client/server model for a LAN attempts to improve the security process in a communication network that employs TCP/IP. Finally, clients and server cooperate in such a way that each of the clients does authenticate each of its application, host and user, to the server. The server will verify these information in turn.

The proposed algorithm was implemented on a LAN network topology consisting of ten computers, one server and six authorized clients in addition to three unauthorized clients. The numbers of applications that are used are seven authorized applications and four unauthorized applications. In order to test the proposed implementation, the execution of the client portion began with reading the Host ID and appended to each packet going to the server.

4.2 AUTHORIZATION/DETECTION ALGORITHM

The check sum is calculated at the sender and the value obtained is sent with the packet. The receiver repeats the same calculation on the whole packet including the check sum. If the result is satisfactory (match) then the packet is accepted; otherwise, it is rejected. Since, the lengths of the IP and UDP/TCP packets are increased by adding the extra information, and since the contents of these packets are changed, a recalculation of the check sum for both the IP and the UDP/TCP packets are necessary.

The Clients procedures algorithm is as follow:

Step-1: Specify the network card
Step-2: Load the driver
Step-3: read the user name and password
Step-4: Perform the Host ID function to get the Host ID
Step-5: Wait until outgoing packet is reached to the driver
Step-6: Transfer the packet from the kernel mode to the user mode
Step-7: Add Application ID, Host ID, user name and password to the packet
Step-8: Perform authentication process to the packet payload using UMAC method
Step-9: Perform ciphering process to the packet payload using DES method
Step-10: Recalculate checksum of the packet
Step-11: Transfer the packet to the kernel mode
Step-12: Transmit the packet
Step-13: go to step -5.

In order to increase the speed of verifying the incoming extended information, IDs for the authorized information are presented. This proposed algorithm has two types of ID at the server (Host ID, User ID and Application ID) as shown in Figure 4.1**.**

First is the Host ID, User ID which is composed of three fields:

i. Host ID: which contains the Host ID of the client that is connected to the server at an instant of time

ii. User ID: which contains the User ID that uses the connected client (Host ID) at an instant of time

iii. A pointer to the corresponding Application ID.

Second, is the Application ID. There are N applications ID in the system, where N is the number of the Host ID User ID entries. Each Application ID contains the Application ID of each application that are executed on the connected client (Host ID) by the User ID at the Host ID User ID entry. These structures are considered, because there is only one user on one host at a time, however many applications may run on the client (Host) by the user.

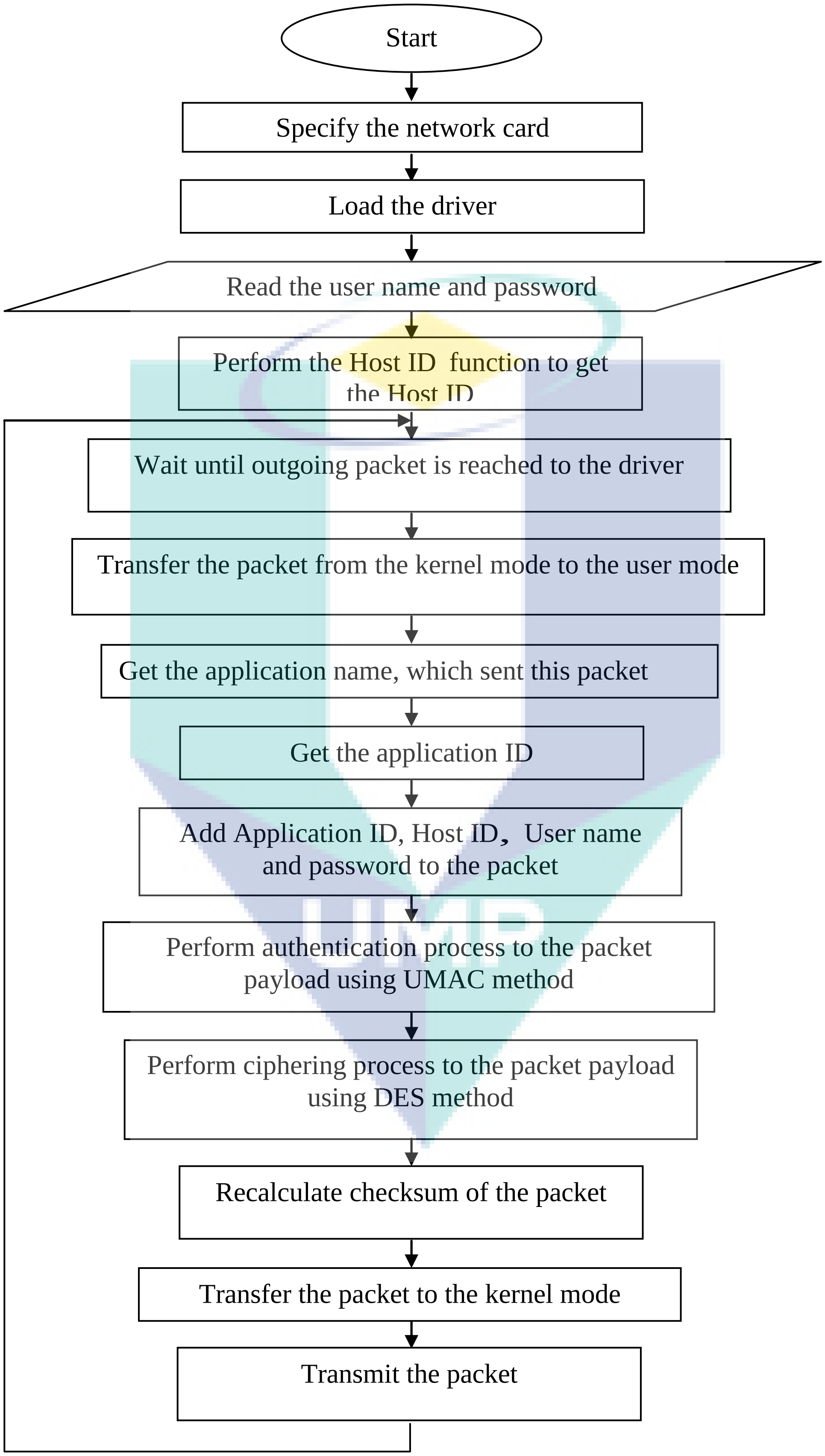

Start
Specify the network card
Load the driver
Read the user name and password
Perform the Host ID function to get the Host ID
Wait until outgoing packet is reached to the driver
Transfer the packet from the kernel mode to the user mode
Get the application name, which sent this packet
Get the application ID
Add Application ID, Host ID，User name and password to the packet
Perform authentication process to the packet payload using UMAC method
Perform ciphering process to the packet payload using DES method
Recalculate checksum of the packet
Transfer the packet to the kernel mode
Transmit the packet

Figure 4.1: Client's main procedures

4.3 VERIFICATION PROCESS

When the extended firewall (verifier) receives an extended IP datagram from a protected host, it has to check whether it should accept or deny. So, the processing for each incoming message begins with loading the driver for the specified adapter, and then transfers the incoming packet from kernel mode to user mode. This operation is also important for retransmitting the authorized packet. When the packet is in user mode, the verifying process will be easy.

The next step is deciphering the message. After that, the authenticity of the message must be checked. This is done by verifying the MAC value received from the client with the message appended to it. If this value is correct, the extended firewall will verify the authenticity for the User ID, Application ID and Host ID.

The verification process of the hosted is accomplished by searching in a data base file which contains the authorized information. If MAC value was incorrect or the authenticity for the User ID, Application ID and Host ID was incorrect, this packet will be dropped and logging in a log file. If the packet was authorized, the original packet must be extracted from the received one by removing the extended information and recalculating the checksum. The last step is resending the packet.

The problem with the server is that it may receive a large number of packets in a short time. According to this fact, the server has to process and verify the incoming packets as fast as possible. The server deciphers each packet by using a symmetric ciphering method, which is very fast comparing to asymmetric ciphering methods. In addition, the server uses the fastest method of authentication process.

The matching process for each of the Host ID, User ID, and Application ID, which will be searched in each database file, is very slow. In order to increase the speed of this process, complete information in the database is essential. Eq. (4.1) shows the authentication and verification process.

$$\sum_{i=1}^{n} f(x_i) = \begin{cases} \sum_{i=1}^{m} u_i, \sum_{i=1}^{m} p_i, \forall\, u_i, p_i & pass \wedge \\ \sum_{j=1}^{n} id_j \;, \forall\, id_j & pass \;\; \wedge \\ \sum_{i=1}^{n} \;\; \sum_{k=1}^{l} t_{ik}, \;\forall\, t_{ijk} & pass \end{cases} \qquad (4.1)$$

Where:

n is a number of available users in the server.

m is a number of available and unavailable users.

l is a number of applications active with each users. m ≥ n

The server main procedures algorithm is as follow:

Step-1: Specify the network card

Step-2: Load the driver

Step-3: Read the user name and password

Step-4: Open authorized user ID file, authorized Host ID and authorized application ID

Step-5: Wait until incoming packet is reached to the specified driver

Step-6: Transfer the packet from kernel mode to user mode

Step-7: Deciphering the packet

Step-8: Recalculate the packet authentication value using the key

Step-9: If Match with the sent one go to step -15-

Step-10: If the User ID, Application ID and Host ID match with authorized,
go to step-15.

Step-11- Rebuild the IP packet without the extended information

Step-12- Recalculate checksum

Step-13- Send the packet

Step-14- Go to step -5-

Step-15- Send into a log file

Step-16- Drop the packet

Step-17- Go to step -5-

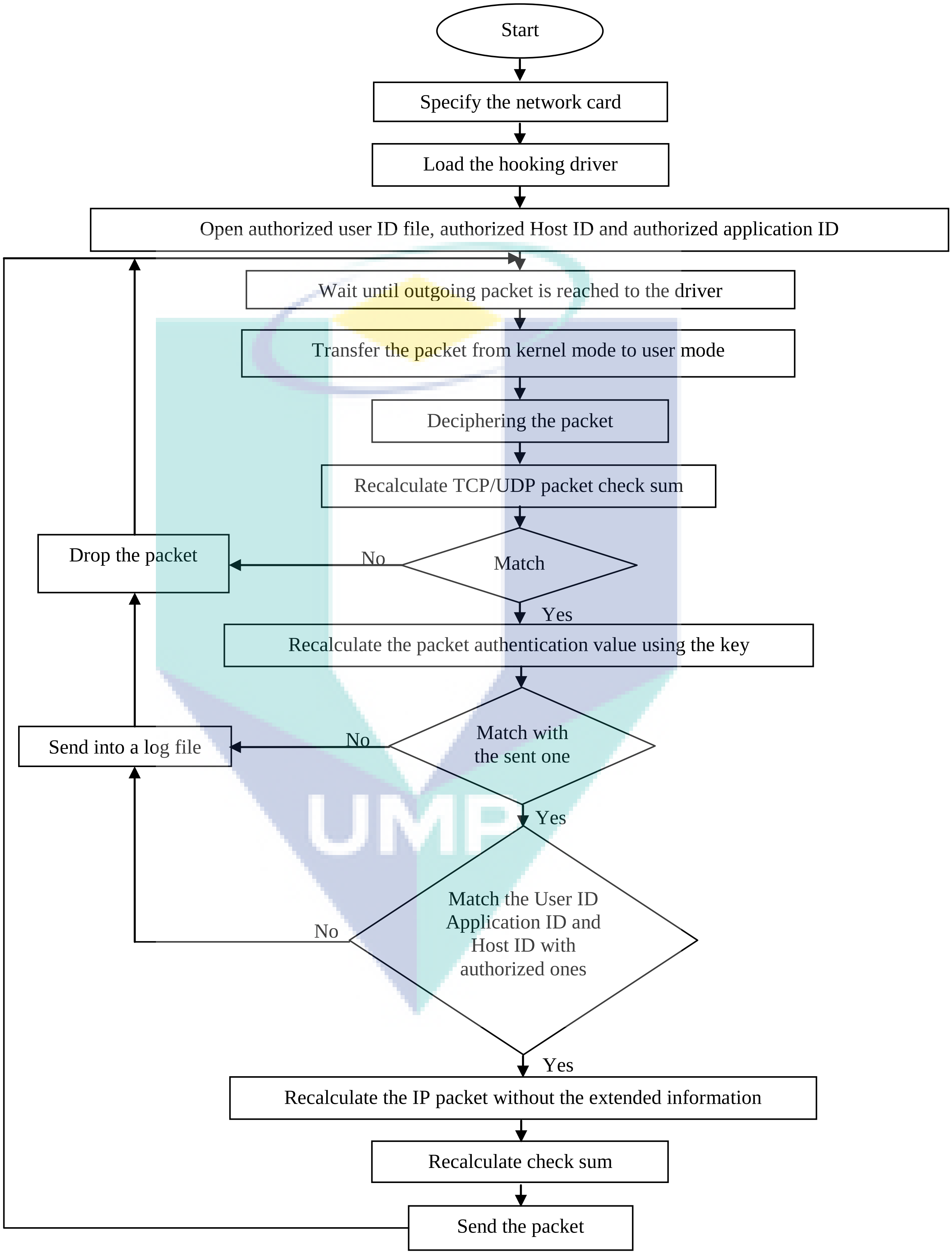

Start
Specify the network card
Load the hooking driver
Open authorized user ID file, authorized Host ID and authorized application ID
Wait until outgoing packet is reached to the driver
Transfer the packet from kernel mode to user mode
Deciphering the packet
Recalculate TCP/UDP packet check sum
Match
No
Yes
Drop the packet
Recalculate the packet authentication value using the key
Match with the sent one
No
Yes
Send into a log file
Match the User ID Application ID and Host ID with authorized ones
No
Yes
Recalculate the IP packet without the extended information
Recalculate check sum
Send the packet

Figure 4.2: Server main procedures

When the incoming packet is deciphered and authorized by the UMAC method, the Host ID is read from the packet and searched in the Host ID, User ID. There are two situations. First, if it is not found, the authorized Host ID file will be searched. If it is not found in the file, then this Host ID is unauthorized and this packet will be dropped and logged. Otherwise, a new Host ID, User ID entry is added. When a new entry is added, the following operations must be executed in order to fill the entry fields:

i. Add the packet Host ID into the Host ID flied in the Host ID, User ID.

ii. Read the User ID from the packet and search it in the authorized User ID file. If it is not found, then it is unauthorized, else it will be added to the Host ID, User ID.

iii. Allocate a new Application ID for this entry and makes the pointer field points to the beginning of this Application ID.

iv. Read the Application ID from the packet, search it in the authorized Application ID from the packet, and in the authorized Application file. If it is not found, then it is unauthorized, else it will be added to Application ID.

In the second situation, the packet's Host ID is found in the Host ID, User IDs. In this case, the packet User ID is read and matched with User ID field in the same Host ID, User ID entry. Here, another two situations raised. First, if the User ID match, the packet Application ID will be read and searched in the corresponding Application ID. If it is found, then this packet is authorized. On the other hand, the packet's Application lD in the authorized Application ID file will be searched. If it is not found, then it is unauthorized. In contrast, this Application ID will be appended to the Application ID and the packet will be authorized.

In the second situation, the packet's User ID is not found in the User ID field (of the same entry of that found the packet's Host ID) in the Host ID, User ID entry. This means that a new user is using an old connected client (host) instead of the old user. In such a case, the old corresponding Application ID entries are removed first and the Application ID will be checked as described above.

4.4 SUMMARY

In this chapter, a brief introduction to software design model is described including identification of client-server model, and verification model for User ID, Host ID and Application ID. The clients procedure algorithm is perform authentication process before transmit the packet and the server main procedures algorithm verify that User ID, Host ID and Application ID are authorized.

CHAPTER 5

RESULT AND DISCUSSION

5.1 INTRUSION DETECTION ANALYSIS

The prototype application was run in a LAN environment which consists of 10 computers. From the total number of computer, one computer is a server, six computers operate as authorized clients and the other three computers operate as unauthorized clients. Seven authorized applications and four authorized applications were tested.

Table 5.1: illustrates the client computer names with their Host ID which are retrieved by using the proposed solution at the client, in addition to the authorization state (depending on a database at the server) and the server's action for this packet.

Table 5.1 shows the second stage of checking for client where the server checked the Host ID-HD and Host ID-CPU by searching in its authorized database for each computer. In this example, Computer 1, Computer 2, Computer 3, Computer 4, Computer 5 and Computer 6 had succeeded as well passed this step. On the other hand, Computer Hack1, Computer Hack2 and Computer Hack3 had strange Host ID's which means unauthorized. This is because the server recognized that the HD or CPU, or both are not permitted. As a result, this packet was dropped.

Table 5.1: Clients' computer names, Host ID, their authorization state and server reaction

Computer Name	Computer Host ID-HD	Computer Host ID-CPU	Authorization Stat	Server's Reaction
Computer 1	WD-WMA8E8787590	BFW 9FBFF000006V7	Authorized	Pass
Computer 2	WD-WMA8E8787882	BFW 9FBFF00000558	Authorized	Pass
Computer 3	WD-WMA8E8787872	BFW 9FBFF00000769	Authorized	Pass
Computer 4	WD-WMA8E8787876	BFW 9FBFF000004X7	Authorized	Pass
Computer 5	WD-WMA8E8787862	BFW 9FBFF00000999	Authorized	Pass
Computer 6	WD-WMA8E8787931	BFW 9FBFF000003Z4	Authorized	Pass
Computer Hack1	WD-WMA8E8787882	BFW 9FBFF00000905	Unauthorized	Drop
Computer Hack2	WD-WMA8E8787992	BFW 9FBFF000004X7	Unauthorized	Drop
Computer Hack3	WD-WMA8E8787999	BFW 9FBFF00000777	Unauthorized	Drop

When testing against ApplicationID, the following applications were chosen for the test. Table 5.2 shows that each packet coming from authorized application (has authorized ApplicaitonID) (Adobe Acrobat 9 Pro, Download Manager, Windows Live Photo Gallery, MS Word, MS Powerpoint, MS Excel and MS Access) had succeeded.

In contrast, applications (Internet Explorer, Calculator, WM Player and MS Paint) are unauthorized because the server had detected that the applications are not permitted and as a result, this packet was dropped.

Table 5.2: Clients' applications' names, their authorization state and server reaction

Application's Names	**Authorization Stat**	**Server's Reaction**
Adobe Acrobat 9 Pro	Authorized	Pass
Download Manager	Authorized	Pass
Windows Live Photo Gallery	Authorized	Pass
MS Word	Authorized	Pass
MS Powerpoint	Authorized	Pass
MS Excel	Authorized	Pass
MS Access	Authorized	Pass
Internet Explorer	Unauthorized	Drop
Calculator	Unauthorized	Drop
WM Player	Unauthorized	Drop
MS Paint	Unauthorized	Drop

UMP

The third parameter tested was the User ID. Two situations were considered and tested. In the first situation, the authorized user can access the network from any authorized host when both the user name and password are correct.

This operation is made by the client, where the server checked the User ID by searching in its authorized database. Table 5.3 shows this situation and the corresponding server's reaction.

Table 5.3: Clients user names, user password, their authorization state, and server reaction

User Name	User Password	Authorization Stat	Server's Reaction
ADRIAN	Ad10	Authorized	Pass
CARROLL	Honey	Authorized	Pass
CONNOR	Queen	Authorized	Pass
MARYAM	Ma01	Authorized	Pass
ALEXANDER	Ale6	Authorized	Pass
MUSTAFA	Must5	Authorized	Pass
ALEXANDER	Axx09	Unauthorized	Drop
NOOR	N1n1	Unauthorized	Drop
CARROLL	Om99	Unauthorized	Drop

In the second situation, the authorized user can only access the outer network from his/her own client. In such a case, the user name was XOR with user password was XOR again with the HostID. Table 5.4 illustrates the different tested states for this situation. The results show that even if both the user and the client were authorized but the client was not the intended for that user, the packet was dropped.

Table 5.4: Client user names, user password, Host ID, their authorization state, and servers reaction

User Name	**User Password**	**Host ID-Hard Disk**	**Host ID-CPU**	**Authorization Stat**	**Server's Reaction**
ADRIAN	Ad10	WD-WMA8E8787590	BFW 9FBFF000006V7	Authorized	Pass
CARROLL	Honey	WD-WMA8E8787557	BFW 9FBFF00000558	Authorized	Pass
CONNOR	Queen	WD-WMA8E8787862	BFW 9FBFF00000769	Authorized	Pass
MARYAM	Ma01	WD-WMA8E8787931	BFW 9FBFF000004X7	Authorized	Pass
ALEXANDER	Ale6	WD-WMA8E8787662	BFW 9FBFF00000999	Authorized	Pass
MUSTAFA	Must5	WD-WMA8E87879981	BFW 9FBFF000003Z4	Authorized	Pass
ADRIAN	Ad10	WD-WMA8E8787017	BFW 9FBFF000006V7	Unauthorized	Drop
CARROLL	Honey	WD-WMA8E8787557	BFW 9FBFF000001C4	Unauthorized	Drop
CONNOR	Queen	WD-WMA8E8787352	BFW 9FBFF00000777	Unauthorized	Drop

The proposed implementation also overcomes the security aspects such as interception, modification and fabrication by following ciphering and authentication mechanisms. For ciphering, the packet overcomes the interception whereas authentication overcomes both the modification and fabrication aspects.

UMP

The fourth tested parameter was the authentication value. There are sixteen different states can be raised in server portion. All the sixteen different tested states and the corresponding server reaction are as shown in Table 5.5.

Table 5.5: All tested states and the corresponding servers reaction

User ID Authorization	**Host IDs Authorization**	**Application ID Authorization**	**Authentication Correction**	**Server's Reaction**
Authorized	Authorized	Authorized	Correct	Pass
Authorized	Authorized	Authorized	Incorrect	Drop
Authorized	Authorized	Unauthorized	Correct	Drop
Authorized	Authorized	Unauthorized	Incorrect	Drop
Authorized	Unauthorized	Authorized	Correct	Drop
Authorized	Unauthorized	Authorized	Incorrect	Drop
Authorized	Unauthorized	Unauthorized	Correct	Drop
Authorized	Unauthorized	Unauthorized	Incorrect	Drop
Unauthorized	Authorized	Authorized	Correct	Drop
Unauthorized	Authorized	Authorized	Incorrect	Drop
Unauthorized	Authorized	Unauthorized	Correct	Drop
Unauthorized	Authorized	Unauthorized	Incorrect	Drop
Unauthorized	Unauthorized	Authorized	Correct	Drop
Unauthorized	Unauthorized	Authorized	Incorrect	Drop
Unauthorized	Unauthorized	Unauthorized	Correct	Drop
Unauthorized	Unauthorized	Unauthorized	Incorrect	Drop

UMP

Any violation to the token authorization or incorrect authentication from any client will be detected and reported in a log file as shown in Table 5.6:

Table 5.6: Error log file

Error type	IP address	Time	Action
Unauthorized User ID	172.25.181.137	22:10:37	Dropped
Unauthorized Host ID	172.25.181.163	22:16:53	Dropped
Incorrect UMAC value	172.25.181.178	22:32:18	Dropped

5.2 PERFORMANCE EVALUATION

In order to use the proposed solution practically, it must be efficient at handling a large number of packets accurately. In other words, it is intended to emphasis the proposed solution to show that the users can gain a considerable increase in security by using the design, with only an insignificant performance penalty on the network bandwidth.

Testing the performance of the implementation is the concern. In particular, evaluating whether there is a considerable impact on performance either the bandwidth or on the secured hosts behind the extended firewall, compared to a similar configuration using only the normal Windows firewall.

The tested LAN network was connected to the Internet through a 384 kbps connection; this is the maximum attainable transfer rate to this network. The LAN network hosts were connected together by a 100Mbps hub, so the bottleneck was definitely the 384 kb connection. The test was focused on a mix of uploading and downloading transfers.

When downloading data through a TCP connection, one acknowledgement packet was sent for each received data packet. However, the data packet was much larger than the acknowledgement packet. Often the data packet will be 1514 bytes as a maximum, while the acknowledgement packet is only 40 bytes.

Since the performance impact of the extended firewall is only on outgoing packets, thus, in a download test, the protected host will only have to authenticate the smaller acknowledgement packets, while the opposite is valid for an upload test. Therefore, the upload and download data transfers were tested.

i. The first test was a download test; i.e. the bulk of the transfer was incoming packet. This is what happens most of the time when people browse on the web, download files using either HTTP or FTP transfers, and also when they read email. However, there was a performance impact here too, since all the above mentioned transfers are based on TCP. This means that the TCP protocol acknowledges all incoming packets, and that all these acknowledgements must be authenticated to the firewall.

ii. The second test was an upload test, where the bulk of the transfer as outgoing packet. This is what happens when people upload a file to an FTP server or send email. In this case, the secure host and the firewall were authenticating or verifying almost all the traffic; therefore the largest performance impact was expected on this test.

Since there is much larger bandwidth on the LAN network than to Internet, the performance of Internet communication should only be minimally affected if a security system adds only to the LAN bandwidth.

The performance tests results for download are as shown in Table 5.7 and Table 5.8 shows the results for upload. The "ordinary" columns are the test results made with the conventional Windows kernel firewall. The "extended" columns are the test results made with the proposed extended firewall.

Each of the downloading and uploading testing was repeated ten times and all the tested results were measured in seconds. Based on the results, an average time for each host was calculated. Since the transfer for each test involved a fixed size (2.6 Mb), an average transfer rate in Kb/second for each host with and without the extended firewall was calculated. Lastly, a transfer index for the "extended" columns, based on a transfer index of 100 for the "ordinary" columns was calculated.

The proposed extended firewall was evaluated based on these two equations:

Transfer rate = File size/average time (Kb/Sec) (5.1)

Transfer index (Extended download) =
transfer rate (Extended download)×100 / transfer rate (Ordinary download) (5.2)

Table 5.7 shows that the averages of 10 computers downloading 2.6 MB took 337.7 seconds using the conventional firewall whereas it took 344.5 seconds with the extended firewall. For the transfer rate, the average transfer rate using the conventional firewall was 7.69 Kb/second and this was calculated based on Eq (5.2). However, using the extended firewall, the transfer rate was 7.54 Kb/second. The transfer index shows percentage of transfer rate for the extended firewall compared to that of the conventional firewall using Eq. (5.2). The transfer index achieved 98.04% for the conventional firewall, which is within the expectations.

Table 5.7: Download performance in ordinary and extended way

Computer No.	Ordinary Download (Time in Sec.)	Extended Download (Time in Sec.)
1	330	341
2	335	339
3	342	345
4	334	346
5	348	354
6	335	352
7	340	349
8	334	337
9	337	338
10	342	344
Average	337.7	344.5
Transfer rate (Kb/Second)	7.69	7.54
Transfer index	100	98.04

Table 5.8 shows the averages of 10 computers uploading 2.6 Mb using the conventional firewall and it took 338.4 seconds on average for uploading time. On the other hand, the uploading time was 346.6 seconds using the extended firewall. Similarly, the transfer rate was calculated using Eq. (5.1) which is 7.68 Kb/second for the ordinary case and 7.50 Kb/second using the extended firewall. The transfer index using was 97.65%.

Table 5.8: Upload performance in ordinary and extended way

Computer No.	Ordinary Upload (Time in Sec.)	Extended Upload (Time in Seconds)
1	331	351
2	342	343
3	345	349
4	340	348
5	337	350
6	338	342
7	330	349
8	341	345
9	344	352
10	336	337
Average	338.4	346.6
Transfer rate (Kb/Second)	7.68	7.50
Transfer index	100	97.65

The obtained results show that the proposed solution is slightly affected the overall network performance which is decreased by 1.96 % in downloading. Table 5.6 show the transfer index for ordinary download versus extended download. The transfer index for uploading is 2.34 which, is shown in Table 5.7. This illustrates enhancement in security aspects which is due to the fact that the LAN network bandwidth is much larger than the bandwidth available from Internet. In addition, the bandwidth of the LAN is constrained by the available bandwidth from Internet when connecting to internet.

5.3 SUMMARY

This chapter discusses the test results including the server. In addition, the tests were also performed to evaluate the downloading and uploading time. The performance was also evaluated by comparing both the ordinary as well as extended firewall, and the results show that the system performance has been improved using the extended firewall.

UMP

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

Until recently, there is no unique security solution for a network. After all, it is necessary to accomplish different and multistage defense lines. The administrator must have good and updated knowledge for the software used inside the network, and what level of crisis their will have. It is preferable that the number of software that connects with the outside network is controlled.

Attacks from the outside are protected against very well by common firewalls, therefore the study was focused on enhancing the security of the firewall with respect to intruding attacks launched from the inside and no disclose information to the untrusted network except when the communication link originates from a trusted source (a trusted host running a trusted application, under an authenticated user). In order to protect the firewall against internal intruding attacks it needs more information than what is available in traditional IP-datagram. This needs some changes in the protected hosts since they must provide these extra information to the extended firewall. Furthermore, the capable to trust the information in the packets required calls for further changes to the extended firewall.

This thesis conveys important finding to solve internal intruding or inside attacks which can lead to a disaster, by deploying two algorithms one at each client and another at the server. The first will provide some important authenticated features to each packet destined to leave the network, while the second verifies the information, allowing the authenticated packet to pass into its destined while dropping and documenting the unauthorized one. Modifying or fabricating any IP packet does not enable the adversary to communicate with the external network, as the adversary would be lack of the capability to authenticate the IP packet to the extended firewall. The Host ID can be more reliable than the IP address for the connected host, especially when the host uses the DHCP.

The programmer can choose a Host ID and use it when developing software in order to guarantee that the copy of software is the only copy that cannot execute on another host. Using this method, the network administrator at the server can identify what connected applications are run on the clients. Both programs operate under Microsoft Windows operation system environment.

Finally, the proposed algorithm shows that it can be used for protection against internal intrusion. The results show that this algorithm can increase the performance of downloading and uploading in a local area network environment. This proposed method is suitable for organizations that have sensitive information. In addition, some employees of an organization do not have sufficient security knowledge. Therefore, they can consider this method to protect their local area network from internal attack.

6.2 RECOMMENDATION OF FUTURE WORK

In the presentation of the extended firewall, it had been divided into two parts, the authenticator and the verifier. However, there is no technical problem in incorporating the two into one and using it to build a personal firewall, i. e. a firewall that protects the single host that it resides on. Future work could be combining these two processes into one process. The authentication of the packets would not be needed though, since the firewall would authenticate information to itself.

With some changes, the programs can be used in such a way that suits another type of networking such as star, mesh and so on. Applying other methods of ciphering and comparing the performance with the proposed method is recommended. In addition, it is recommended to improve the ability of the proposed method to make the server as a controller for preventing any client from connecting to the outside, which is also another challenge.

The authentication daemon could obtain the set of networks, for which no diversion should take place, directly from the extended firewall when it creates a new session. Thus, it would be easier for the administrator to change subnets. This would have to be made on a host-by-host, but it would not be hard to make a system where hosts could be grouped when defining the diversion "rules". When the diversion rules are changed for a host, the verification daemon can automatically close the connection to this client, thereby forcing it to create a new session and to use the new diversion rules. Therefore, for the future work, the authentication and verification can be integrated in the same firewall.

UMP

LIST OF PUBLICATIONS

1. Norrozila Sulaiman **Muamer N. Mohammad**, “A Study on a Load Balancing Model for Improving Network Performance “, International Conference ICSECS' 09, Kuantan, Malaysia.
2. **Muamer N. Mohammad** , Norrozila Sulaiman and Osama Abdulkarim Muhsin, ”A New Intrusion Detection System Model for Local Network Based on Support Vector Machine“, International Conference MIC-CSC2009, Imman, Jourdan.
3. **Muamer N. Mohammad**, Norrozila Sulaiman and Osama Abdulkarim Muhsin, ” A Novel Intrusion Detection System by using Intelligent Data Mining in Weka Environment“, International conference WCIT2010, Istanbul, Turkey, Published by Procedia Computer Science 3(2011), index– Elsevier- ISI. Pages no. 1237–1242.
4. **Muamer N. Mohammad** and Norrozila Sulaiman,” A Novel Local Network Intrusion Detection System Based on Support Vector Machine“, submitted to Journal of Computer Science, ISI.

UMP

REFERENCES

Alex, X. L., and Torng, E. 2007. Firewall Compressor: An Algorithm for Minimizing Firewall Policies, supported in part by the National Science Foundation under Grant, Michigan State University.

Amorso, E. 1999. Intrusion Detection: An Introduction to Internet Survellance, Correlation, Trace Back, Traps, and Response, Intrusion.net.

Anderson, J. 1980. Computer Security Threats Monitoring and Surveillance, Technical Report, Fort Washington, Pennsylvania.

Anderson, F. and Karlsson, M. 2000. Secure Jini Services in Ad Hoc Networks, Royal Institute of Technology.

Bace, R. and Mell, P. 2002. Intrusion Detection Systems. National Institute of Standard and Technology.

Bagwill, R., Barkly, J. and Carnahan, L. 1994. Security in Open Systems, Published by Computer Systems Technology, U.S., Department of Commerce.

Balmer, S. 1999. Framework for a High Assurance Security Extension to Commercial Network Clients, Naval Postgraduate School, Monterey, CA.

Bentham, J. 2000. TCP/IP Lean Web Servers for Embedded Systems, CMP books.

Bishop, M. 2002. Computer Security: Art and Science, Addison Wesley.

Bora, T. 2000. Analysis for a Trusted Computing Base Extension Prototype Board. Naval Postgraduate School, Monterey, California.

Brown, D. J., Suckow, B. and Wang, T. 2001. A Survey of Intrusion Detection Systems, University of California, San Diego. USA.

Bryer, S. and Heller, S. 1999. Secure Local Area Network Services for a High Assurance Multilevel Network, Naval Postgraduate School, Monterey, CA.

Carr, S. 1998. Networking Concepts. University of Waterloo.

Chow, S. S. M., Hui, L. C.K., Yiu, S. M., Chow, K. P. and Lui, R. W. C. 2005. A generic anti-spyware solution by access control list at kernel level. *The Journal of Systems and Software*. (75): 227–234

Cutler, K. and Pole, J. 2002. Guidelines on Firewalls and Firewall Policy, MIS Training Institute, National Institute of Standards and Technology, Gaithersburg, MD 20899-8930, January.

Endorf, C., Schultze, G. and Mellander, J. 2004. Intrusion Detection and Prevention, McGrew-Hill/Osborne.

Irvine, C. E., Levin, T. E., Ngayen, T. D., Shifflett, D., Khosalim, J., Clarck, P. C., Wong, A., Afinidad, F., Bibighaus, D. and Sears, J. 2004. Overview of a High Assurance Architecture for Distributed Multilevel Security. *IEEE system*, United states Military Academy.

Daley, W. M. 1999. Data Encryption Standard (DES). Processing Standards Publication 46-3. U.S. Department Of Commerce/National Institute of Standards and Technology.

Denning, D. E. 1987. An intrusion detection Model. *IEEE Transactions on S.W engineering*. (13)**2**:222-232.

Duan, Z., Yuan, X. and Chandrashekar, J. 2008. Controlling IP Spoofing through Interdomain Packet Filters. *IEEE Transactions on Dependable And Secure Computing*. (5): 1.

Fisch, E. A. and White, G. B. 2000. Secure computers and networks: Analysis, design, and implementation by CRC press LLC.

Forouzan, B. 2003. *TCP/IP protocol Suite*. 2nd Edition, McGraw-Hill Higher Education.

Garuba, M., Liu C. and Fraites D. 2008. Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems, *In proc. Fifth IEEE int. conf*. 592-598.

Golnabi, K. and Al-Shaer, E. 2006. Analysis of Firewall Policy Rules Using Data Mining Techniques, *NOMS, 10th IEEE/IFIP Transactions*.305-315.

Gordon, A., Loeb, P., Lucyshyn, W. and Richardson, R. 2004. CSI/FBI Computer Crime and Security Survey, Annual Report, Computer Security Institute.

Gordon, B. and Gray, J. 2002. High Performance Computing: Crays, Clusters, and Centers, communications of the ACM.

Grout, V. and Davies, J. N. 2010. A Simplified Method for Optimising Sequentially Processed Access Control Lists, *Sixth Advanced International Conference on*

Telecommunications.

Grout, V., Davies J. and McGinn, J. 2007. An Argument for Simple Embedded ACL Optimisation. Computer Communications. **30** (2):280-287.

Habtamu, A. 2000. An Overview of Firewall Technologies, Norwegian Computing Center, January.

Hairui, W. and Hua, W. 2008. Research and Design of Multi-agent Based Intrusion Detection System on Wireless Network. International Symposium on Computational Intelligence and Design. *IEEE*.

Han, W., Xu, M., Zhao, W. and Li, G. 2010. A trusted decentralized access control framework for the client/server architecture. *Journal of Network and Computer Applications*. (33): 76–83.

Hombrebueno, D., Sicat, M. G. C. E., Niguidula, J. D., Chavez, E. P. and Hernandez, A. A. 2009. Symmetric Cryptosystem Based on Data Encryption Standard Integrating HMAC and Digital Signature Scheme Implemented in Multi-Cast Messenger Application. *Second International Conference on Computer and Electrical Engineering IEEE.*

Hu, Y. and Panda, B. 2010. Two-dimensional Traceability Link Rule Mining for Detection of Insider Attacks. Proceedings of the 43rd Hawaii International Conference on System Sciences. *IEEE Computer Society*, USA.

Hu, T. 2004. Design and Implement of Firewall-log-based Online Attack Design System, ACM, Zhejiang University of Technology.

Jones, A. and Sieken, S. 2000. Computer System Intrusion Detection: A Survey, Technical Report, Department of Computer Science, University of Virginia.

Khazal, H. 2004. A simulated IDS Using Packet Capture, Ph.D. Thesis, Computer Science Dep. of the University of Technology, Baghdad.

Krsul, I. 1997. Computer Vulnerability Analysis, Technical Report CDS-TR-97-026, Department of Computer Science, Purdue University.

Lakshmi, V. and Agrawal, D. 2001. An Optimized Inter-router Authentication Scheme for Ad Hoc Networks, *International Conference of Wireless Communications*, PP 129-146.

Lowery, J. C. 2002. Computer System Security, Ph.D. Thesis.

Lunt, T. 1993. A Survey of Intrusion Detection Techniques. *Computer & Security*. 405-418.

Mark, J. and Benjamenn, A. 200l. A Building Block Approach to Intrusion Detection, CERIAS Purdue University.

Marks, L. 2004. Security Inside Out, The Systems Consulting Consortium Inc.

Mell, P. 2002. An Overview of Issues in Testing Intrusion Detection System. Sponsored by the Defere Advanced Research Projects Agency under Air Force Contract.

Microsoft. 2001. Windows XP Device Driver Kit (DDK), Microsoft Corporation.

Moller, J. and Donbaek, T. 2001. Internal Network Security. Department of Computer Science at the University of Aarhus.

Pachghare, V. K., Kulkarni, P. and Nikam, D. M. 2009. Intrusion Detection System Using Self Organizing Maps, *International Conference on Intelligent Agent & Multi-Agent Systems*.

Platos, J. Snasel, V., Kromer, P. and Abraham, A. 2009. Detecting Insider Attacks Using Non-negative Matrix Factorization. *Fifth International Conference on Information Assurance and Security*. *IEEE.*

QU, Z. 2009. Investigation of PCFA in Assessing Main Function Indexes of Intrusion Detection System in Network Security. *International Symposium on Information Engineering and Electronic Commerce.IEEE*. Computer Society Washington, DC, USA.

Rossetti, R. 2000. A Mail File Administration Tool For a Multilevel High Assurance LAN, Naval Postgraduate School, Monterey, CA.

Russinovich, M. and Solomon, D. 2004. *Microsoft Windows Internals*. *4th Edition, Microsoft Press.*

Taherkhani, S., Ever, E. and Gemikonakli, O. 2010. Implementation of Non-Pipelined and Pipelined Data Encryption Standard (DES) Using Xilinx Virtex-6 FPGA Technology. *10th IEEE International Conference on Computer and Information Technology*.

Zaman, S. and Karray, F. 2009. TCP/IP Model and Intrusion Detection Systems.

International Conference on Advanced Information Networking and Applications Workshops.

Seberry, J. and Pieprzyk, J. 1989. Cryptography- An Introduction to Computer Security. Prentice Hall of Australia Ltd.

Shanbhag, S. and Wolf, T. 2008. Massively parallel anomaly detection in online network measurement, *the Seventeenth IEEE International Conference on Computer Communications and Networks (ICCCN).*

Shaw, H., Hussein, S. and Helgert, H. 2010. Prototype Genomics-Based keyed-Hash Message Authentication Code Protocol. *Second International Conference on Evolving Internet*

Shipley, G. 2001. Maximum security a Hacker's Guide to Protecting Your Internet Site and Networking, *SAMS*. 3nd Edition.

Petrovic, S. and Bakke, S. 2008. Improving the Efficiency of Misuse Detection by Means of the q-gram Distance. *The Fourth International Conference on Information Assurance and Security*.

Stallings, W. 1999. *Cryptography and Network Security*, 2nd Edition, Prentice- Hall.

Stallings, W. 2000. Network Security Essentials: Application and Standards, Printce-Hall.

Stallings, W. 1997. *Data and Computer Communication*, five edition, ISBN 0-13-571274-2, New Jersy, Printice-Hall, Inc.

Steven, R., Gihan, V. and Doglus, L. 1991. DIDS: Distributed Intrusion Detection System, Motivation, Architecture and Early Prototype, *In the Proceedings of the 14th National Computer Security Conference, Washington*.

Strassberg, K., Rollie, G. and Gondek, R. 2002. Firewalls: the Complete Reference, McGrew-Hill/Osborne.

Strand, L. 2004. Adaptive Distributed Firewall Using Intrusion Detection, University of Oslo.

Tangney, B. 1988. *Local area networks and their applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

Tihomir, K. and Predrag, P. 2007. Optimization of Firewall Rules. *29th International Conference on Information Technology Interfaces- Zagreb*. University Computing

Centre. 685-690.

Uribe, T. E. and Cheung, S. 2004. Automatic Analysis of Firewall and Network Intrusion Detection System Configurations, ACM, Washington, DC, USA.

Turkia, M. 2002. Introduction to Intrusion detection systems NEC Research Institute.

Vakili, G., Riahy, G. H. and Rezaie, A. H. 2006. Combination of a Transparent Firewall and a DoS Attack Detection System. Second International Conference on Information and Communication Technologies, ICTTA'06, Damascus, Syria.

Varekova, P., Varekova, I. and Cerna, I. 2010. Automated Computing of the Maximal Number of Handled Clients for Client-Server Systems, *Electronic Notes in Theoretical Computer Science.* (260)243–259.

Vimalathithan, R. and Valarmathi, M. L. 2009. Cryptanalysis of S-DES using Genetic Algorithm. *International Journal of Recent Trends in Engineering.* (2)4.

Wilson, J. 2000. A Trusted Connection Framework For Multilevel Secure Local Area Networks, Naval Postgraduate School, Monterey, California.

Yurcik, W. 2002. Controlling Intrusion Detection System by Generating False Positives: Seqealine Proof-of- Concept. *27th Annual IEEE Conference on Local Computer Networks.*

Yanzhi, R., Choo, M. C., Jie, Y. and Yingying, C. 2010. Detecting Blackhole Attacks in Disruption-Tolerant Networks through Packet Exchange Recording. *World of Wireless Mobile and Multimedia Networks (WoWMoM) Canada, IEEE International Symposium.*

Ying, L., Yan, Z. and Yang-Jia, O. 2010. The Design and Implementation of Host-based Intrusion Detection System. *Third International Symposium on Intelligent Information Technology and Security Informatics, IEEE.*

Zaw, M. P. P. and Su, S. M. M. 2008. Design and Implementation of Client Server Network Management System for Ethernet LAN. World Academy of Science, Engineering and Technology.

Zhang, N., Yu, W., Fu, X. and Das, S. K. 2010. Maintaining Defender's Reputation in Anomaly Detection Against Insider Attacks. IEEE Transactions on Systems. (40) 3.