الخلاصة

تتطلب عملية ادارة الشبكات الحاسوبية في الشركات و الجامعات و غيرها متابعة الافراد العاملين عليها من اجل السيطرة على فعاليات الشبكة لكشف ومنع اي عملية سوء استخدام للشبكة.

من الطرائق المستخدمة في عملية المراقبة هي طريقة كشف الشذوذ(Anomaly) Detection، كشف الاستعمال السيء (Misuse Detection) وطريقة مراقبة الهدف (Target Monitoring).

في هدا البحث تم التركيز على بناء نموذجين من المراقبة ينتميان الى طريقة مراقبة الهدف(Target Monitoring) وفي ظل نظام التشغيل نوافد(Windows). الاول هو المراقبة الفورية او المباشرة(Online Monitoring) والاخر المراقبة غير فورية اوغيرالمباشرة(Offline Monitoring).

النوع الاول المراقبة الفورية او المباشرة Online Monitoring تهتم بتقديرات الوقت العام لعملية التوليد ، النشر، و تمثيل البيانات المراقبة. في هذا النموذج تراقب شاشة حاسبة المستخدم وحركة المؤشر(الفأرة) بالاضافة الى ما يتم كتابته باستخدام لوحة المفاتيح. يمكن لمراقب الشبكة يمكن أرسال تنبيه الى مستخدم الحاسبة التي يتم مراقبتها، اذا قام المستخدم باستخدام الحاسبة بشكل غير مناسب اواستخدام خاطى وغيرها كما يمكن اطفاء الحاسبة ان تطلب الامر. وتم استخدام فكرة برامجيات الزبون والخادم (Client/Server) و أستخدام التجاويف (Windows Sockets) لنموذج المراقبة الفورية.

أما النوع الثاني المراقبة غيرالفورية او غير مباشرة Offline Monitoring لا تتوقف على قيود الوقت. لذلك،فان مراقب الشبكة ربما يحصل على بيانات المراقبة في وقت اعتباطي بعد ان تتولد هذه البيانات بواسطة وكلاء(Agents) المراقبة مما يتطلب مساحة خزنية هائلة لكي تعرض على المراقب عند الطلب، لذلك، حدّدت المراقبة إلى لوحة المفاتيح فقط كون المعلومات المتولدة هي فقط على شكل نص من الرموز.

كلا النموذجين تم بنائهما باستخدام لغة (Visual Basic 6) ونظام direxX 7 المجهز بولسطة بيئة النوافذ. وتم اختبار النظامين في بيئة ويندوز XP ، ويندوز ٢٠٠٠ ، و ويندوز Melleniume. وقد اثبت النظام كفاءة متميزة.

Abstract

Network operators and administrators have a substantial interest to trace the state and performance of their networks and their components. Administrators needed to observe users to avoid any attack attempts. This can be done through System Monitoring.

Traditional types of monitoring system include **Anomaly Detection**, **Misuse Detection** and **Target Monitoring**. This work concerns with the development of two types of target monitoring works under windows environments. These are **Online Monitoring** and **Offline Monitoring**.

Online monitoring concerned with time constraints on the overall time its takes to generate process, disseminate, and present monitoring data. Online monitoring can monitor the screen picture, mouse motion, and keystrokes. The administrator can send warning messages to the remote computer user. When the user performs suspicious actions on his computer, in this case, the administrator can force the remote user to logoff or shutdown windows. In the online Monitoring, uses the concept of client/server software system and Windows sockets.

Offline monitoring do not concerned time constraints. Hence, administrator may obtain monitoring data at an arbitrary time after its generation by monitoring agents. In this phase the monitoring limited to the keyboard only.

Both the online monitoring and offline monitoring are implemented using Visual Basic programming Language version 6 (VB) and direxX 7 system provides by windows environment. The developed systems were tested under different environments (Windows 2000, Windows XP) and the results obtained are quite encouraging.

الاسم:- دلال نعيم حمود الزيدي

العنوان:- بغداد – الكرخ – السيدية- م٨٥٣- ز ١١- دار ١٥

البريد الالكتروني:-

Dalal_hmood@yahoo.com

تاريخ المناقشة: ٢٠٠٥/٥/٣

لاجل ايجاد معلومات عن موضوع الاطروحة استعمل الكلمات التالية:-

Monitoring System
Target Monitoring
Online and Offline Monitoring Systems
Networks
Protocols
Ports

Acknowledgment

First of all I would like to express my sincere qratitude and appreciation to my supervisors Dr. Taha S. Bashaga and Dr. Venuse W. Samawi for their able guidance, supervision and untiring efforts during the course of this work.

Special thanks to the college of science, dean of the collage for the continuous support and encouragement during the period of my study.

Finally, my special thanks to my family for their contiouous support and encouragement during the period of my study.

Appendix E

Appendix A

Network Monitoring

Monitoring tools are classified into two categories, according to type of functions and output they provide:

1- Fault Monitoring

Fault monitoring is a simple but pragmatic approach to network monitoring, as it focuses on the state of the various network elements only, i.e. whether an element is operational and whether it is accomplishing the services and tasks that are assigned to it. It doesn't provide any information on the performance or activity of the network and its nodes. Service faults are detected instantly whereupon the operator is alerted.

3.2 Performance Monitoring

Performance Monitoring is a more sophisticated and more instructive concept, as it allows the measuring and tracing of the progression of the actual usage and performance of various network features, e.g. the output traffic of a router or the number of users logged onto a server.

Moreover, depending on the implementation, the progression of the network activity might be stored into a data base to allow further statistical processing and extraction. Performance monitoring resp. measurement for large network structures is complex, and requires careful consideration of the following aspects, which are explained below:

1-Active vs. Passive Measurement

Active measurements inject test packets into the network and trace their behavior, whereas passive measurements observe the actual

traffic without generating additional traffic. However, passive monitors must process the entire load of a link to determine packets of interest, which might be problematic on high-speed links. Usually traffic flow data is collected either from routers and switches, or measured by stand-alone traffic meters.

Monitor Placement within the Network

Another crucial issue is where to place the meter(s) within the network infrastructure.

One approach is to identify the links, which handle the biggest amount of traffic and place monitors there. Alternatively, one could begin by monitoring traffic at all border routers of the network.

Metric Selection

Popular network metrics are *throughput*, *latency*, *packet loss*, *link utilization* and as aforementioned *availability*.

Data Collection and Archiving

To allow statistical processing over a long period of time, the collected data must be archived. A common approach involves the building of a trace file repository that enables users to request a report on specific hosts, metrics or time intervals.

Once a measurement data repository is established, it is important to provide a clear, easy-to-use interface to the data. There is no point in collecting data if users can’t access and interpret it easily.

Appendix B

ISO Reference Model: -

The International Standards Organization (ISO) open Systems Interconnection (OSI) model consists of Layers by which a network communication occurs. The OSI reference model consists of seven layers. Each Layer serves a separate function. Figure (1) shows a seven-layer representation of the OSI model.

Application	Layer 7: The Application layer within which all network application reside.
Presentation	Layer 6: The Presentation layer provides data representation support.
Session	Layer 5: The Session layer provides for data exchange through dialogs.
Transport	Layer 4: The Transport layer provides end-to-end communication.
Network	Layer 3: The Network layer provides internetwork connectivity.
Data Link	Layer 2: The Data Link layer provides protocols for transmitting and receiving data between directly linked systems.
Physical	Layer 1: The Physical layer dictates the physical characteristics of communication.

Figure (1) The OSI-RM

Layer 7 all of the capabilities of networking begin in the Application layer. File transfer, messaging, web browsing, and other applications are in this layer. Each such application will appropriately invoke processing of data for transmission through well-defined interfaces to layer(s) below this one.

Layer 6 the Presentation layer is responsible for data formatting. It takes care of such things as bit and byte ordering and floating-point representation.

Layer 5 the Session layer handles the exchange of data through dialog procedures or chat or conversation protocols.

Layer 4 the Transport layer is responsible for the reliable transfer of data between systems. It manages the communication session including flow control, ordering of information, error detection, and recovery of data.

Layer 3 the Network layer owns the responsibility of delivering data between different systems in different interconnected networks (Internets).

Layer 2 the Data Link layer provides rules foe sending and receiving data between tow connected nodes over a particular physical medium.

Layer 1 the Physical layer defines the required hardware, such as cables and interfaces, for a given medium of communication, such as electrical, radio frequency, and light-based. In this way, methods for transmitting and receiving bit-streams of information are defined.

Appendix E

Components of DirectX Foundation

DirectX Foundation is made up of the following components:

- **Direct3D** provides a high-level Retained Mode interface that allows applications to easily implement a complete 3-D graphical system, and a low-level Immediate Mode interface that lets applications take complete control over the rendering pipeline.
- **Direct3D** now includes a **Direct3DX** Utility Library that is a helper layer to simplify common tasks encountered by 3-D graphics developers.
- **DirectDraw** accelerates hardware and software animation techniques by providing direct access to bitmaps in off-screen display memory, as well as extremely fast access to the blitting and buffer-flipping capabilities of the hardware.
- **DirectInput** provides support for every kind of input and force-feedback device. For more information.
- **DirectMusic** works with message-based data and provides a complete system for playing music and ***DLS***-based sound effects with run-time composition and variation.
- **DirectPlay** makes connecting games over a modem link or network easy.
- **DirectSetup** provides a one-call installation procedure for DirectX.
- **DirectSound** enables playback of wave sounds and includes support for hardware and software mixing, 3-D positioning, and sound capture.

Why Use DirectInput?

Aside from providing new services for devices not supported by the Microsoft Win32 API, DirectInput gives faster access to input data by communicating directly with the hardware drivers, rather than relying on Microsoft Windows messages.

DirectInput enables an application to retrieve data from input devices even when the application is in the background. It also provides full support for any type of input device, as well as for force feedback.

The extended services and improved performance of DirectInput make it a valuable tool for games, simulations, and other real-time interactive applications running under Windows.

Why Use DirectDraw?

The DirectDraw component brings many powerful features to you, the Windows graphics programmer:

- The ***hardware abstraction layer (HAL)*** of DirectDraw provides a consistent interface through which to work directly with the display hardware, getting maximum performance.
- DirectDraw assesses the video hardware's capabilities, making use of special hardware features whenever possible. For example, if your video card supports hardware blitting, DirectDraw delegates blits to the video card, greatly increasing performance. Additionally, DirectDraw provides a hardware emulation layer (HEL) to support features when the hardware does not.
- DirectDraw exists under Windows, gaining the advantage of 32-bit memory addressing and a flat memory model that the operating system provides. DirectDraw presents video and system memory as large blocks of storage, not as small segments. If you've ever used segment: offset addressing, you will quickly begin to appreciate this "flat" memory model.
- DirectDraw makes it easy for you to implement ***page flipping*** with multiple ***back buffers*** in full-screen applications.
- Support for clipping in windowed or full-screen applications.
- Support for 3-D z-buffers.
- Support for hardware-assisted overlays with z-ordering.

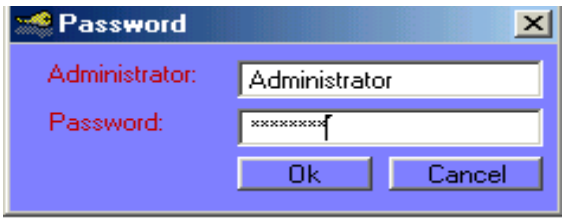
- Access to image-***stretching*** hardware.
- Simultaneous access to standard and enhanced display-device memory areas.
- Other features include custom and dynamic palettes, exclusive hardware access, and resolution switching.

Appendix H
TM User Interface

1- How To Use TM System

The designed TM system with 4-forms to enable the administrator to monitors the remote computer.

- ***Password Form (figure 1):*** when the administrator start the TM system the new window as shown in figure (1), appears to check the authenticity of the administrator.



Figure(1) Password Form

when administrator enters the correct Administrator and password, the monitoring system window as shown in figure(2) will appear to the administrator.

- ***Monitoring Form(figure 2):*** when the administrator start the TM system, the main form is activated which consisting of:

1- ***Computer Name Combobox:*** allows the administrator to select the the computer name that being monitored to provide the ability to connect to that computer.

2- ***Moitoring traced***: this part display some information about the monitored computer example, computer name, date, time,etc.

3- ***Remote Computer:*** this part display the images of the remote screen and display the movement of mouse.

4- ***Keyboard information form:*** this form responsible to display any key pressed by the user on the remote computer.

5- ***Left, Right, and Middle buttons***: these labels display the state of mouse buttoms. If the user on remote computer press the left buttom, change the caption of left buttom label to ***true*** else that change to the ***false.***

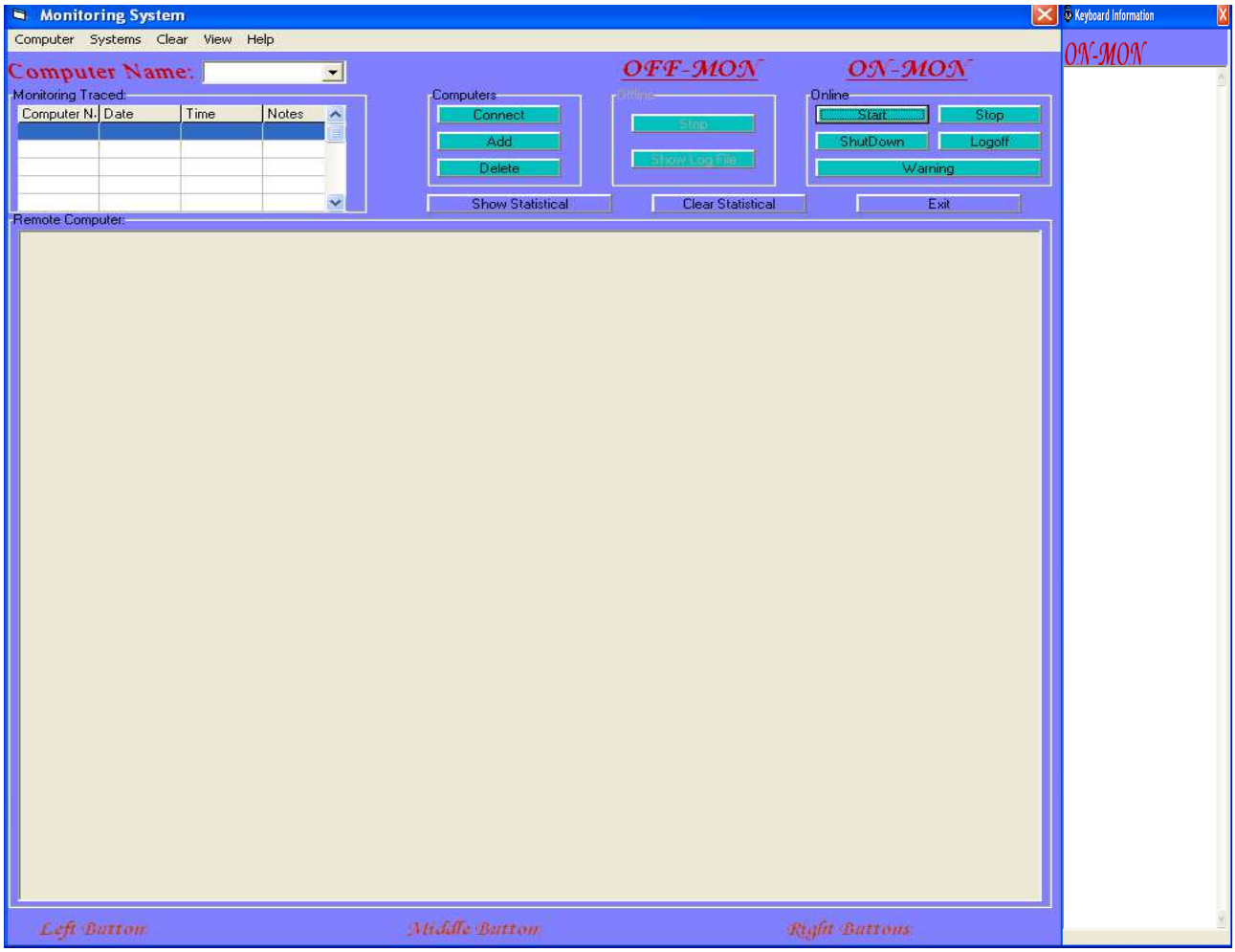


Figure (2) Main Window of Monitoring System

In addition to these parts, the client form provides set of frames contains buttons that allows administrator to perform the monitoring functions on the remote computer. The function of each button is illustrated below:

1- Computer Frame:

- **Connect:** the connect button is used to connect to the computer is specified in the computer name combobox.
- **Add Computer**: this button allows administrator to add new computer.
- **Delete Computer:** this button allows administrator to delete computer after selected computer by computer name combobox.

2- Online Frame:

- **Start:** this button starts the transfer information between the two sides by getting information about the keyboard, mouse, and screen devices in the remote computer.
- **Stop:** this button stops monitoring the keyboard, mouse, and screen devices in the remote computer.
- **Logoff**: this button forces the remote computer to logoff.
- **Shutdown:** this button makes the shutdown to the remote computer.
- **Warning:** this button displaying a warning massage to the remote computer.

3-Offline Frame:

- **Stop offline: :** this button stops the monitored to keyboard device in the remote computer.
- **Show Log File:** this buttons allows to administrator to view the log file, when press this command, the offline monitoring form is appears,as shown in figure (3).

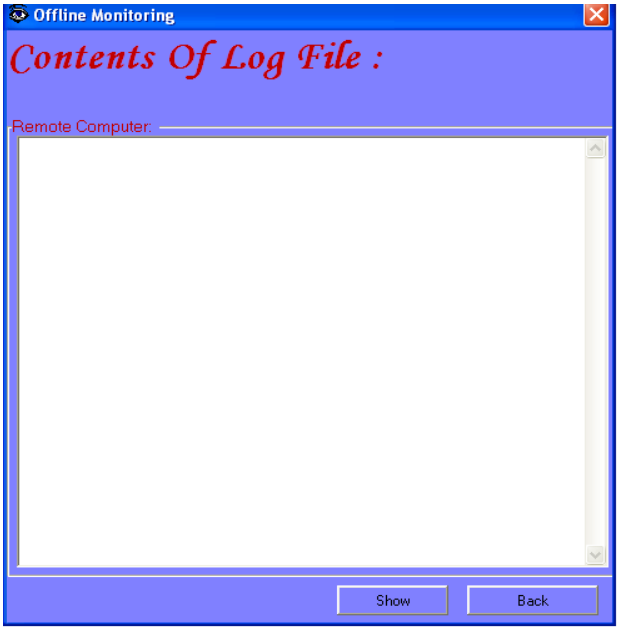


Figure (3) Offline Monitoring

When the administrator press the show button that shows the contents of log file displaying in this form and if the administrator press the back button that hide this form and return to the monitoring system form figure (2).

In addition to these buttons, the client form provides three buttons that allows administrator to display or delete some information. The function of each button is illustrated below:

- **Show Statistical**: this button enables the administrator to show all information about each computer monitored when the monitoring system running. When this button is pressed, the system will display the content of the statistical file as shown in figure (4).

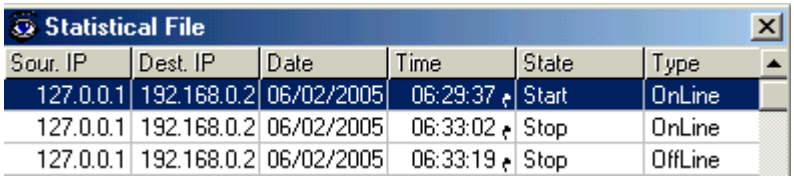


Figure (4) Statistical file

- **Clear Statistical:** this button enables the administrator to clear (delete) the statistical file.
- **Exit:** this button to end the execution of the monitoring system.

Also main form contained menu of five options. These options illustrated below:

1- **Computer**: this option allows administrator to add or delete computer.

2- **System**: this option allows administrator to movement between online frame and offline frame.

3- Clear: which provides set of options that allows the administrator to **clear window** or **clear statistical**:

- **Clear window:** this function is clear the remote computer window, keyboard information window and the computer name text.

4- View: this option contains three function:

- **Hide/Show Client Window:** this function gave the ability to hide or show the client window.
- **Hide/Show keyboard Information Window:** this function gave the ability to hide or show the Keyboard information window.
- **Hide/Show Commands Window:** this function gave the ability to hide or show the commands window.

5- Help: this option contains two function:

- **Help about online monitoring:** this function appears the massage box contain short informtion about the online monitoring system**.**
- **Help About offline monitoring:** this function appears the massage box contain short informtion about the offline monitoring system**.**

Example: when the administrator wants to monitor the remote computer he should perform the following steps:

Step 1: the administrator select the remote computer and clicking on the connect button as in figure (5).

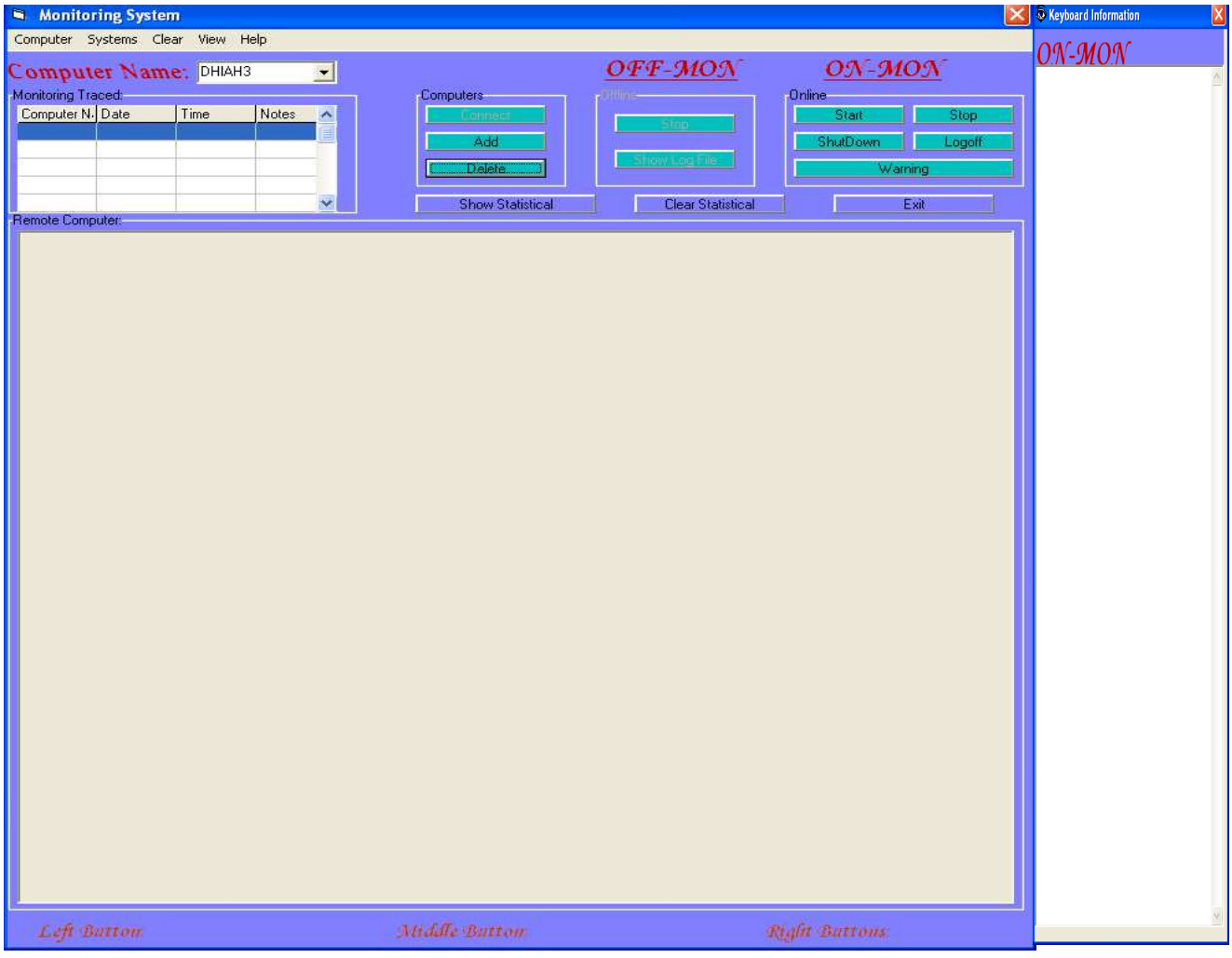


Figure (5)

Step 2: the administrator clicking on the start button that starts monitor as in figure (6).

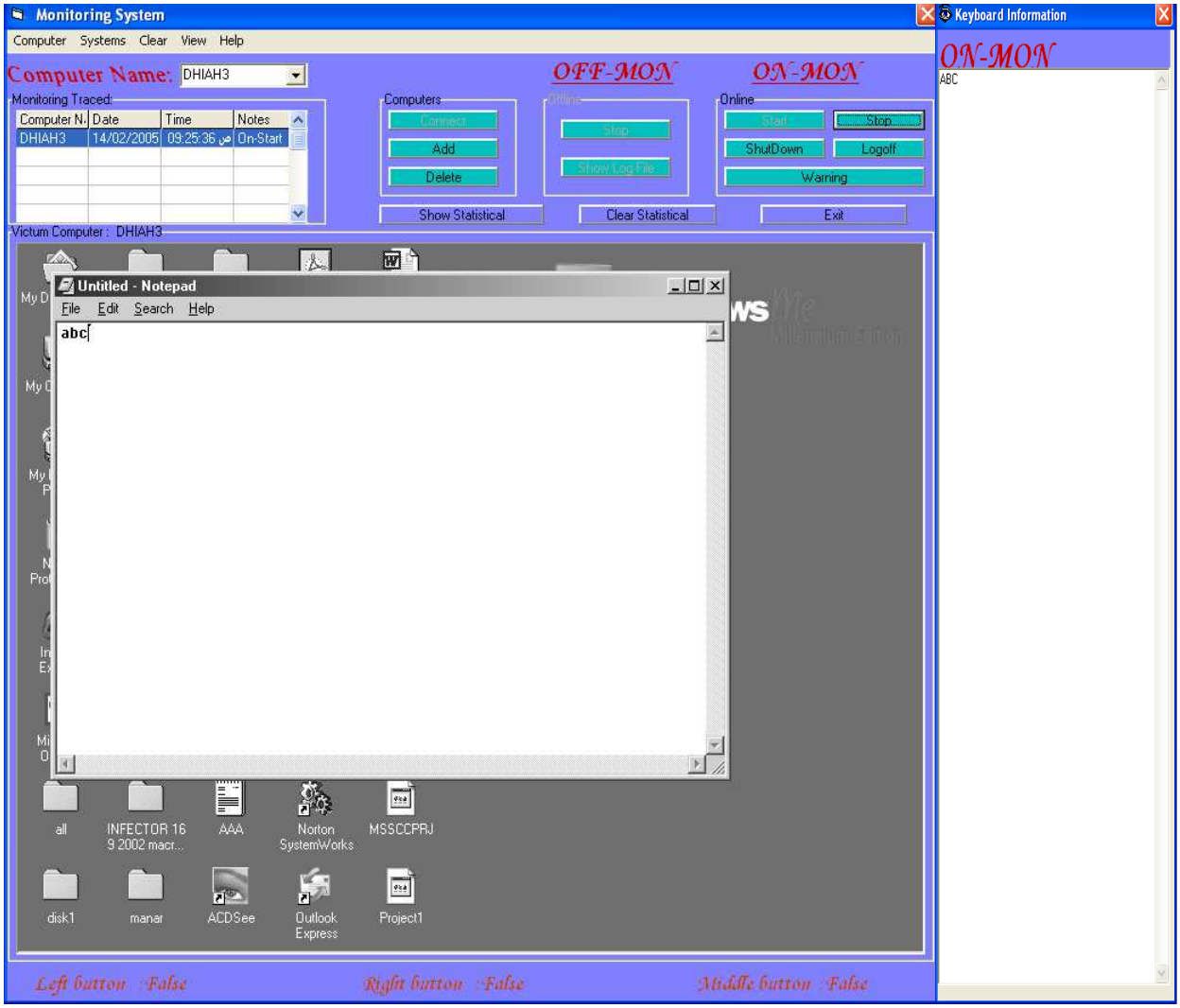


Figure (6)

Step 3: the administrator clicking on the stop button that if he wants to stop monitoring as in figure(7).

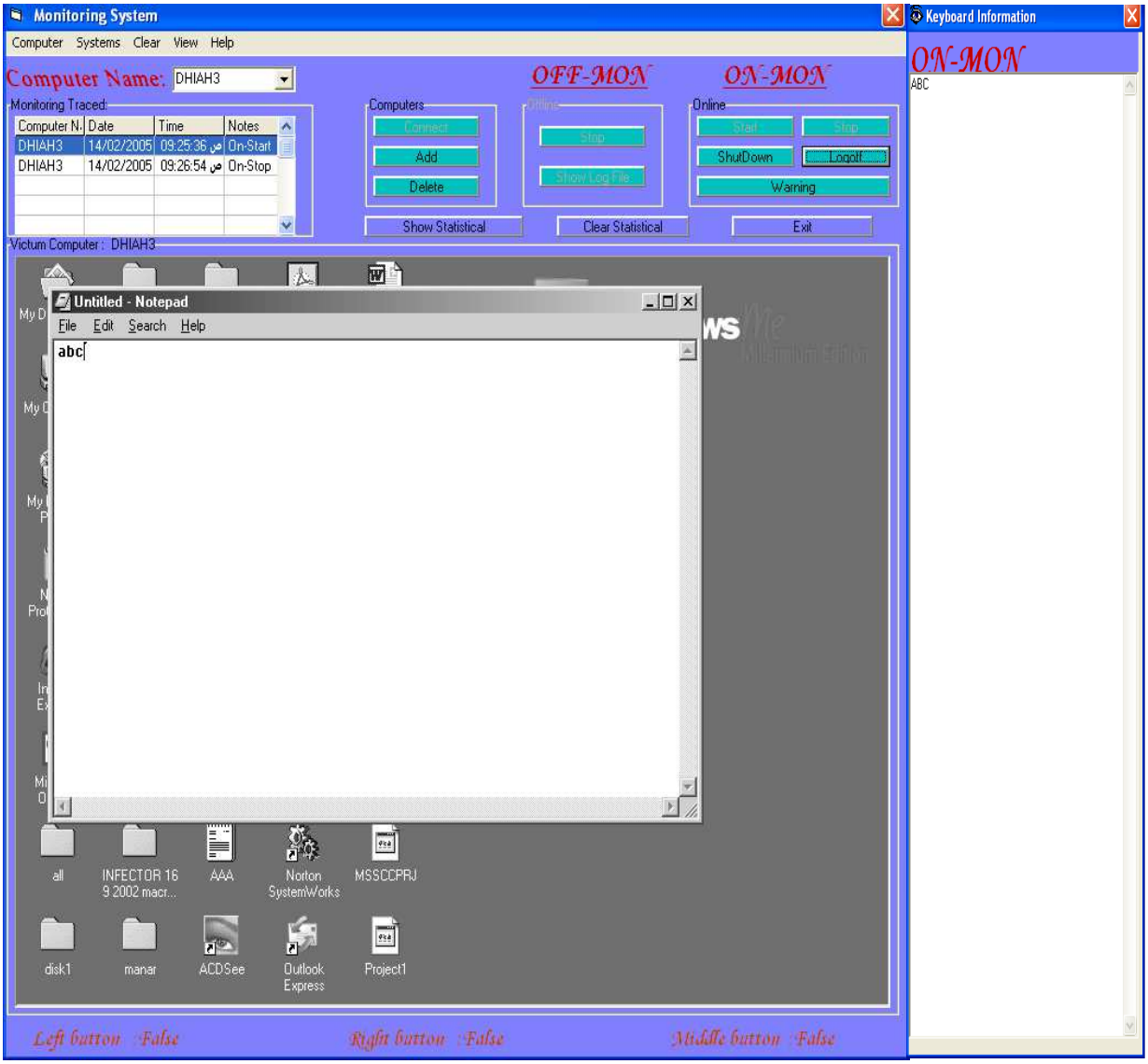


Figure (7)

Appendix C

Protocols in the TCP/IP model

1. Hardware Layer

The Hardware layer is responsible for exactly that-hardware. This includes cables, interface cards, and repeaters. It accepts the data passed to it by the Network Interface Layer.

2. The Network Interface Layer

Network Interface layer as a collection of device drivers. Its responsibility is to prepare the data passed to it from the Internet layer for signaling. The network interface layer may be:

2.1 ARPANET: is consisting of a subnet and host computers. The subnet would consist of minicomputers called IMPs (interface Message Processors) connected by transmission lines. For high reliability, each IMP would be connected to at least two other IMPs. So if some lines and IMPs were destroyed, messages could be automatically rerouted along alternative paths.

2.2 SATNET: - Southern Arizona Trauma Network is a non-profit organization whose mission continues to be that of providing trauma related educational opportunities to trauma professionals throughout southern Arizona, the state and the region.

2.3 Packet radio: - is a form of digital data transmission used in amateur radio to construct wireless computer networks. Its name is a reference to the use of packet switching between network nodes,

which allows multiple virtual circuits to coexist on a single radio channel.

2.4 LAN: Local Area Network, generally called LANs are privately owned networks within a single building or campus of up to a few kilometers in size. They are widely used to connect personal computers and workstations in company offices and factories to share resources (e.g., printers) and exchange information.

3. Application Layer:-

3.1 Telnet: The Telnet program provides a remote login capability. This lets a user on one machine log onto another machine and act as though he or she were directly in front of the second machine. The connection can be anywhere on the local network or on another network anywhere in the world, as long as the user has permission to log onto the remote system.

Telnet can be use when need to perform actions on a machine across the country. This isn't often done except in a LAN or WAN context, but a few systems accessible through the Internet allow Telnet sessions while users play around with a new application or operating system.

3.2 File Transfer Protocol: File Transfer Protocol (FTP) enables a file on one system to be copied to another system. The user doesn't actually log in as a full user to the machine he or she wants to access, as with Telnet, but instead uses the FTP program to enable access. Again, the correct permissions are necessary to provide access to the files.

Once the connection to a remote machine has been established, FTP enables you to copy one or more files to your machine. (The term *transfer* implies that the file is moved from one system to another but the

original is not affected. Files are copied.) FTP is a widely used service on the Internet, as well as on many large LANs and WANs.

3.3 Simple Mail Transfer Protocol: Simple Mail Transfer Protocol (SMTP) is used for transferring electronic mail. SMTP is completely transparent to the user. Behind the scenes, SMTP connects to remote machines and transfers mail messages much like FTP transfers files. Users are almost never aware of SMTP working, and few system administrators have to bother with it. SMTP is a mostly trouble-free protocol and is in very wide use.

3.4 Domain Name System (DNS)

Domain Name System (DNS) enables a computer with a common name to be converted to a special network address. For example, a PC called Darkstar cannot be accessed by another machine on the same network (or any other connected network) unless some method of checking the local machine name and replacing the name with the machine's hardware address is available. DNS provides a conversion from the common local name to the unique physical address of the device’s network connection.

Appendix D

Well – known ports

Some of the most common well-known ports:

Default port	Service name	Application
20	ftp-data	FTP (File Transfer Protocol) data
21	ftp	FTP (File Transfer Protocol) control
23	telnet	Telnet terminal handler
25	smtp	SMTP (Simple Mail Transfer Protocol)
53	domain	DNS (Domain Name Service) lookup
79	finger	Finger
80	http	HTTP (Hypertext Transfer Protocol)
110	pop3	POP3 (Post Office Protocol 3)
113	auth	Ident Authentication Service
119	nntp	NNTP (Network News Transfer Protocol)
137	nbname	NetBIOS name (Microsoft Networking)
138	nbdatagram	NetBIOS datagram (Microsoft Networking)
139	nbsession	NetBIOS session (Microsoft Networking)
143	imap	IMAP (Internet Message Access Protocol)
194	irc	IRC (Internet Relay Chat)
389	ldap	LDAP (Lightweight Directory Access Protocol)
443	https	HTTPS (Secure HTTP)

بسم الله الرحمن الرحيم

قَالُواْ سُبْحَانَكَ لاعِلْمَ لَنَآ إِلا مَا
عَلمْتَنَآ إِنكَ أَنْتَ العَلِيمُ
الحَكِيمُ

صدق الله العلي العظيم

البقرة-٣٢

Certification of the Examination Committee

We chairman and members of the examination committee, certify that we have studied the thesis entitled (**Windows Based Target Monitoring System**) presented by the student **Dalal Naeem Hmood Al-Zaidi** and examined her its content and in what is related to it, and we have found it worthy to be accepted for the degree of Master of Science in Computer Science with grade Execllent.

Signature:
Name: **Dr. Sattar B. Sadkhan**
Title: **Assistant Professor**
Date: / /2005
(Chairman)

Signature:
Name: **Loay E. George**
Title: : **Chief Researcher**
Date: **/ /2005**
(Member)

Signature:
Name: **Ban Nadeem Thanoon**
Title: **Lecturer**
Date: **/ /2005**
(Member)

Signature:
Name: **Dr. Taha S. Bashaga**
Title: **Lecturer**
Date: **/ /2005**
(Supervisor)

Signature:
Name: Dr. **Venus W. Samawi**
Title: **Assistant Professor**
Date: **/ /2005**
(Supervisor)

Signature:
Name: **Laith A. Al-Ani**
Title: **Dean of Collage of Science**
Date: **/ /2005**

Chapter Four
Discussion, Conclusions and Future Work

4.1 Discussion and Conclusions

This work concerned with the development of a Target Monitoring (TM) system, which enables network administrator to watch multiple computers connected throught a LAN with one client and many servers. TM is developed and tested under Windows using Microsoft Visual Basic and directX 7 system.

Several problems are raised during the system development:

- The first problem concerned with the computer remote control at which it is not possible to get information from remote computer unless there is an Agent installed on the remote computer which is responsible for providing the Administrator with the needed information throught the network connection.
- Another problem, the size of desktop screen image is huge. It is very difficult to send it as one packet. Therefore, the image must be divided into small packets and send them one after another.
- The CPU usage in the agent side is the maximum because the agent is responsible for taking a snapshot to the user desktop screen than send it to the administrator throught the network.

  The CPU usage in the administrator side is smaller than the agent side, but the CPU usage increased if more than one Agent are monitored.

4.2 Future Work

1- Develop Monitoring and Controlling system:

Add some other controlling on the monitoring system (control on the mouse and key events). The administrator can control both of them to change the user action, or prevent the user from some unallowable action.

2- Screen Picture Compression:

to increase the image transfer, the size of the image must be reduced. There is more than one way to do that. By using the image compression techniqes, or using another techniqe like XOR operation to get the difference between two images, then send the difference to update the old image in the other side of the network instead of sending the whole image, which will reduce the amount of data needed to be transferred.

3-Develop a Fault Monitoring

Fault monitoring is a simple approach to network monitoring, as it focuses on the state of the various network elements only.

4- Monitoring System For Multi node

Construct a monitor system that monitors more than one computer at the same time and measure the maximum number of computers that could be monitored without effecting the system and make some action if the user make an illegle operation.

Chapter One
Introduction

1.1 Preface

Network operators and administrators have a substantial interest to trace the state and performance of their networks and network components. Administrators needed to observe or monitor users from his place [Dan03]. This can be done through System Monitoring. Monitoring system is a beneficial for administrators that allows them to efficiently trace system activity. It can watch events on the system, or a large number of systems, by logs specified activity[Tod01]. Monitoring system may be able to manually perform either by physically accessing each system directly, or by performing the processes remotely over a network connection. In these situation, automated monitoring tools and network management systems are frequently used.

Automated tools which are called ***"agents"*** reside on the host computer (i.e. the computer being monitored) and communicate with a management console via a network connection.The agent watches usage patterns, processor workload, log files, and other items for signs of a problem [Wil03].

Monitoring the basic activities and needs of a system involves observing such things as usage levels, available resources and the overall health of a given system ,when changes are detected by monitoring; the administrator may be alerted before the system crashes or stops responding. This gives the administrator more flexibility in mitigating the risks of its occurring [Kel02].

Monitoring software is legal because the vast majority of its developers are law-abiding people who create their programs exclusively

for legitimate purposes. There are many situations when monitoring the computer activity is perfectly legal: the parents can use key loggers to protect their children from online abuse, the companies may use Internet monitoring software to ensure that their employees don't misuse corporate Internet connection and so on.

One of the benefits of security monitoring is the early identification of wrongdoing or new security vulnerabilities. This early identification can help to block wrongdoing or venerability before harm can be done or at least to minimize the potential impact[Cis04].

The most popular Monitoring tools are **Computer Security Monitoring (CSM)** and **Network Monitoring (NM) .**

Computer security monitoring (CSM) systems have been partitioned into two categories, these are **Anomaly detection** and **Misuse detection**. There is a less frequently articulated third category called **Target monitoring** that can be considered a special subcategory of both anomaly and Misuse detection .

Network Monitoring tools are classified into two categories, these are ***Fault monitoring*** and ***Performance monitoring***(for more information see Appendix A)[Kup04].

1.2 Literature Survey

Various efforts in the field of monitoring system were introduced, some of these efforts are summarized below: -

- ***Daniel Grub [Dan03]*** developed method for monitoring system of a large Wireless LANs (WLANs). There are two categories of monitoring (fault monitoring and performance monitoring) according to the type of functions and output that provides.

- ***ELM Enterprise Manager [Elm00]:*** ELM Log Manager is a system to see all events log with unrivaled clarity. ELM Log Manager is a client/server application that administrator tasks to monitor and manage user event logs.
- ***INFOSEC group [INF02] Computer Technology Associates (CTA)*** develop a Non-Real time monitoring to detect internal attacks and misuses as well as external attacks.

  When a review of a log file or other non-real time detects an event (e.g., attack or misuse), the reviewer generates an incident report.
- ***Manar Saad Salih Al-Taie [Man02]*** developed a type of malicious code called Remote Access Trojon (RAT) works under windows environments. RAT provides an access to remote computer: some are used for monitoring purposes (such as capture desktop).
- ***Spy Arsenal [Spy03]*** Personal Desktop Spy (version 2.00) software for monitoring the activity of users on a PC by automatic capturing of desktop/active application screenshots and saving them to a specified directory on the hard drive.

1.3 Aim Of Thesis

The aim of this thesis is to develop a Target Monitoring TM to protect any computer from misuse, or avoid any attempt of misuse. This system allows for monitoring of LANs.

The thesis is constructed to support two types of monitoring:

1) Online Monitoring(ON-MON), which provides the ability of spying function.The constructed ON-MON designed depending on the concept of client/server system where the server resides on the remote computer, and can be controlled by the client

side to perform the monitoring functions. This system can monitor **the remote computer screen, mouse motion, and keystrokes**. In addition to sending warning messages to the remote computer user. The administrator can force the remote user to logoff or shutdown windows when the user performs suspicious actions on his computer.

2) Offline Monitoring (OFF-MON), which provides the ability of monitoring keystrokes only.

1.4 Thesis Outline

- **Chapter Two**

This chapter concerned with the definition of monitoring system, architecture of monitoring system, monitoring types depend on the time, and monitoring benefits. This chapter also introduces networks, TCP/IP model protocols, explanation of the windows Registry, and explain of the DirectX 7.

- **Chapter Three**

This chapter describes the design and implementation part of the system, it shows how the Target monitoring (TM) is designed and implementation, the explanation of each function and how each function is implemented is also expressed.

- **Chapter Four**

This chapter explores discussion and conclusions of this work, and the suggestions for future work.

Chapter Three

Development of Target Monitoring

3.1 Introduction

The project is concerned with developing a Target Monitoring (TM), that provides observation or watching remote computer and allows the administrator to perform set of functions on the remote computer. Target Monitoring(TM) composed of two integrated components: ***a security monitoring system (Agents)***, and ***a complete security assessment of critical mission control system.***

This monitoring system is constructed using Microsoft Visual Basic with DirectX 7 and is tested on LAN (using windows 2000, windows Millennium, and windows XP operating Systems).

3.2 Target Monitoring Architecture

Target Monitoring (TM) Architecture depends on the concept of client/server system; it uses the connection oriented of communication. TM mainly consists of two sides (figure 3.1):

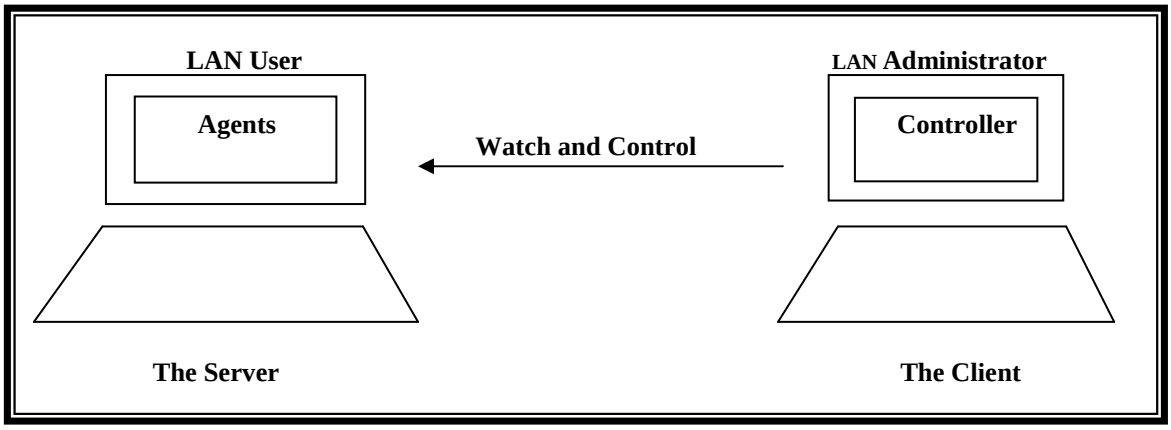


Figure 3.1 Target Monitoring Architecher

3.2.1 User side :(Server side, Remote Computer or Agents) this side is the remote computer at which set of agents reside on. These agents allows the administrator to observe users activities and perform specific

function provided by the TM on the remote computer(Server contains many Agents as shown in figure (3.2).

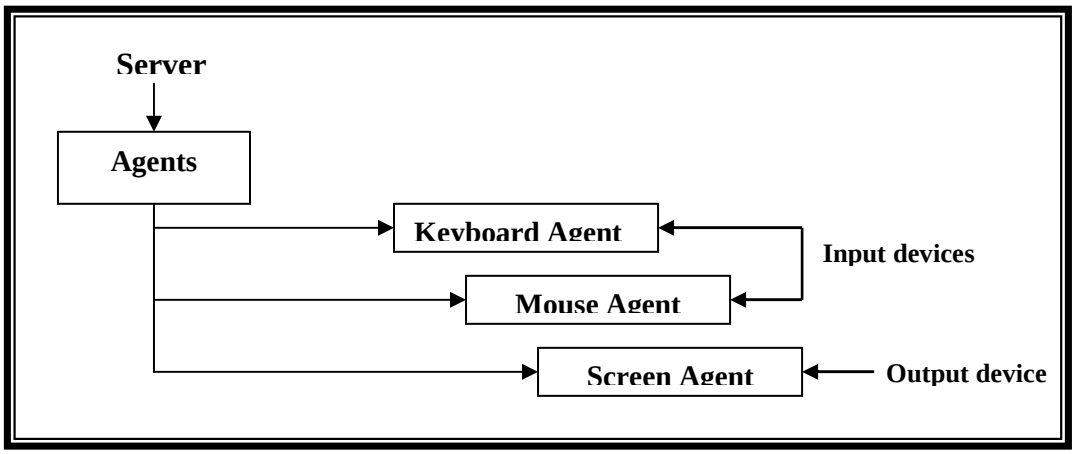


Figure 3.2 Agents on the Server

3.2.2 The Administrator Side:- (Client side, Local Computer or Controller) this side used by the administrator, it provides control over the Server side (i.e., the administrator can only perform the TM functions on the remote computer (client side)) Administrator has many activities as shown in figure (3.3).

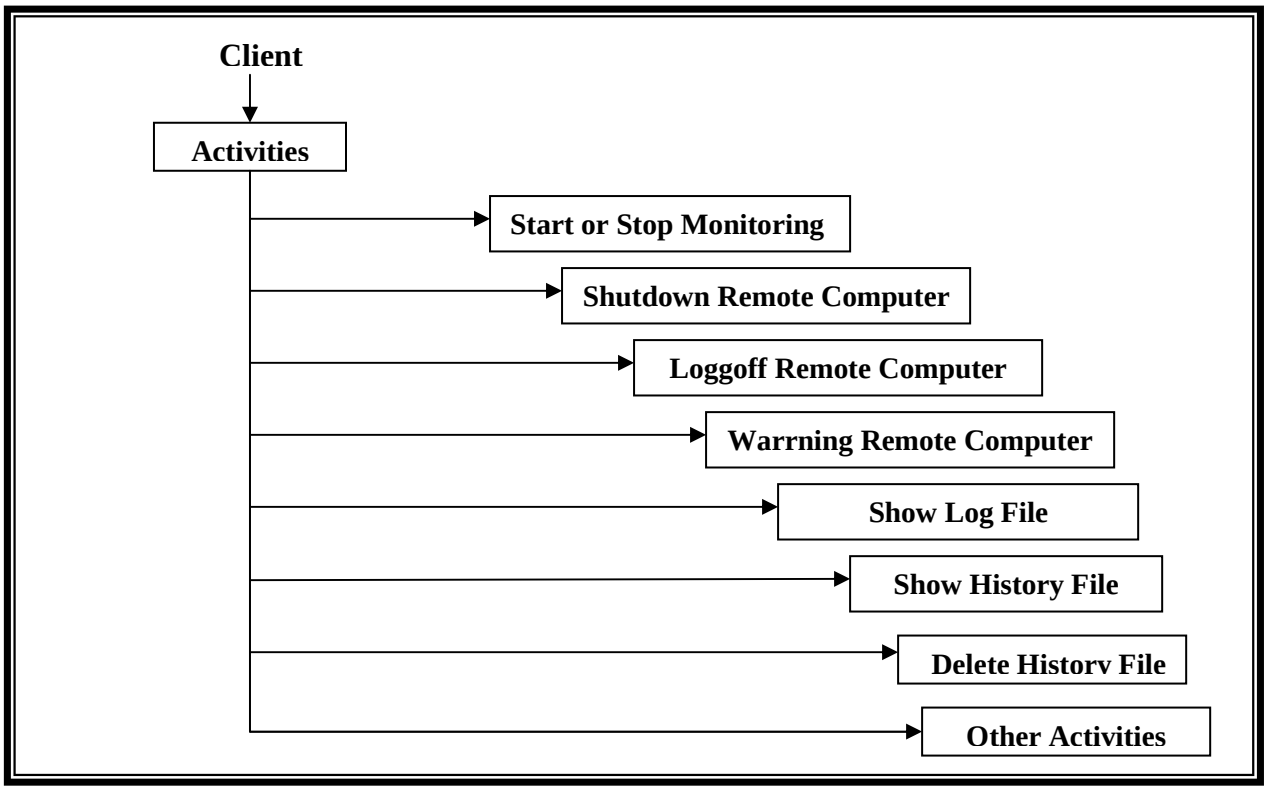


Figure 3.3 Client Activities

TM is designed to provide important functions that are used for monitoring purposes. Before exploring the design concept of TM, it is important to know how TM works (figure 3.4).

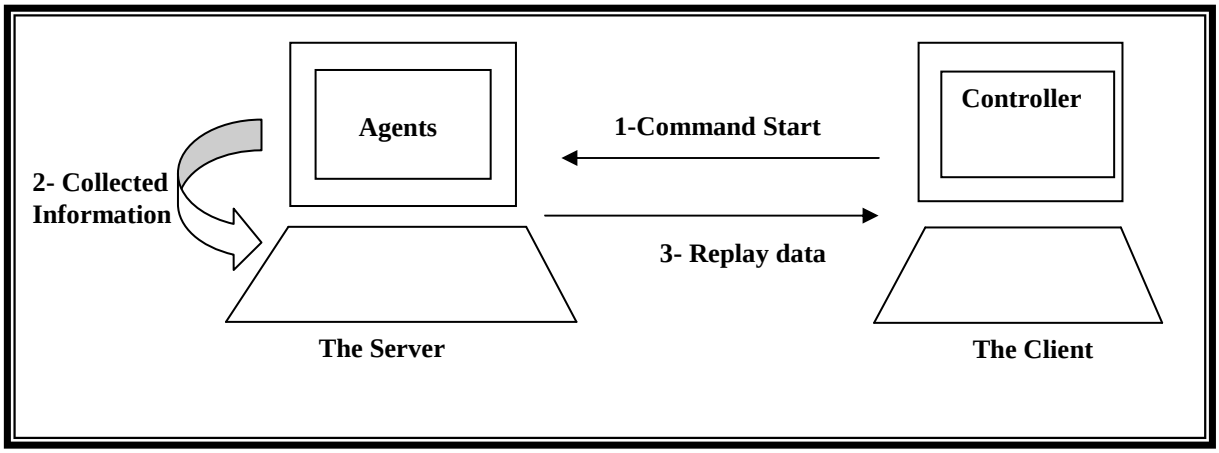


Figure 3.4 TM intercommunication

- The Server program registered on the registry and win.ini file. So that, when remote computer power on, the Server programs begins to execute.
- when windows next restarted (on the remote computer), the Server triggered and perform the following operation:-
    - Hide server from the Ctrl+Alt+Del menu to prevent the user from knowing that his computer being monitored.
    - Server start listening to the ports.
- The administrator runs on Client side, and then connect to the remote computer using the remote computer name.
- On the remote computer, the server receives the connection request and accepts that request.
- At this time the two computers are connected, and the administrator can perform any function that the TM is capable to perform on the remote computer (such as shutdown remote computer). At the same time, the agents on the server port start

working and send information (concerning keyboard, mouse, and displaying user) to the administrator.

3.3 TM Functions

As mentioned before, TM provides set of functions with which the client has control on the remote computer, figure (3.5) shows the flow control of TM along with its functions. The description of each step with its pseudo code and functions provided by the TM is illustrated below:

Let: ***Client:*** represents the windows socket control on the Client side.

Server: represents the windows socket control on the Server side.

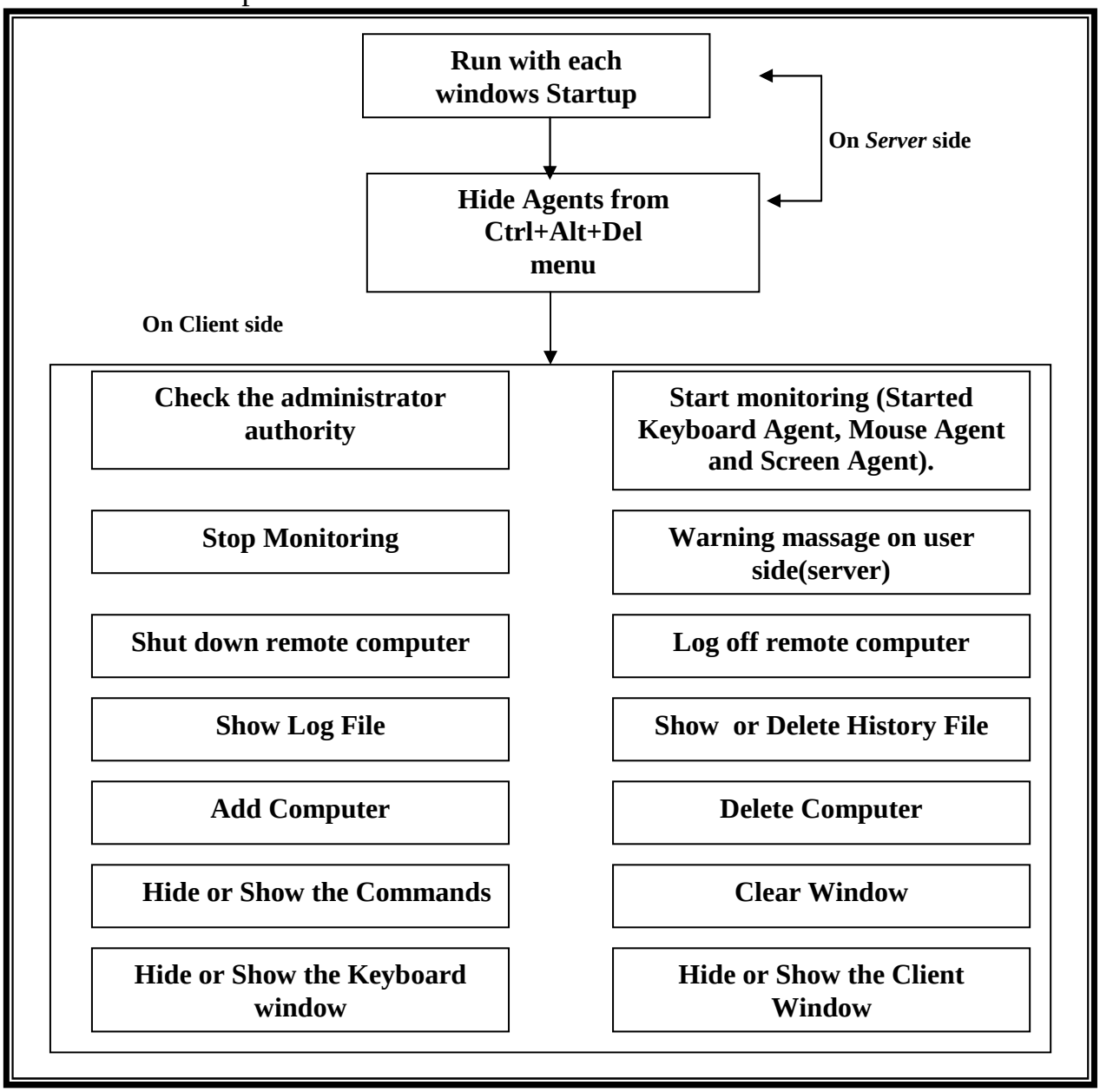


Figure 3.5 The Flow Control of TM

3.3.1 Run with each windows startup: this property allows the Server to be executed each time windows started. This is done by implementing the following steps:

Step1: Adding an entry in the windows registry key:

HKEY_LOCAL_MACHINE\software\Microsoft\CurrentVersion\Run

Step2: Adding entry in the win.ini file so that if one of the entries is deleted (from start up or win.ini), the program will continue to run with each windows startup.

3.3.2 Hide Agents from (Ctrl+Alt+Del) menu: As known, when pressing combination of keys (Ctrl+Alt+Del), a list of all current processes will appear, which allows the user to discover if there is any unusual process. Taking in consideration that any process in that list can be ended (stop running) by selecting it and press end-task button, therefore server agents must be hidden from that list. This feature is implemented using the following windows API functions :

GetCurrentProcessId ( ): this function returns the process identifier of the calling process.

RegisterServiceProcess ( ): this function registers or unregisters a service process (hide process from the Ctrl+Alt+Del menu).

3.3.3 Agents on the server side agents on the server side consist of the three agents. The ***Keyboard Monitoring Agent*** that monitors the keyboard, the ***Mouse Monitoring Agent*** monitors Mouse, and the ***Screen Monitoring Agent*** monitors the screen. These agents are described by the following:

1. Keyboard Monitoring Agent

This function provides for the Client the ability of observation or monitoring keyboard of the remote computer by ***Keyboard monitoring Agent.*** The Client sends a start monitoring command to the *Server*. The *Server* start monitoring (i.e., *Server* capture any Key pressed and send it to the client, character by character), *Server* check if the client sends the finish command; in this case, the *Server* stops monitoring (see figure 3.6). Keyboard monitoring agent at *Server* is implemented with the aid of DirectX7 functions, (the Algorithem 3.1 the keyboard monitoring agent).

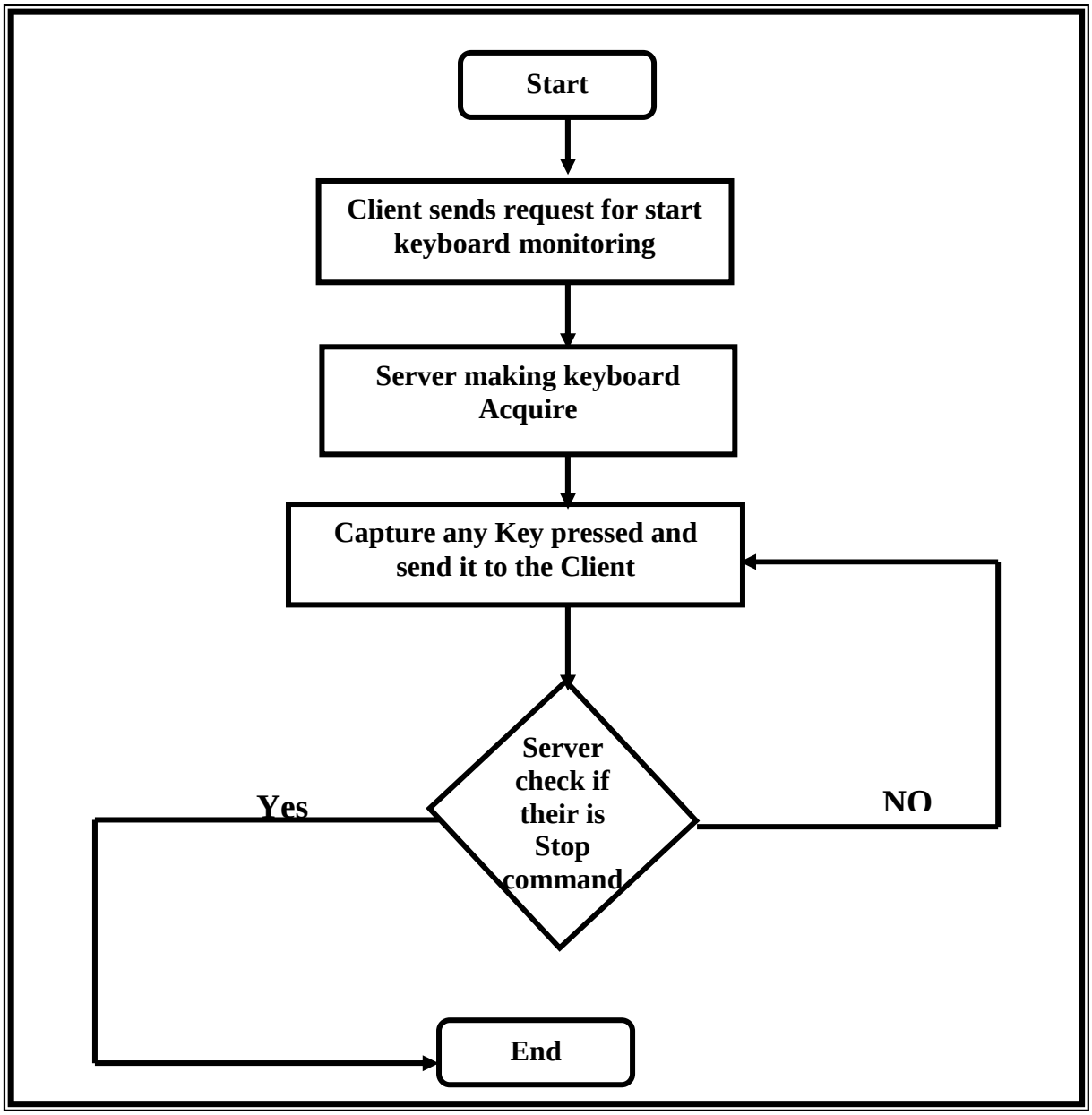


Figure 3.6 Flow control keyboard monitoring agent

Algorithem (3.1) Keyboard Monitoring Agent

Method: this algorithm is used when the administrator wants to Monitor the keyboard device.

Input:

S is a string that represented a **Start requested** sended by the client side.

Output:

Any key pressed on the server keyboard device.

Algorithm Steps:

Initialize

//create the direct input keyboard device (see algorithem 3.2)

If (Start requested =S) then IdID.Acquire

Else

wait

//check if the S is start requested arrival to server side then direct input interface makes aquire before retrieving data from it. The Acquire method accepts no parameters.

/* This procedure is recall if the event "Key press " making.*/

if (Keypress="ON) then

IdID.GetDeviceStateKeyboard DIKeyboardSTATE

//Retrieve data from the device Keyboard acquired by calling *GetDeviceState* function. The **GetDeviceState** method accepts two parameters: the size of a buffer to be filled with device state data, and a pointer to that buffer. For keyboards, always declare a buffer of 256 unsigned bytes

Server.SendData j

// j is key pressed that Send to the client side.

If stop requested then keyboard device must unacquire.

// see algorithem 3.8

Exit

Algorithem (3.2) Initialize for monitoring keyboard.

Method: this algorithm is used to create directinput for Keyboard device.

Input:

None.

Output:

The variable "IdID" is representing as interface for using keyboard device.

Algorithm Steps:

Note: The variable Dx is an instance of the DirectX7 object.

//To using keyboard input, first must create an instance of a DirectInput object. Then use the built-in function ***DirectInputCreate*** method to create an instance of an ***IdI*** interface**.** The **IdI** interface methods are used to manipulate the device, set its behavior, and retrieve data.

Step 1: IdI = DX.DirectInputCreate ()

// Create IdI is a variable representing directInput7 interface.

Step 2: IdID= IdI.CreateDevice ("GUID_SysKeyboard")

//determine the type of the input device by using *CreateDevice* function. The parameter is the GUID for the device being created. Since the system keyboard will be used, then pass the *GUID_SysKeyboard*.

Step 3: IdID.SetCommonDataFormate DIFORMAT_KEYBOARD

// specifying the data format provided for keyboard device by calling the *SetCommonDataFormat* function (this step identifying the data format to use for the device).

Step 4: IdID.SetCooperativeLevel hwndMain, DISCL_BACKGROUND Or DISCL_NONEXCLUSIVE

// The cooperative level of a keyboard determines how the input is shared with other applications and with the Windows system set it by using the ***SetCooperativeLevel*** function. The parameters are the handle to the top-level window associated with the device (generally the application window) and one or more flags.

2. Mouse Monitoring Agent

This function provides for the Client the ability of observation or monitoring Mouse of the remote computer by ***Mouse Monitoring Agent.*** The Client sends a start monitoring command to the *Server*. The *Server* starts monitoring (i.e.,*Server* capture mouse event **"mouse position and buttons** "and send to the client), *Server* checks if the client sends the finish command, the *Server* finishs monitoring (see figure 3.7). Mouse Monitoring Agent at *Server* side is implemented using DirectX7 functions (the Algorithm 3.3 the mouse monitoring agent).

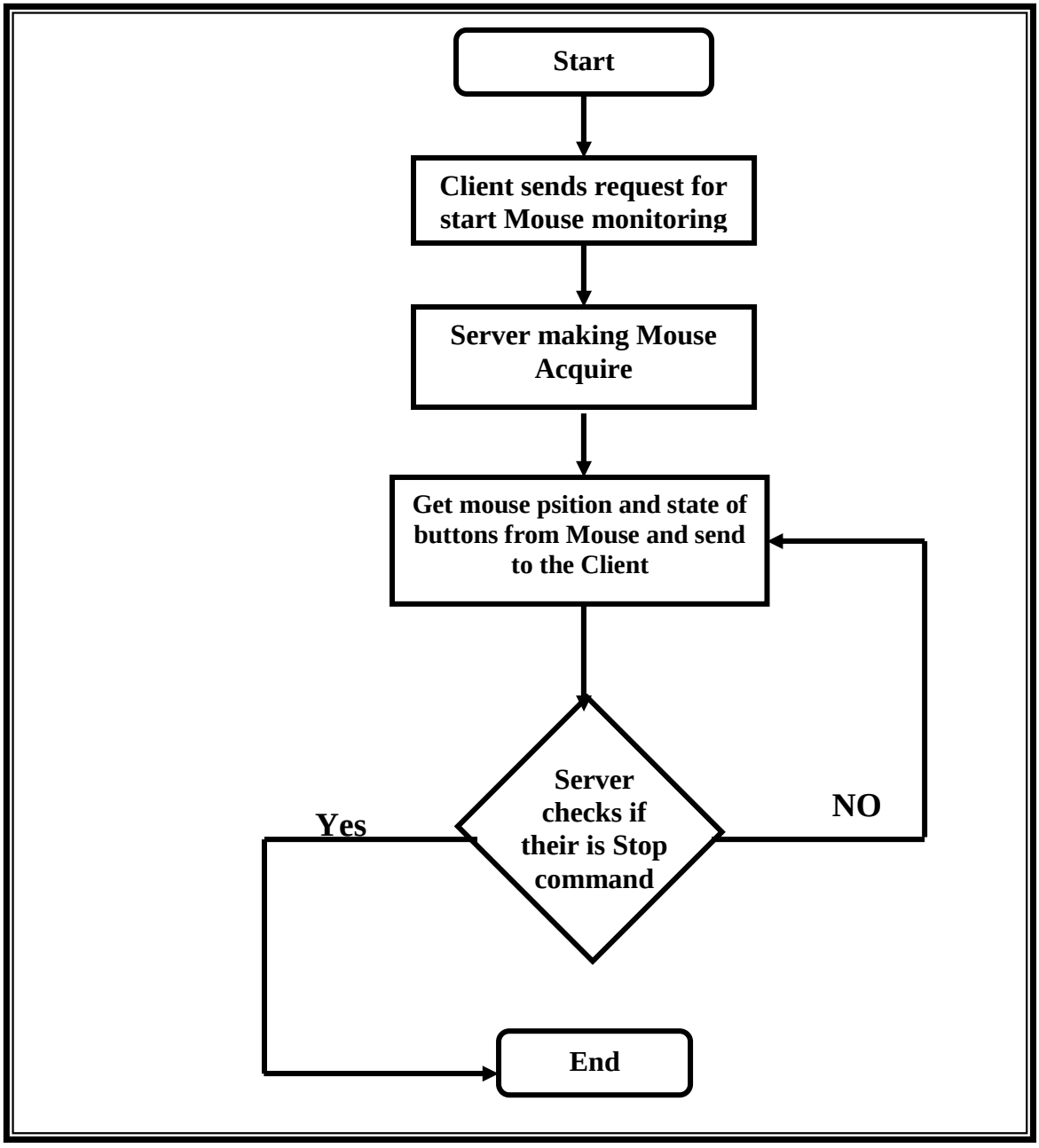


Figure 3.7 Flow control of mouse monitoring agent

Algorithem (3.3) Mouse Monitoring Agent

Method: this algorithm is used when the administrator wants to Monitoring of the mouse device.

Input:

S is a string that represented a **Start requested** sended by the client side

Output:

Positions and buttons of the server's mouse.

Algorithm Steps:

Initialize

//create the direct input mouse device (**algorithem 3.3**)

If (Start requested =S) then IdID.Acquire

Else

wait

// check if the S is a start requested that arrival to server side then direct input interface makes aquire before retrieving data from it. The Acquire method accepts no parameters.

/* This procedure is recall if the event "Mouse Movement " or "Mouse Click" making.*/

if (Mouse Movement ="ON") or (Mouse Click="ON") then

IdID. GetDeviceStateMouse DIMouseSTATE

//Retrieve data from the device Keyboard acquired by calling *GetDeviceState* function. The **GetDeviceState** method accepts two parameters: the size of a buffer to be filled with device state data, and a pointer to that buffer.

GetCursorPos(x , y)

// read the mouse position (x and y) by **GetCursorPos** function.

Server.SendData Pos_Mouse

// Pos_Mouse is the record contains mouse position and buttons Send the mouse positions and buttons to the client side

If Stop requested then **Mouse device must unacquire**.

//see algorithem 3.9.

Exit

Algorithem (3.4) Initialize for monitoring Mouse.

Method: this algorithm is used to create directinput for Mouse device.

Input:

None.

Output:

The variable "IdID" is representing as interface for using mouse device.

Algorithm Steps:

Note: The variable Dx is an instance of the DirectX7 object.

//To use mouse input, first, must create an instance of a DirectInput object. Then use the built-in function ***DirectInputCreate*** method to create an instance of an ***IDI*** interface**.** The **IDI** interface methods are used to manipulate the device, set its behavior, and retrieve data.

Step1: IDI = DX.DirectInputCreate ()

// Create IDirectInput7 interface.

Step 2: IdID= IDI.CreateDevice ("GUID_SysMouse")

//determine the type of the input device by using *CreateDevice* function. The parameter is the GUID for the device being created. Since the system mouse will be used, then pass the *GUID_SysMouse*.

Step 3: IdID.SetCommonDataFormate DIFORMAT_MOUSE

// specifying the data format provided for mouse device by calling the *SetCommonDataFormat* function (this step identifying the data format to use for the device).

Step 4: IdID.SetCooperativeLevel hwndMain,

DISCL_BACKGROUND Or DISCL_NONEXCLUSIVE

// The cooperative level of a mouse determines how the input is shared with other applications and with the Windows system set it by using the ***SetCooperativeLevel*** function. The parameters are the handle to the top-level window associated with the device (generally the application window) and one or more flags.

3. Screen Monitoring Agent

This function provides for the Client the ability of observation or monitoring Screen of the remote computer by the Screen Monitoring Agent. The Client sends a start monitoring command to the *Server*. The *Server* starts monitoring (Server Capture Screen and send to the client), *Server* checks if the client sends the finish command, the *Server* finishs

monitoring (see figure 3.8). Screen monitoring agent at *Server* side is implemented using DirectX7 functions (the Algorithm 3.5 the Screen monitoring agent).

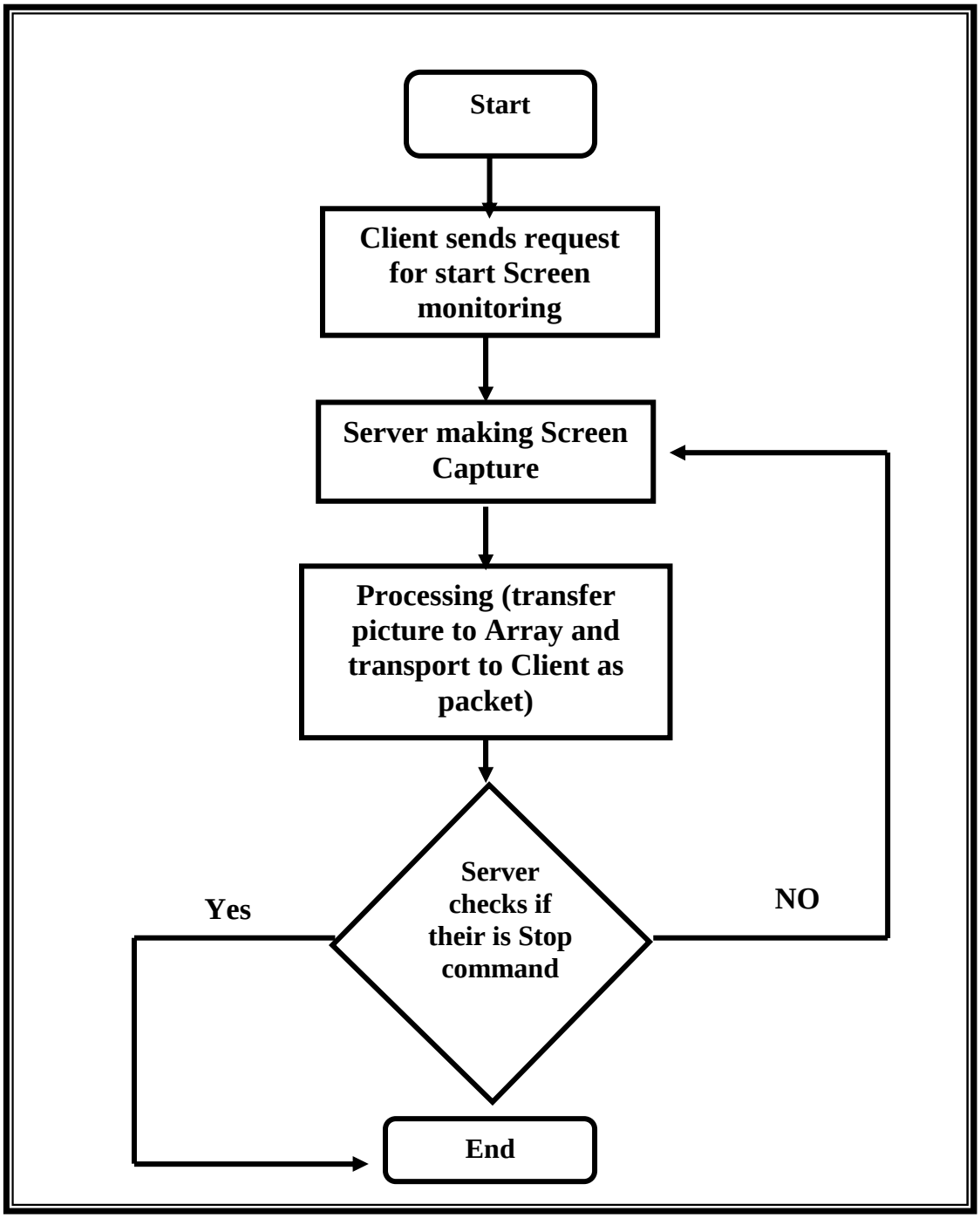


Figure 3.8 Flow control of screen monitoring agent

Algorithem (3.5) Screen Monitoring Agent

Method: this algorithm is used when the administrator wants to Monitor the screen device.

Input:

S is a string that represented a **Start requested** sended by the client side

Output:

displaying window of server's screen.

Algorithm Steps:

Initialize

//create the direct draw (see algorithem 3.6)

If (Start requested =S) then Goto Lable1

Else

wait

//check if the start requested "S" is arrival to server side then makes capture screen

/* This procedure is recall (every 2 Second) if the event "Timer1" */

Label1: Timer1.enable=true

// making the timer is set

if Timer1 then

Backbuffer.Blt (R2, Primary, R1, DDBLT_WAIT)

// Blitting is a process of transfering blocks of images data from one surface to another.**Blit** function is called on the backbuffer surface and receive the Primary surface as parameter.

Backbuffer.GetLockedArray (P)

// Transfer the image (on backbuffer surface) to array, by using the **GetLockedArray** function.

Server.SendData P

//P is the array of pixel send the array of image to the client side

If stop requested then Deallocate all surfaces.

//call algorithem 3.10.

Exit

Algorithem (3.6) Initialize for Monitoring Screen

Method: this algorithm is used to create DirectDraw for screen device.

Input:

None

Output:

The variable "IdD" is representing as instance for using screen device.

Algorithm Steps:

Note: the variable DX is an instance of the DirectX7 object.

// To using screen draw, first must create an instance of a DirectDraw object. Then use the built-in function DirectDrawCreate method to create an instance of an IdD interface. The IdD interface methods are used to manipulate the device, set its behavior, and retrive data.

Step1: IdD=DX.DirectDrawCreate()

// create IdD represented an interface of DirectDraw object.

Step2: IdD.SetCooperativeLevel (Me.hWnd,DDSCL_FULSCREEN or DDSCL_EXCLUSIVE)

// the cooperative level of a screen determines how the input is shared with other applications and with the windows system set it by using the SetCooperativeLevel function. The parameters are the handle to the top-level window associated with the device (generally the application window) and one or more flags, these flags give complete control over the display device, and no other application will be able to share it.

Step3: Primary=IdD.CreateSurface(d_DDSURFACEDESC2)

// tomanipulate images in direct draw, must create objects known as surfaces. A Surface provides a rectangual region of memory that can be used for image storage, creates the surfaces by using CreateSurface function. The parameter is a d_ DDSURFACEDESC2 that is a pointer to the DDSURFACEDESC2 stracture that contains a descripion of the surface to be created.

Step 4: backbuffer= primary.GetAttachedSurface(caps)

//retrieve a poiter to the backbuffer surface by using the GetAttachedSurface function.

3.4 Monitoring Activities on Client Side

Client side has control on the server side at which set of functions are performed such as:

3.4.1 Check for administrator authority: this function used to check administrator authority before open the main window of monitoring program. Algorithm 3.7 used to check thr administrator authority.

Algorithm 3.7 (Check administrator)

Method: this algorithm used to check administrator authorization.

Input:

Pass: the password of the adminstator.

ID: the name of the administrator.

Output:

Either true or false.

Algorithm steps:

Step 1: K=0 /* K represents number of tries when illegal user ID and /or Password entered*/

Step2: Input pass and ID.

Step3: check the ID and password

```
If(ID=administator)and(Pass=password)then
    Return(true)
Else
   Inc(K)
   If K<3 then goto step2
```

Step4: Return(false)

Step5: Exit

3.4.2 Start monitoring (Started monitoring for Keyboard, Mouse and Screen. this function provides for the Client the ability of start monitoring the remote computer. The Client sends a start command to the Server (to Keyboard, mouse, and screen agents). The *Server* (keyboard, mouse, and screen agents) starts monitoring using algorithm (3.1), algorithm (3.3), and algorithm (3.5) respectly.

3.4.3 - Stop Monitoring

This function is provided for the Client to give his the ability to stop monitoring of the remote computer. The Client sends a stop command to the *Server* (to Keyboard Agent, mouse Agent, and screen Agent). The *Server* (keyboard agent, mouse Agent, and screen Agent) stops monitoring.

All monitoring Agents at *Server* side use DirectX7 functions to stop monitoring (the algorithem 3.8 to stop keyboard monitoring agent, the algorithem 3.9 to stop Mouse monitoring agent, and the algorithem 3.10 to stop Screen monitoring agent).

Algorithm 3.8 (Stop Keyboard Monitoring Agent)

Method: this algorithm is used when the administrator wants to stop keyboard Monitoring.

Input:

C1 is a string of Stop requested that send by the client side.

Output:

None.

Algorithm Steps:

Step 1: If (Stop requested =C1) then Goto step 2.

Else

wait

Continue

//check if the stop requested is C1 that arrival to server side then makes keyboard monitoring stoped.

Step 2: IdID.Unacquire

// Unacquire all DirectInput devices

Step 3: IdID.Release

// Release all DirectInput devices

Step 4: IdI.Release

// Release the DirectInput object

Step 5: End

Algorithm 3.9 (Stop Mouse Monitoring Agent)

Method: this algorithm is used when the administrator wants to stop mouse Monitoring.

Input:

C2 is a string of Stop requested that send by the client side.

Output:

None.

Algorithm Steps:

Step 1: **If (Stop requested =C2) then Goto step 2.**
Else
wait
//check if the stop requested is C2 that arrival to server side then makes keyboard monitoring stoped.

Step 2: IdID.Unacquire

// Unacquire all DirectInput devices

Step 3: IdID.Release

// Release all DirectInput devices

Step 4: IdI.Release

// Release the DirectInput object

Step 5: End

Algorithm 3.10 (Stop Screen Monitoring Agent)

Method: this algorithm is used when the administrator wants to stop screen Monitoring.

Input:

C3 is a string of Stop requested that send by the client side.

Output:

None.

Algorithm Steps:

Step 1: If (Stop requested =C3) then Goto step 2.
Else
wait
//check if the stop requested "C3" is arrival to server side then makes screen monitoring stoped.

Step 2: IdD.RestoreAllSurfaces

// Deallocate all surfaces by using **RestoreAllSurfaces** function.

Step 3: End

3.4.4 Shutdown Remote computer

The administrator can force the remote computer to perform Shutdown operation. This can be done by sending Shutdown command to the *Server* to Shut down remote computer, the *Server* then recognize that command and call the windows API function

Algorithm 3.11 (Shutdown remote computer)

Method: this algorithm is used when the administrator wants to shutdown remmote computer.

Input:

K is a string of Shutdown requested that is send by the client side.

Output:

None.

Algorithm Steps:

Continue

Step 1: If (Shutdown requested =K) then Goto step 2.
Else
Wait
// check if the shutdown requested is K that arrival to server side then makes Shutdown for server computer.
Step 2: ExitWindowsEx (EWX_SHUTDOWN, 0)
// makes shutdown computer.
Step 3: End

3.4.5 Logoff Remote Computer

The administrator can force the remote computer to perform logoff operation. This can be done by sending logoff command to the *Server* to logoff remote computer, the *Server* then recognize that command and call the windows API function.

Algorithm 3.12 (Logoff remote computer)

Method: this algorithm is used when the administrator wants to logoff remmote computer.

Input:

F is a string of Logoff requested that is send by the client side

Output:

None.

Algorithm Steps:

Step 1: If (Logoff requested =F) then Goto step 2.
Else
Wait
// check if the logoff requested is F that is arrival to server side then makes logoff for server computer.

Step 2: ExitWindowsEx (EWX_FORCE, 0)
// makes logoff computer.
Step 3: End

3.4.6- Display warning massage to remote computer

This function provides for the Client the ability of displaying a warning massage to the remote computer. The Client sends a display warning message command to the *Server*. The *Server* displays a warning message to user. Displaying warning message agent at *Server* side is implemented using DirectX7 functions (algorithm 3.13 of the Display warning message).

Algorithem (3.13) Displaying Warrning Massage

Method: this algorithm is used when the administrator wants to Display Warrning Massage to the server side.

Input:

S1 is a string of Start requested that is send by the client side.

Output:

displaying Warrning Massage.

Algorithm Steps:

Step 1: Initialize
//create the direct draw (see algorithem 3.6)

Step 2: If (Start requested =S1) then Goto step 3
Else
wait
//check if the start requested "S1" is arrival to server side then makes capture screen.

Step 3: backbuffer. DrawText (10, 10, "You enter worring Area; Please try to Exit.", False)
//Writes some text on the back buffer surface by using standard Windows GDI functions (**DrawText** function).

Step 4: primary.Flip Nothing, DDFLIP_WAIT
// **Flip** function used to flip the ***back buffer*** to theprimary surface(the front and back surfaces are exchanged).

Step 5: If time_interval<>0 then goto (step 4)
// time_interval is a constant that equal 25 second.

Step 6: IdD.RestoreAllSurfaces
// Deallocate all surfaces by using **RestoeAllAurfaces** function

Step 7: Exit

3.4.7- Show History File:

This function allow administrator to see some information about the monitored computers.

Statistical file contains the following information: **Local IP Address**, **Remote IP Address, Date of monitoring, time of monitoring, state (Start or Stop monitoring), and Type of monitoring (online monitoring or offline monitoring).**

This information obtained and recorded by the Administrator (controller). When administrator who wants to monitor any server (i.e, the client send the start monitoring request to the server), the administartor records all information about this monitoring in statistical file and also record these informatiom when administrator stop monitoring. Table (3.1) shows an example of the information that is stored in statistical file:

Table 3.1: Statistical-File Example.

Source IP	Dest. IP	Date	Time	State	Type
192.168.0.2	192.168.0.5	31/01/2005	12:02:09م	Start	Online
192.168.0.2	192.168.0.5	31/01/2005	12:20:10م	Stop	Online
192.168.0.2	192.168.0.3	4/4/2005	1:20:00م	Stop	Offline

In the above table, the first row tells the administrator that the client with the Local IP address 192.168.0.2 monitors the remote computer with the IP address 192.168.0.5 in date 31/01/2005 at time 12:02:09 and the type of monitoring is online.

3.4.8 Clear History File: Clear History file function is used if the administrator wants to delete statistical file.

3.4.9 Other Activities

Other activities represent the functions in the menu bar of monitoring system at the administrator side. These functions are:

3.4.9.1 Computers: which provides set of options that allows the administrator to add or delete a computer name from list of the remote computer names.

1- AddComputer:

Add Computer function aid administrator to add new computer name. When press on this command; requires the **computer name** and **IP Address** and check if computer name found in the computer names file this computer cannot added, but if not found the required IP address and check if IP address found in the IP address file. If found can not add this computer but if not found then add the computer name to combo box (add the computer name to combo box and to the computer names file) and add the IP address to the IP address file,see figure 3.9.

2-Delete Computer:

Press this command, after selecting the computer name by the combo box. When pressing this command, gets the index of the IP Address of this computer name and deleted this computer name from combo box and delete the IP Address of this computer from the IP address file,see figure 3.10.

3. Exit:

This command end (Stop) running the program (on Client side).

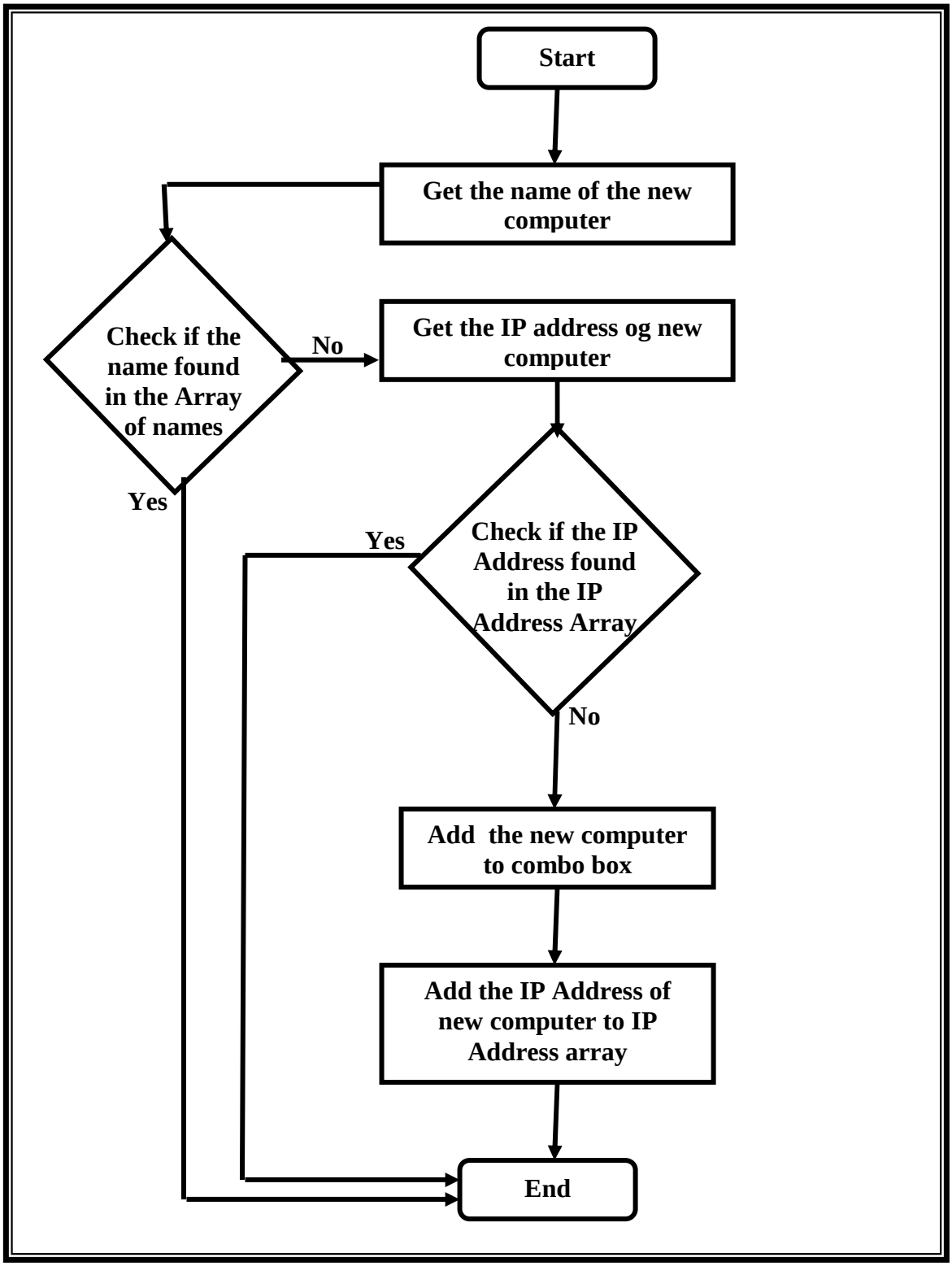


Figure 3.9 Add the name of new computer to file of computer names

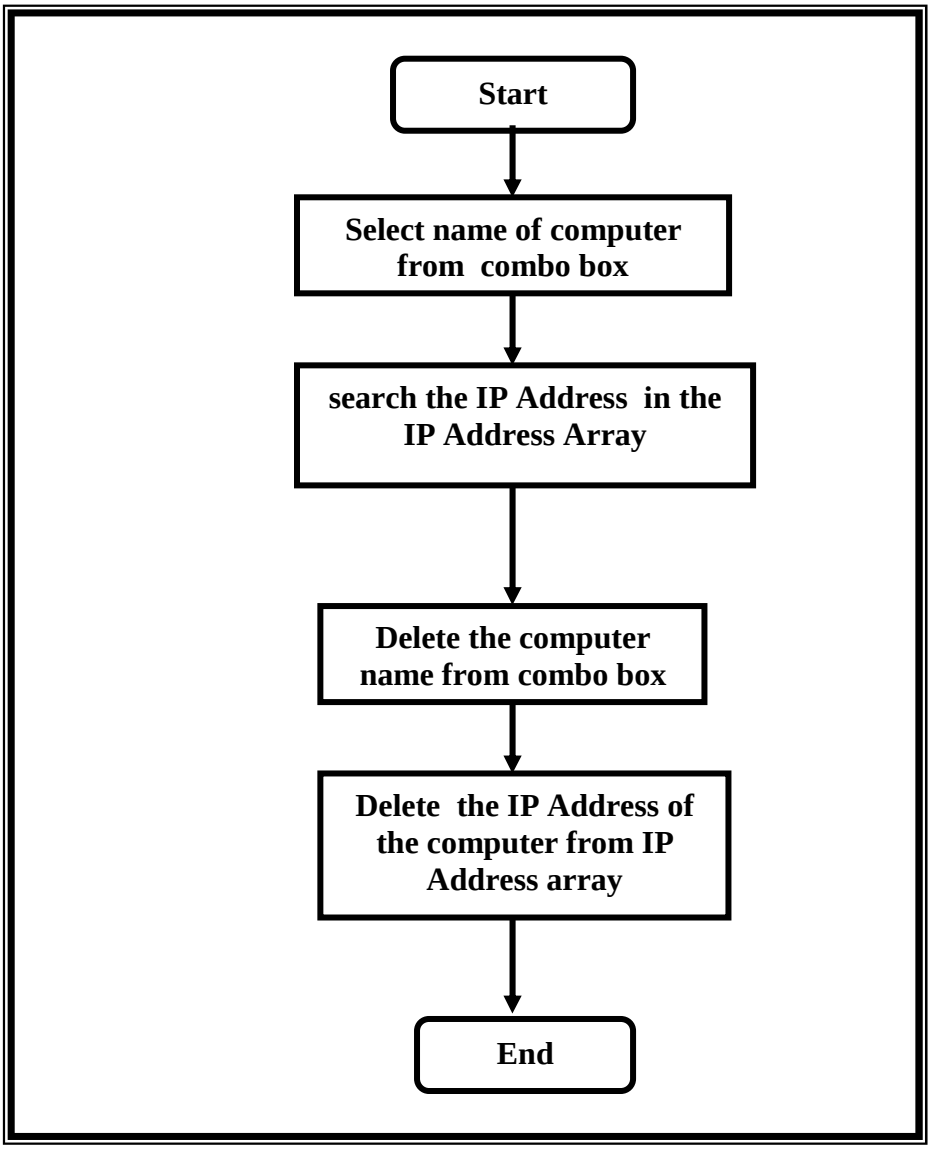


Figure 3.10 Delete computer name from file of computer names

3.4.9.2- Clear: which provides set of options that allows the administrator to **clear window** or **clear history file**:

- **Clear window:** this function is to clear the victim computer window, keyboard information window and the computer name text.

3.4.9.3- System: which provides set of options that allows the administrator to select online or offline system.

3.4.9.4- View: this menu contains three function:

- **Hide/Show Client Window:** this function gave the ability to hide or show the client window.
- **Hide/Show keyboard Information Window:** this function gave the ability to hide or show the Keyboard information window.
- **Hide/Show Commands Window:** this function gave the ability to hide or show the commands window.

3.4.9.5- Help: this menu contains two functions:

- **Help about online monitoring:** this function appears the massage box contain short informtion about the online monitoring system**.**
- **Help About offline monitoring:** this function appears the massage box contain short informtion about the offline monitoring system**.**

3.5 How to use the TM System

The TM system has 4-forms to enable the administrator to monitors the remote computer. These forms are:

1- ***Password form***: used for check authenticity of the administrator.
2- ***Monitoring Form***: used to show captured screen and mouse movment of the remote computer.
3- ***Keyboard information Form:*** used to display any key pressed by the user on the remote computer.
4- ***Offline Form:*** used to displaying log file (for more information see Appendix H, which is allustrates TM user interface).

CHAPTER TWO
Concepts of Computer Monitoring

2.1 Introduction

System provids valuable (not normally available) information with standard activity logs, could assist in identifying, and possibly controlling the damage inflicted upon the system, is called “Monitoring system” [Fch00].

The monitoring can be considered as a one of the security mechanisms since it is responsible for watching or observing users behavior. By monitoring changes, the administrator may be alerted before the system crashes, stops responding, or attacked, which allows the administrator more flexibility in mitigating the risks of its occurring [Kel02] .

Monitoring System seeks to answer a number of questions including the following [Nik02]:

• What happened on a remote computer?

• When did the events occur on a remote computer?

• In what order did the events occur on a remote computer?

• What was the cause of these events on a remote computer?

Traditionally, computer security monitoring (CSM) systems have been partition into two categories based on technique employed to determine if an alert should be generated from the audit data. These two categories are *Anomaly detection* and *Misuse detection.* There is a less frequently articulated third category called *Target monitoring* that can be considered a special subcategory of both anomaly and Misuse detection [Kup04].

Monitoring system performs network, systems and application monitoring with three steps. The first is to install Agents on remote computers to be monitored, then reports the data back to the Controller at

set intervals, and finally Controller perform custom checks and actions, if configured to do so[MIO02].

This chapter mainly consists of four parts: part one covers network system and the necessary concepts of the TCP/IP protocol. Part two concerns monitoring system, architecture of monitoring system, types and benefits of CSM. Part three and four concerned with software tools that could help in developing monitoring system under windows environment: DirectX and overview of windows registry.

2.2 Network System

A computer network is an interconnected system of computing devices that provides shared economical access to computer services. Networks are important since they provide several benefits such as ***Resource Sharing***, ***saving money***, and etc [Tan96].

Two types of transmission technology could be used with network system: ***Client/ Server Networks*** and ***Peer-to-peer Networks***.

2.2.1 The Client/Server Networks [Bkf98 ,Tan96]

The Client/server concept describes a computing system in which the actual processing needed to complete a particular task is divided between a centralized host computer (the *server*), and a user's individual workstation (the *Client*). The two are connected by cables (or infrared, radio, or microwaves) for communication. Figure 2.1 shows the logical architecture of client/server network.

Although client/server are both PCs with the same basic architecture, the client and server computers usually have very different hardware and software configurations. The primary function of each in

relation to the other can be stated in simple terms: the client requests services from the server, and the server responds by providing those services.

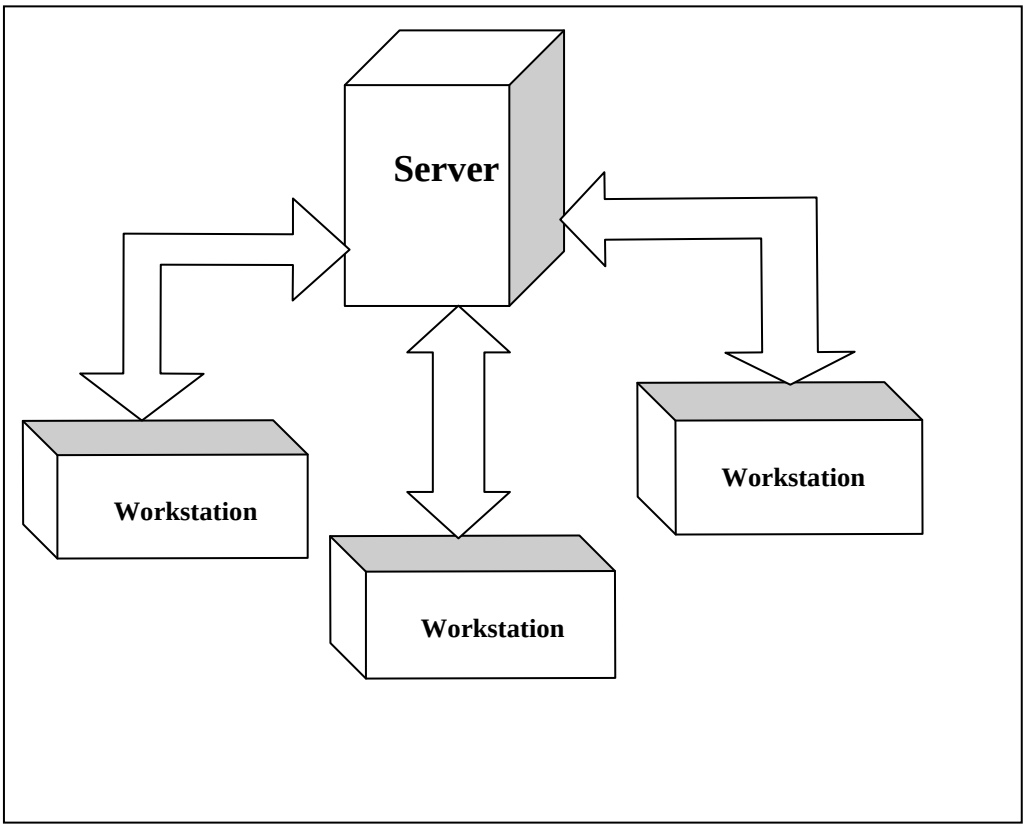


Figure 2.1 The logical architecture of client/server network.

2.2.2 The Peer-To-Peer Networks [Bkf98, Tan96]

While client/server networks are distinguished by how different the clients and servers are, each with a clearly defined role, peer-to peer networks are just the opposite.

There are still clients and servers in a peer-to-peer system, but generally, any fully functioning client may simultaneously act as a server. The resources of any computer on the network could be disk drives, files, applications, printers, modems, and so on. Figure 2.2 shows the logical architecture of peer-to-peer network.

Peer-to-peer networks are often comprised of only a few workstations, perhaps sharing printers and any other external devices as data file, stored on the hard disk of any workstation. The upper limit of the number of nodes that can function as both clients and servers on a peer-to-peer network with performance levels that remain reasonable is usually some where between 10 and 25 nodes.

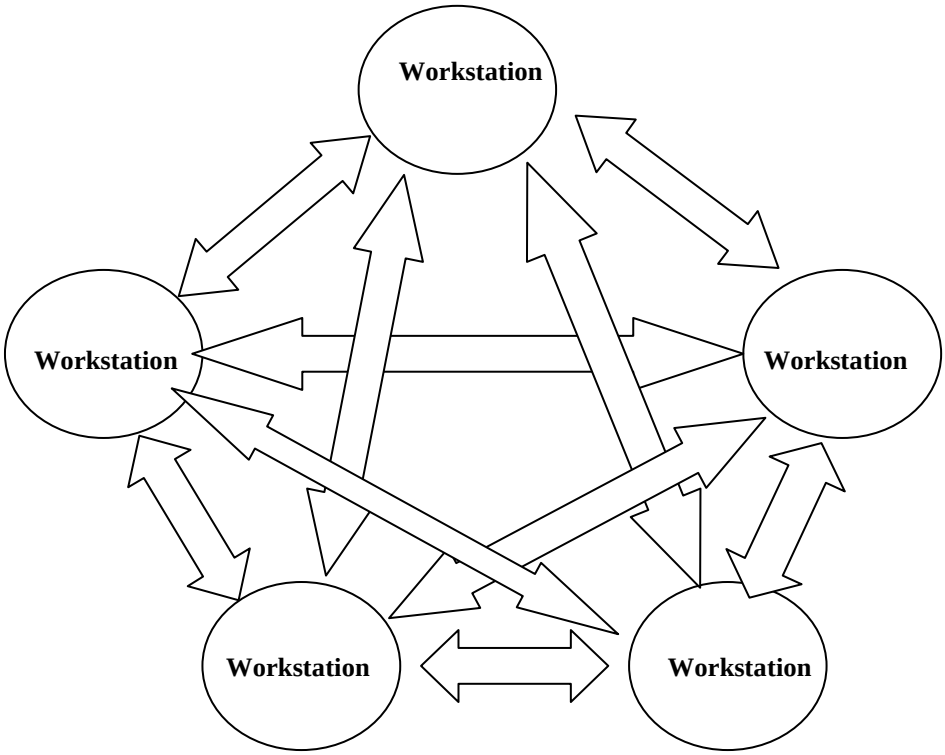


Figure 2.2 The logical architecture of peer-to-peer network.

2.3 Network Models

A network model (also referred to as *protocol suits*) reflects the design or architecture to accomplish communication between different systems. A network model usually consists of layers. Each layer of a model represents specific functionality.One of the most popular Network models is TCP/IP[Hel00].

The standard model of a layered network is the 7-layer International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference model. The entire OSI model is not implemented, where the most common layered set of protocols in use is the Transmission Control Protocol/Internet Protocol (TCP/IP) set of protocols. TCP/IP works in a vary similar manner to the OSI model in that it takes a layered approach to provide network services. Each layer in the TCP/IP model communicates with the layers above and below it in the same way that the layers in the OSI model do[Hel00]. (For more information about OSI see Appendix B)

The TCP/IP network model takes its name from two of its protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP). Figure 2.3 shows the five-layer represent of the TCP/IP model [Bks99, Tan96].

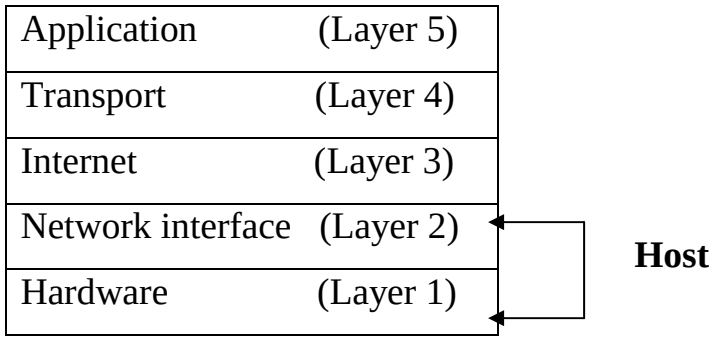


Figure 2.3 the TCP/IP Model

Each of the top three layers of the TCP/IP model actually consists of multiple protocols (as shown in figure 2.4). The following are the most popular protocols of ***Internet, Transport*** and ***Application*** Layers are illustrated below [Bks99, Tan96]: -

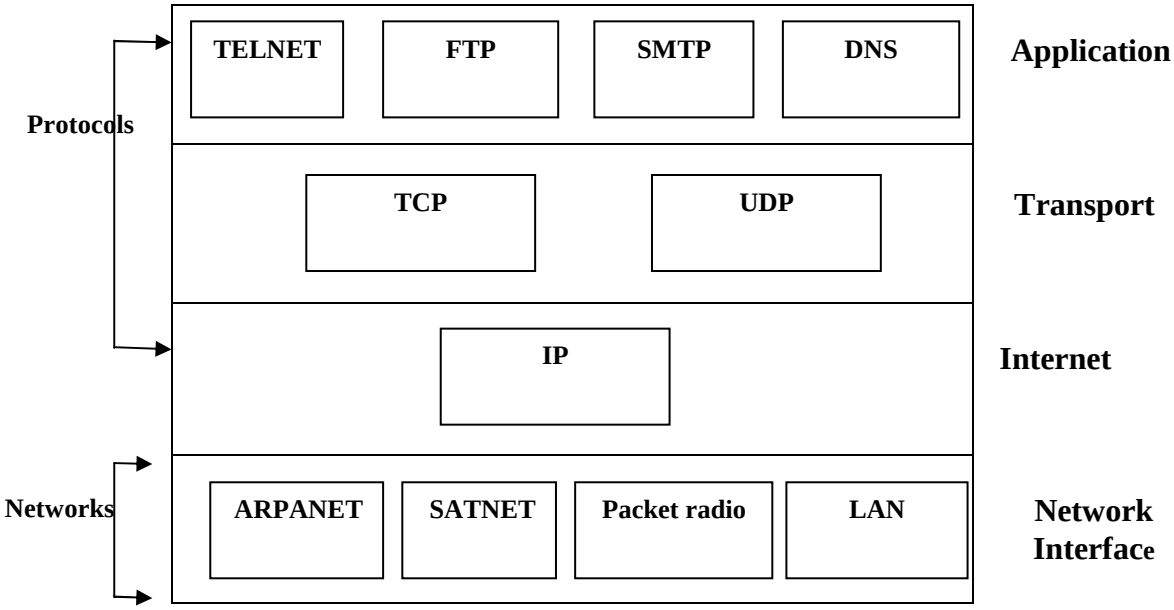


Figure 2.4 Protocols and networks in the TCP/IP model

2.3.1 IP (Internet Protocol)[Bks99, Tan96]

Internet Protocol resides into Internet layer. Its main tasks are addressing of information datagrams (Packets) between computers and managing the fragmentation process of these datagrams. The protocol has a formal definition of the layout of a datagram of information and the formation of a header composed of information about the datagram.

IP is responsible for the ***routing of a datagram***, ***determining where it will be sent*** and ***devising alternate routes in case of problems***.

Another important aspect of IP's purpose has to do with unreliable delivery of a datagram. Unreliable in the IP sense means that the delivery of the datagram is not guaranteed ،because it can get *delayed*, *misrouted*, or *mangled in the breakdown*. IP has nothing to do with flow control or reliability: there is no inherent capability to verify that a sent message is correctly received. IP does not have a checksum for the data contents of a datagram, only for the header information.

Part of the IP system defines how gateways manage datagrams, how and when they should produce error messages, and how to recover from problems that might arise.

2.3.2 Transport layer

At the ***transport layer***, the two common protocols are the Transmission Control Protocol (TCP) and the User Data-gram Protocol (UDP):

1- TCP (Transmission Control Protocol)[Bks99,Tan96]: The transmission control protocol provides a considerable number of services to the IP layer and the upper layers. Most importantly, it provides a connection-oriented protocol to the upper layers that enable an application to be sure that a datagram sent out over the network was received in its entirety. In this role, TCP acts as a *message-validation protocol providing reliable communications*. If a datagram is corrupted or lost, TCP usually handles the retransmission rather than the applications in the higher layers.

TCP manages the flow of datagrams from the higher layers to the IP layer, as well as incoming datagrams from the IP layer up to the higher-level protocols. TCP has to ensure that priorities and security are properly respected.

The isolation of all these services in a separate layer enables applications to be designed without regard to flow control or message reliability. Without the TCP layer, each application would have to implement the services themselves, which is a waste of resources. Because TCP is a connection-oriented protocol responsible for ensuring the transfer of a datagram from the source to destination machine (end-to-end communications), TCP must receive

communication messages from the destination machine to acknowledge receipt of the datagram.

2- **UDP (User Datagram Protocol)[Hel00]:** UDP is based upon the datagram method of transport for application for which an occasional lost packet is not considerd serious. Thus, UDP represents a connectionless, unreliable, best-effort transport service.

UDP does not issue acknowledgments to the originator upon receipt of data nor provide order to incoming datagrams. UDP does not provide error detection or capabilities to recover from the situation where packets become lost. Instead, it is up to the application to detect lost or missing data, typically by noting the absence of a response within a predefined period of time and then if appropriate retransmitting the data that was presumed to be lost.

2.3.3 Application Layer

TCP/IP model provides a set of protocols at ***application layer***. For example, Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), File transfer Protocol (FTP), and others. (For more information about application layer protocols see Appendix C).

2.4 Ports and Sockets [Bks99, Tan96]

All upper-layer applications that use TCP (or User Datagram Protocol "UDP") have a ***port*** number that identifies the application. In theory, port numbers can be assigned on individual machines however the administrator desires, but some conventions have been adopted to enable better communications between TCP implementations, which enables the port number to identify the type of service that one TCP system is requesting from another, (for example, SMTP service available on port 25; see Appendix (D)). A port number is 16-bit integer. (i.e. Ports are

numbered 1 through 65535). Port numbers less than 1025 are reserved for use only by the privileged root (super user) account on UNIX systems, port numbers between 1025 and 5000 are assigned by the TCP/IP for use by code inside the kernel, and port numbers above 5000 are available for use by ordinary (non-root) users.

Port numbers can be changed, although this can cause difficulties. Most systems maintain a file of port numbers and their corresponding service. Each communication circuit into and out of the TCP layer is uniquely identified by a combination of two numbers, which together are called a ***socket***. The socket is composed of the IP address of the machine and the port number used by the TCP software. Both the sending and receiving machines have sockets. Since the IP address is unique across the Internetwork, and the port numbers are unique to the individual machine, the socket numbers are also unique across the entire Internetwork. This enables a process to talk to another process across the network based entirely on the socket number.

TCP/IP network provides set of services (protocols), each of which has a specific port number. The commonly used port numbers are shown in Table 2.2.

Table 2.2 Frequently Used TCP Port Numbers

Port Number	***Process Name***	***Description***
21	FTP	File Transfer Protocol
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer Protocol

2.5 Computer Security Monitoring (CSM)

Traditionally, computer security monitoring (CSM) systems have been partitioned into two groups based on technique employed to determine if an alert should be generated from the audit data. These two

categorize are A*nomaly detection* and *Misuse detection.* There is a less frequently articulated third category called T*arget monitoring* that can be considered, it is a special subcategory of both anomaly and Misuse detection. Figure 2.5 illustrates CSM categories[Kup04].

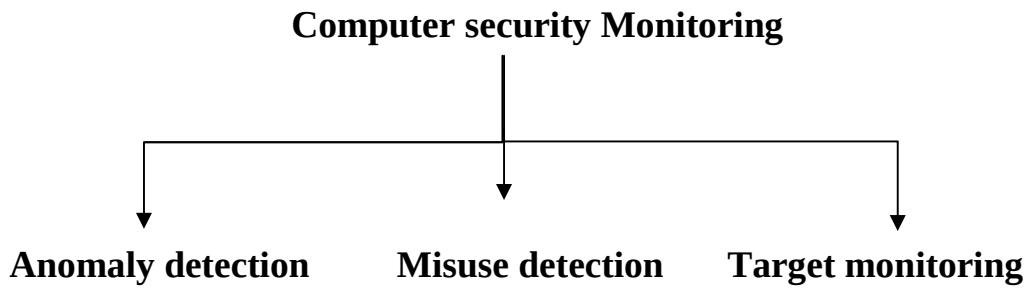


Figure 2.5 categorize of Computer Security Monitoring (CSM)

2.5.1 Anomaly Detection[CIS04]

Anomaly detection tools look for unusual activities or statistical anomalous behaviors. These tools assume that intrusions are rare and will appear unusual when compared with normal activities. In 1980, John P. Anderson improves the computer security auditing and surveillance capability of computer system, and partition threats (based on the authorization of using the computer facility as well as the authorization to use whatever data or programs that are of concern) against a computer system into two parts:

1) External penetration as either an outsider or unauthorized employee accessing a computer system.
2) Internal penetration was broken down into four subcategories:
    - *Masqueraders* who have proper credentials of some other user who is authorized to access data or program resources.

- *Legitimate users* who access data or programs that they are not normally authorized to access as part of their job.
- *Clandestine users* who are able to avoid or disable the audit system.
- *Misfeasance* where users abuse their legitimately granted access (see figure 2.6).

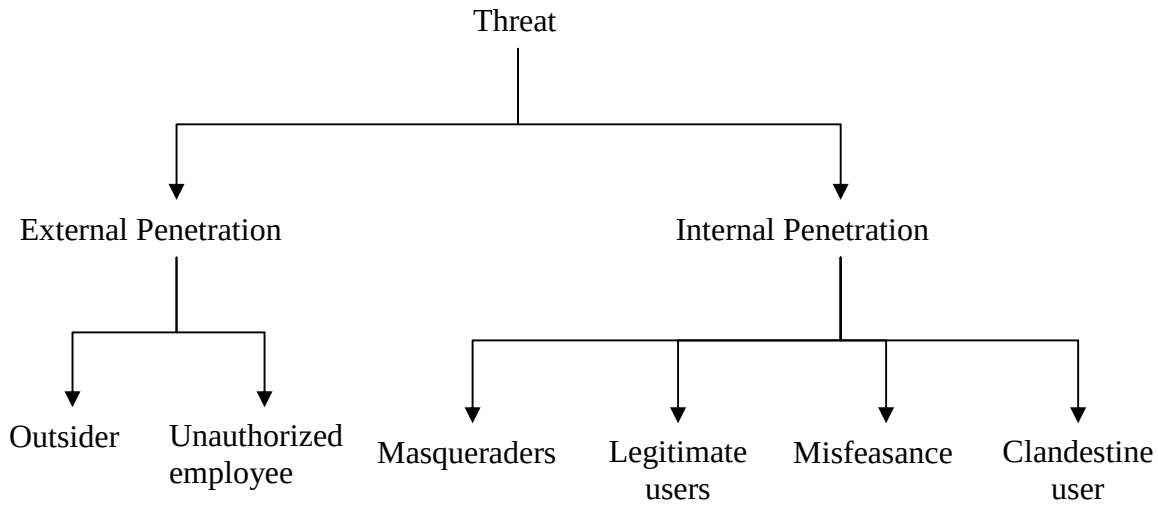


Figure 2.6 Anderson's general cases of threats against a computer system

All types of attacker's can be detected by functional description and task breakdown of both a surveillance subsystem and system traces component. Anomaly techniques using statistical to detect masqueraders by abnormal time of use, frequency of use, volume of data referenced, or from patterns of reference.

In 1987, Dorothy Denning published a model of a detection system. The model was designed to be independent of any particular computer system, application environment, set of vulnerabilities, or types of intrusions. The basis for this model was the hypothesis that "exploitation of a systems' vulnerabilities invoked abnormal use of the system; therefore, security violations could be detected from abnormal patterns of system usage".

2.5.2 Misuse Detection[Fch00]

Based on experience building and operating anomaly detection systems, some types of actions and activities were considered to always be worthy of notice. These are usually classified under the heading misuse. A misuse detection based system is generally based on a set of fixed rules used to determine if a particular audit event should trigger an alarm. There are three sub-categories of misuse detection frequently encountered in literatures: (see figure 2.7)

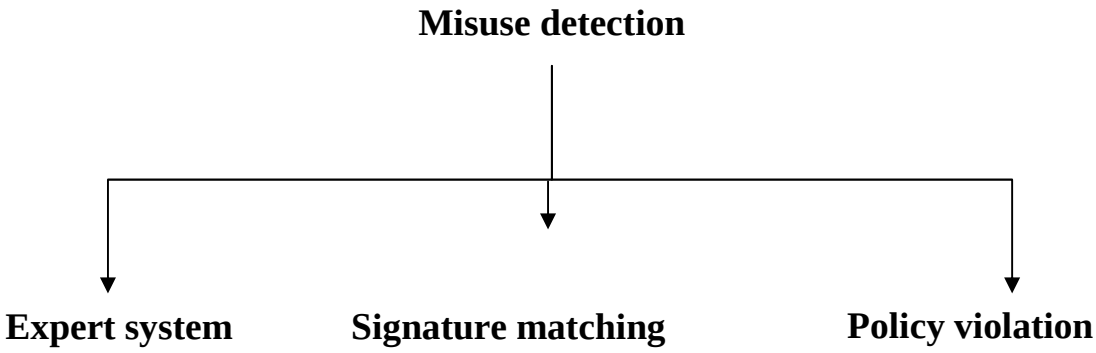


Figure 2.7 sub-categories of misuse detection

1- Expert system – A set of fixed rules that when violated will generate an alert. This system supposedly encodes the decision making process of a human expert in the subject domain into the decision making component of the computerized system. The Monitoring Intrusion Detection Expert System (MIDES) was one of the earliest intrusion detection systems based on an expert system analysis core. As these systems are based on modeling a human expert's behavior, they are limited by whatever information that a human expert is a ware of, including any misconceptions.

2- Signature matching – A process of looking for pre-specified, exact pattern matches to be contained within a data under analysis. This could be reviewed as a subset of an expert system; however,

signature matching systems are usually considered separately as they deal strictly with the pattern matching process and not any other decision making processes. Most virus scanning tools utilize a set of signatures as part of their detection similarly limited by expert knowledge and awareness, it has been noted that not all misuse can be represented by a simple pattern.

3- Policy violation – this system involves the encoding of a set of administrative policies into a machine interpretable form. Actions that are being monitored are compared against this set of policy rules to detect violations or deviations. Many systems that are designed to handle classify information utilize this type of system to determine when an operator is exceeding user authority. Some policies are difficult to model because of unstated assumption that may be held by those creating the policy but are not represented in the formal model.

2.5.3 Target Based Monitoring[Nik02]

There are certain events that can always be labeled as an instance of misuse. In some instances, certain actions can be enumerated that should never be performed by any user of a specified system. In other case, objects can be identified on the system that should never be accessed by any actions whatever. Taken together these objects or actions are the *targets* of system monitoring. Target based monitoring unlike anomaly or misuse monitoring.

Examples of target based monitoring include the following: -

- A machine connected to network that is not supposed to be accessed by either insiders or outsiders. This is also known as “Honeypot”.
- Files that are being monitored for access, modification, or execution. These are known as “tripwires”.

- Directories that are not supposed to be entered.

The described activities are certainly anomalous when compared to normal use. Most of them are objects that are not supposed to be accessed based on the expressed policy, and therefore are also misuses of the system when they occur. This means that target based monitoring can be used in instances where there is no formal policy, nor a known pattern of abuse.

2.6 Timeliness of Detection

One of the characteristics of CSM systems is the timeliness of the data processing, measured by the time between data generation to its correlation and the raising of alarms.

The timeliness specification should be applicable to both active and quiescent systems and apply regardless of whether security related events are occurring or not.

In the early publications on CSM, one of the unobtainable (though desired) goals was the ability to perform “real-time” detection. At the time, the overhead of data generation, network latency, and processing speed of the analysis engine limited the performance of the implemented systems. This led to many of them being run on a dedicated host with audit data collected from the systems of concern and delivered at the end of a day or a week. As the speed and power of computing systems improved, the time period between data generation and analysis was reduced to a point where detection can take place as the system is operating. Historically, the literature describes this simultaneous data generation and processing as taking place “in real time”. There are more than one timeliness category (as shown in figure 2.8) [Alp97].

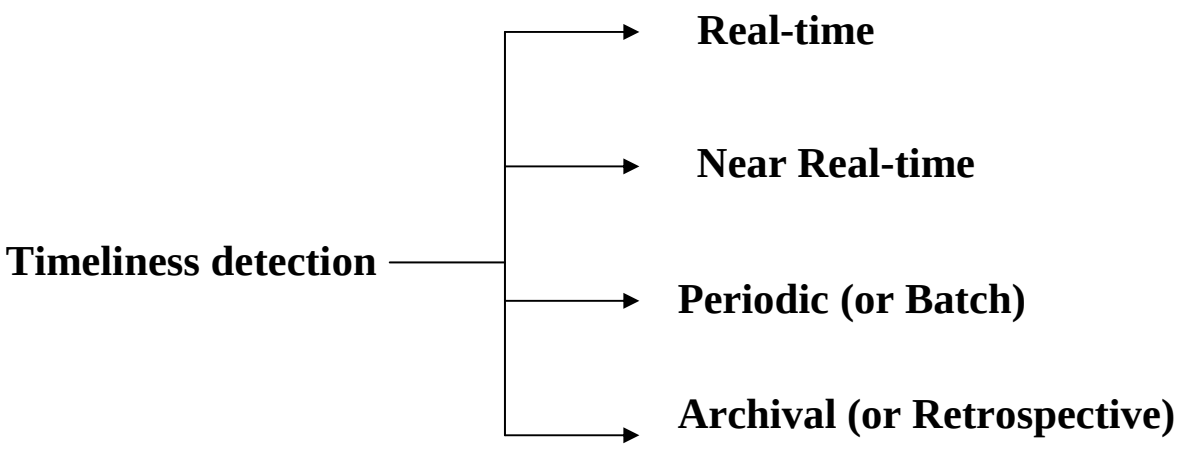


Figure 2.8 types Timeliness detection

2.6.1 Real-Time – Computer Monitoring [Wil03]

A real-time computer monitoring system needs to be able to collect information as an activity is occurring. This information would be related to the ***who, what, when,*** and ***why*** of the activity in question. An example of such a system is Carrier's Session Token Protocol. In this system, when an incoming TCP connection is made, the operating system attempts to recursively track down the source of the connection to the originating machine and user across all intermediate machines. The activity for the connection request is suspended until the information is collected.

2.6.2 Near Real-Time – Computer Monitoring [Rib01]

Near real-time computer Monitoring systems collect information as it is being generated by the system. The Frequency Binary Information's Digital Computer System (FBI's DCS v1.00) software collects digital information as part of a passive electronic wiretap. It is collecting the data shortly after it has been generated and transmitted, but without an intervening storage stage.

2.6.3 Periodic – Computer Monitoring [Rib01]

Periodic monitoring systems collect computer system information after regular intervals. For instance, tripwire can be used to periodically analyze a file system and detect any change to files or directories.

2.6.4 Archival – Computer Monitoring [Wil03]

Archival computer Monitoring tools make up the most common category of Monitoring software. These programs frequently take a computer system as a whole or major subcomponent thereof as input for analysis (e.g., file system, raw disk, memory dump). These tools construct timelines of activity, detect file modifications, attempt to recover deleted information, and so forth.

2.7 Online and Offline Monitoring

In general the monitoring application distinguish two types of monitoring ***online*** and ***offline.***

Online monitoring allows observation and potentially control of an application at run-time. In case of online monitoring, a monitoring application poses real-time constraints on the overall time its takes to generate process, disseminate, and present monitoring data.

Online monitoring is being performed using the resources of the monitored program. In particular, an online monitoring can not detect whether the program gets deadlocked or is stopped unexpectedly[Kup04].

In case of ***Offline monitoring***, a monitoring application poses no such time constraints. Hence, monitoring application may obtain monitoring data at an arbitrary time after its generation by the instrumentation [Dia04].

Offline monitoring code is executed with a different process, potentially on a different machine. One advantage of the offline monitoring is that it allows the centralized computation model, namely one monitor server can be used to monitor multiple programs [Che03].

2.8 Types of Monitoring [Edg02]

Monitoring can be in several categories. These categories are:

1- Some categories may be able to monitor **Computer components** such as:

- **Computer Monitoring**

  Administrators can use computer software that enables them to see what is on the screen or stored in their users' computer terminals and hard disks.

2- Some other categories may be able to monitor ***Computer activity*** such as: -

- **Internet Monitoring**

  Internet monitoring is increasing. Many administrators use the Internet during office hours to find all kinds of information—not related to work.

2.9 Computer Monitoring

There are three monitoring definition as follows: -

1) System provided valuable (not normally available) information with standard activity logs that could assist in identifying and possibly controlling the damage inflicted upon the system, is called "Monitoring system", see figure 2.9 [Fis00].

2) Defines monitor as "to watch, keep track of, or check, usually for special purpose"[Dia04].

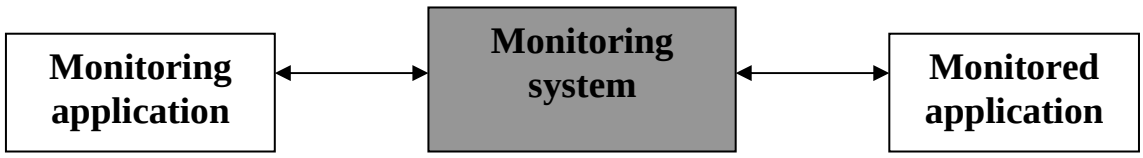


Figure 2.9 parties involved in monitoring

3) Monitoring is the interception of communication, checking of systems, the logging, recording, inspecting and auditing data[Mio02].

2.9.1 Computer Monitoring Model [Dia04]

Monitoring model of monitored application has two terms *entities* and *behavior* of these entities.

The ***entity*** concept represents some physical or logical "thing" associated with the monitored application. For example, an entity may represent a process, an object, or a component. Entities may participate in relation to form the structure of monitored application (e.g., the "association" relation between objects, and "containment" relation between compound component and one of its sub-components).

The ***behavior of an entity*** represents the dynamic characteristic of the entity during application execution. It is contained the activities that an entity performs (such as sending a massage to another entity) and possible relation among those activities (such as order in which an entity performs two activities). Two ways of modeling entity behavior for monitoring can be distinguished: ***Status-based*** and ***event-based***.

2.9.1.1 Status-based modeling

Status-based modeling abstracts from the activities that an entity performs and focuses on the information that the entity maintains at discrete moment of time, this information is called the status of the entity.

Status-based modeling models activities in a system indirectly, since any changes in the status of an entity results from performing activities.

A status vector represents the status of an entity. A status vector consists of status variables. ***Status variables*** represent an individual part of information maintained by an entity. For example, if the entity represents an object, a status variable may correspond to an object attribute. To monitor a status, monitoring system generates an instance of the status vector by recording the values of the status variables at the required moment of time. This instance is called a status report.

2.9.1.2 Event-based modeling

Event-based modeling directly models the activities that an entity performs.

The event concept represents the successful completion of some activity performed by an entity of monitored application. An event either happens in which case the corresponding activity have completed, or does not happen in which case that cannot say anything definite about the activity progress except that it did not complete. This property is called event atomicity.

2.10 Architectured of Computer Monitoring System [Rib01]

Monitoring system is subdivided into ***central element*** and ***monitor agent***. This division is necessary because monitoring is distributed.

Central or centralizing element supplies a communication interface with the user environment. The central element, from a technical point of view, is a service that runs on a machine the manager interacts with. The Central element has to receive collected data from the agents.

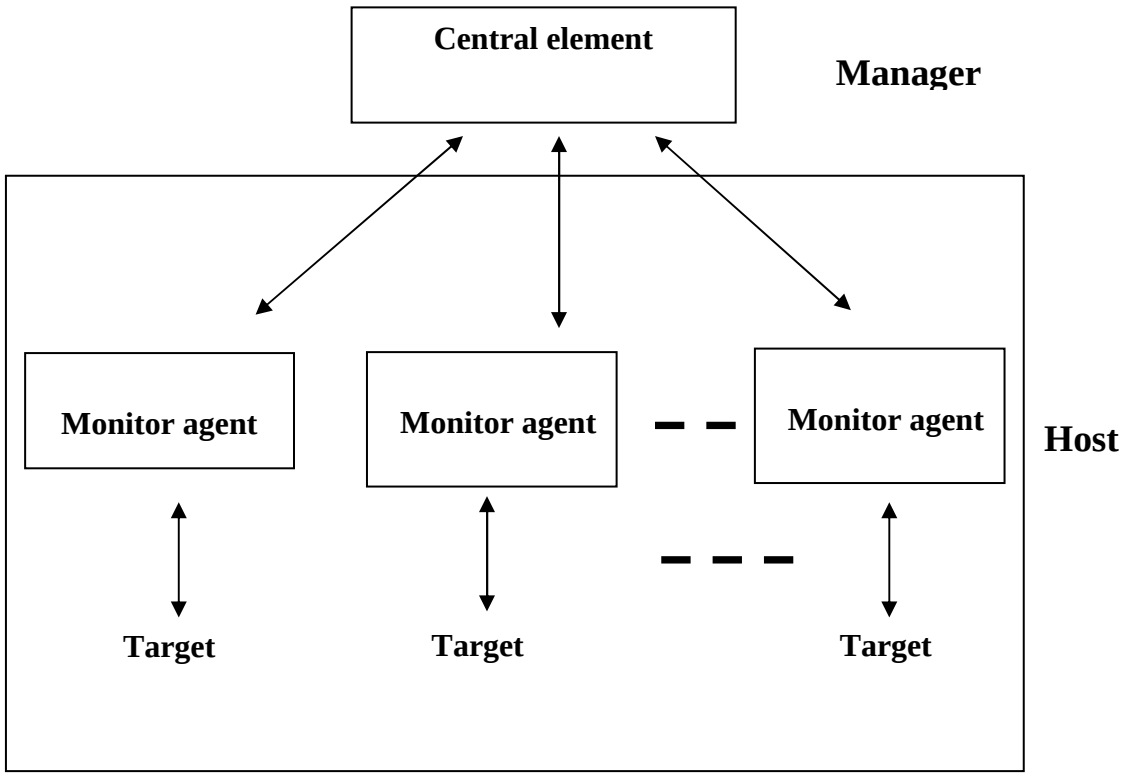


Figure 2.10 Architecture of monitoring system

Where the main tasks of a monitoring agent are to ***monitor flows***, ***collect data***, and ***send information*** about the monitoring requests made by central element. It is important to notice that monitors do not exchange massage with one another but with central element only.

2.11 Microsoft DirectX 7.0 [Dun00]

Microsoft DirectX Foundation is a set of low-level application programming interfaces (APIs) for creating games and other high-performance multimedia applications. It includes support for two-dimensional (2-D) and three-dimensional (3-D) graphics, sound and music, input, force feedback, and network communication for applications such as multiplayer games.

DirectX Foundation is made up of the set components: ***Direct3D, Direct3D*** now includes a ***Direct3DX, DirectDraw*** , ***DirectInput*** ,***DirectMusic*** , ***DirectPlay*** , ***DirectSetup ,and DirectSound***(for more information sees Appendix E).

The most useful components that could be used to design a performe monitoring system are:

2.11.1 DirectDraw [Dun00]

DirectDraw is a software interface that provides direct access to display devices while maintaining compatibility with the Windows graphics device interface (GDI). It is not a high-level application programming interface (API) for graphics. DirectDraw allows user to directly manipulate display memory .The DirectDraw component brings many powerful features to users, such as :

- The hardware abstraction layer (HAL) of DirectDraw provides a consistent interface through which to work directly with the display hardware, getting maximum performance.
- DirectDraw makes it easy for user to implement ***page flipping*** with multiple ***back buffers*** in full-screen applications (for more information see Appendix F).

2.11.2 DirectInput [Msd00]

DirectInput is an API for input devices including the mouse, keyboard, joystick, etc. DirectInput gives faster access to input data by communicating directly with the hardware drivers. DirectInput enables an application to retrieve data from input devices even when the application is in the background. The extended services and improved performance of DirectInput make it a valuable tool for games, simulations, and other real-time interactive applications running under Windows.

2.12 Windows Registry

windows registry is a central repository of information about all aspects of the computer-in particular, its hardware, operating system, applications and users. Although the registry is usually considered to be a single entity, its contents are in fact stored in more than one physical file. In windows 9x, there are two such files: SYSTEM.DAT and USER.DAT. These hold computer-specific and user-specific information respectively. It can be accessed and update under software control and also directly by users.

The registry tree is divided into six sections (five in windows NT). These sections, which all have names beginning with HKEY_, are called root keys or top-level keys. Each root key contains sub-keys, which might in turn contains further sub-keys and so on. The lowest level keys along a given branch are called values [Lew99]. The registry keys are listed below:

***HKEY_CLASSES_ROOT**: it is mainly used to keep track of file extensions and their associated applications, documents and OLE (Object Link Embedding) objects. It is a particularly large branch, with a very large number of sub-keys at the first level down [Lew99].

***HKEY_CURRENT_USER**: it is used to manage specific information about the user who is currently logged on. This information includes [Hip00]:

1- the users Desktop and the appearance and behavior of windows 2000 to the user.

2- All connections to network devices, such as printers and shard disk resources.

3- Desktop program items, application preferences, screen colors, and other personal preferences and security rights. They are stored for later retrieval by the system when the user logs on.

***HKEY_LOCAL_MACHINE:** it is a large root key. It is the home of all the computer-specific information, including details of the hardware configuration and any machine-specific settings for the installed applications. Whereas each user who logs onto the PC sees different settings in HKEY_CURRENT_USER, they all see the same information in HKEY_LOCAL_MACHINE [Lew99].

***HKEY_USERS**: This root key contains a sub-key for each user profile. There is a further sub-key, named *.DEFAULT*, which provides default values for new user profiles. If user profiles are not enabled, *.DEFAULT* stores the settings for the actual user [Lew99].

***HKEY_CURRENT_CONFIC**: KEY_LOCAL_MACHINE\Config contains details of the installed hardware profiles (this applies only to windowes 9x). Each profile has its own key within Config -named 0001,0001 etc- which holds configuration details for the profile. There is always at least one profile key [Lew99].

***HKEY_DYN_DATA**: This final root key (which is not present in NT) is a memory-resident copy of certain other registry items. it contains information, which windows needs to retrieve particularly quickly. The root key contains two sub-keys. The first, named *Config Manager* holds details of the current hardware configuration as seen by the plug-and play Configuration Manager. Windows builds this information (which is sometimes referred to as the hardware tree) by examining the

hardware during booting; the information is then updated dynamically as plug-and-play devices are installed and removed. The other sub-key is named PerfStats. This provides information about network components [Lew99].

Table of Contents

List of Abbreviations

API	Application Programming Interface
CSM	Computer Security Monitoring
CTA	Computer Technology Associates
DNS	Domain Name Service
ELM	Event Log Monitor
FTP	File Transfer Protocol
HAL	Hardware Abstraction Layer
IP	Internet Protocol
ISO	International Standard Organization
GDI	Graphics Device Interface
LAN	Local Area Network
MIDES	Monitoring Intrusion Detection Expert System
NM	Network Monitoring
NOS	Network Operating System
OSI	Open System Interconnection
PC	Personal Computer
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TM	Target Monitoring
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

References

- [Bkf98] bookshelf, **Networking Essentials,** Second Addition**,** 1998.
- [BkS99] bookshelf**, "Teach Yourself TCP/IP in 14 Days"**, Second Addition, 1999.
- [Che03] William R. Cheswick and Steven M. Bellovion, "**Automated Network Monitoring**", 2003
http://www.dcc.ichile/~rbaeza/i2p4.pdf
- [CIS04] R. T. Citron, "**Information Security",** John Wiley & Sons Ltd**,** 2004.
- [Dia04] Nikolay k. Diakov, **" monitoring distributed object and component communication"**, 2004.
- [Dun00] Robert Dunlop, Dale Shepherd, and Mark Martin, "**Teach Yourself DirectX 7 in 24 hours**", Sams Publication, 2000.
- [Edg02] Judi Edgmand," **Employee Monitoring",** published by Addison-Wesly**,** 11 October 2002.
- [Fch00] Fisch E. A. and White G. B., "**Secure computers and Networks: Analysis, Design and Implementation**", CRC press, 2000.
- [Gub03] Kumar P. Gubwani, "Design a monitoring system for Wireless LAN", 2003.
http:/www.erg.uvic.ca/~itraore/seng462-03/notes/monitoring.pdf
- [HEL00] P. T. Helton, "Security in Computing", Prentic Hall PTR, 2000.
- [Hip00] Hipson D. Peter, "**Mastering Windows2000 Registry"**, SYBEX press, 2000.
- [INFO02] INFOSEC group**, "Computer Technology Associates (CTA)",** 2002.

http://www.it.uu.se/products/reports/2002-nc.pdf

- [Kel02] T. J. Kelvinsky, "**Hack I. T. Security Through Penetration Testing**", Pearson Education, 2002.
- [Kup04] Benjamin A. Kuperman," ***A CATEGORIZATION OF COMPUTER SECURITY MONITORING SYSTEMS AND THE IMPACT ON THE DESIGN OF AUDIT SOURCES*** ", August 2004. http://kb.indiana.edu/data/aehm.html
- [Lew99] Lewis Mike, "**Understanding The Registry"**, May 1999, www.pcsuppottadvisor.com
- [Msd00] MSDN Microsoft Corporation**,** "**DirectX New Features Review**", April 2000.
- [MO00] David T. Moreas, "**Monitoring Policies**", AT&T Bell Laboratories, 2000.
- [NiK02] Donald L. Nikpip, "**Information Security Protecting the Global Enterprise**", CRC Press LLC, 2000.
- [Rib01] Marcelo Borges Ribeiro, Lisandro Zambenedetti Granville, Maria Janilce Bosquiroli Almeida, and Liane Margarida Rockenbach Tarouco, "**QoS Monitoring System on IP networks**", 2001. http:\www.it.uu.se/resurch/reports/2002-006/2002-006-nc.pdf.
- [Tai00] Manar Saad Salih Al-Taie, "**Development of Windows Malicious Codes for Remote Computers**", a thesis Submitted to the college of science Al-Nahrein University for M.Sc. degree 2002.
- [Tan96] Tanenbaum S. Andrew, "**Computer Network"**, Third Edition, 1996.
- [Tod01] Chad Todd, "**HACK PROOFING Windows 2000 Server"**, SYN GRESS, 2001.

- [Web1] ”**Monitoring and Evaluation**”, Http:\www.ucm.ogr/catalog/ed7.pdf, 2002.
- [Web2] "**Adaptive Methods for Activity Monitoring of Streaming Data”,** http://www.secodes.net/scb5.pdf, 2002.
- [Wil03] Brandon Williams, “**Remote Logging and Monitoring**”, Published by New Riders, 2003.

Supervisors Certification

We certify that this thesis was prepared under our supervision at the Department of Computer Science / College of Science / Al-Nahrain University, by D**alal Naeem Hmood Al-Zaidi** as a partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature: Signature:

Name: **Dr. Taha S. Bashaga** Name: Dr.**Venus W. Samawi**

Title: **Lecturer** Title: **Assistant Professor**

Date: / /2005 Date: / /2005

In view of the available recommendations, I forward this thesis for debate by the examination committee.

Signature:

Name: **Dr. Taha S. Bashaga**

Title: Head of the Department of Computer Science, Al-Nahrain University.

Date: / / 2005

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة النهرين
كلية العلوم
قسم علوم الحاسبات

نظام مراقبة حاسوبية تحت نظام النوافذ

رسالة
مقدمة الى كلية العلوم في جامعة النهرين كجزء من متطلبات
نيل شهادة الماجستير في علوم الحاسوب

مقدمة من قبل
دلال نعيم حمود الزيدي
(بكالوريوس علوم حاسوب ٢٠٠٢)

المشرفون

د. طه سعدون باشاغا
د. فينوس وزير سماوي

Republic of Iraq
Al-Nahrain University
College of Science

Windows Based Target Monitoring System

A Thesis
Submitted to the College of Science, Al-Nahrain University in Partial Fulfillment of the Requirements for The Degree of Master of Science in Computer Science

BY
Dalal Naeem Hmood Al-Zaidi
(B.Sc. 2002)

SUPERVISORS

Dr. Taha S. Bashaga
Dr. Venus W. Samawi

جامعة النهرين
كلية العلوم
قسم علوم الحاسبات

Al Nahrain University
Collage of Science
Department of Computer Science