

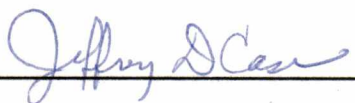
To the Graduate Council:

I am submitting herewith a thesis written by Chad Matthew Pfleger entitled "Local Area Networks: Security Problems and Concerns." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.



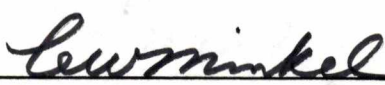
Dr. T.H. Dunigan, Major Professor

We have read this thesis
and recommend its acceptance:





Accepted for the Council:



Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor or, in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature

CM Pflieger

Date

5/26/89

**LOCAL AREA NETWORKS: SECURITY
PROBLEMS AND CONCERNS**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Chad Matthew Pflieger

August 1989

For William and Josephine

*To my Parents, who taught me that in life more is
achieved by participating than by observing.*

ACKNOWLEDGMENTS

I wish to acknowledge the many contributions of Dr. Thomas Dunigan who has provided the stimulus necessary for the urge to continue to learn. For all the time and support that Dr. Dunigan has provided to make this whole endeavor meaningful.

I wish to thank Dr. Jeffrey Case and Dr. William McClain for their service on the thesis committee and for their helpful suggestions.

I wish to thank my management at Martin Marietta Energy Systems, Inc., specifically S. L. Chapman and S. K. Penny, Sr., for their encouragement and support.

I also wish to thank Dr. William McClain, whose unselfishness in teaching courses at night over the past six years made the pursuit of a graduate degree much easier.

Finally, I wish to thank my wife Lila, whose support, understanding, and patience over the past six years have been appreciated more than she will ever know.

ABSTRACT

In the early years of computers, not much, if any, consideration was given to connecting computers to each other. Computers were initially linked to each other via dedicated links, and these could be controlled. But as time and technology changed, several aspects of computers and networking also changed. Still, there was no need for concern because users had the ability to send/receive data via remotely submitted jobs to the computers. Users still did not have true interactive access to the computers. However, technology pressed on and the capabilities of the computers and networking provided innovative features for the users. Now the users could access the computers from almost anywhere at anytime via true interactive high-speed data circuits. Also, users could share data with other users within the same local area or facility. Thus, the local area network (LAN) was established and became widely recognized and accepted. The high visibility that the LAN received raised several concerns and problems.

Presented herein are security concerns and problems that need to be examined before and after a LAN is purchased and installed for use. Some cases described within this document may not actually be solvable, thus recommendations will be made to address these types of situations. The unsolvable situations may be due to the lack of technical expertise in development of both hardware and software. The growth and uncanny acceptance of the LAN has caused technology to be pushed to the limit. With so much hardware, software, and vendor support available to users, the administrative, configuration, and

security controls on the LAN have been neglected. Finally, due to successful break-ins of several computers by hackers via LANs, security controls are now an important aspect of networks. Most security controls can be implemented on LANs by either physical, technical or administrative methods. Discussions of these controls are presented in detail, as are views on networking standards. Views on these standards will be presented as they pertain to networking protocols already established. Methods will be discussed that can be used to further enhance security to local area networks.

TABLE OF CONTENTS

CHAPTER	PAGE
1 INTRODUCTION	1
1.1 Overview of Subsequent Chapters	2
2 INTRODUCTION TO LANs	5
2.1 International Standards Organization Reference Model	6
2.2 Examples of LANs	13
2.2.1 Ethernet Network	13
2.2.2 Token Ring Network	17
2.2.3 Summary of Ethernets and Token Rings	23
2.3 TCP/IP, DECnet and OSI	24
2.4 Connecting LANs to other networks	29
3 PROBLEMS WITH LANs	33
3.1 Promiscuous Mode of Operation	34
3.2 Node Impersonation	36
3.3 Disruption of Service	40
4 PROBLEMS OF SPECIFIC LAN PROTOCOLS	42
4.1 Promiscuous Mode	42
4.1.1 TCP/IP and the Promiscuous Mode Problem	42

4.1.2	DECnet and the Promiscuous Mode Problem	45
4.2	Node Impersonation	48
4.2.1	TCP/IP and the Node Impersonation Problem	49
4.2.2	DECnet and the Node Impersonation Problem	52
4.3	Disruption of Service	60
4.3.1	TCP/IP and the Disruption of Service Problem	61
4.3.2	DECnet and the Disruption of Service Problem	63
4.4	OSI and Various Security Problems	64
4.4.1	Summary of OSI	65
5	SOLUTIONS TO LAN PROBLEMS	72
5.1	Protecting LANs and Data	75
5.1.1	General Protection of LANs	75
5.1.2	Encryption of Data	78
5.1.3	Recommended Solutions	97
	LIST OF REFERENCES	102
	VITA	106

LIST OF TABLES

TABLE	PAGE
1-1 Network Protection Mechanisms	4
2-1 ISO/OSI Reference Model for Networks	7
2-2 ISO/OSI Reference Model and Internet Protocol Suite	25
2-3 ISO/OSI Reference Model and DNA	26
2-4 ISO/OSI Reference Model and OSI Protocol Suite	28

LIST OF FIGURES

FIGURE	PAGE
2-1 Examples of LAN Topologies	9

CHAPTER 1

INTRODUCTION

This paper describes some of the problems associated with the use of broadcast local area networks (LANs). The discussion of LANs in this thesis does not include personal computer networks, such as Novell¹, Corvus², or others. There are a set of concerns (presented below) that pertain to any LAN. If such concerns are not adequately addressed, the protection of data as it migrates on a network will not be achieved. The concerns are as follows [1]:

- the ability to protect data from unauthorized manipulation,
- the ability to provide data integrity,
- the ability to protect data from any type of compromise,
- the capability to verify the sender of any data, and
- the capability to verify the receiver of any data.

This list is by no means complete, and there may be several other items that need attention also. But this list covers most of the common security concerns and vulnerabilities associated with LANs. These security concerns

¹ Novell is a trademark of Novell Incorporated.

² Corvus is a trademark of Corvus Systems, Incorporated.

must be addressed because there exists a set of specific problems that can be used to expose data and information based upon the concerns. The specific set of problems that pertain to existing vulnerabilities (there may be others) with broadcast LANs are:

- promiscuous mode of operation,
- node impersonation on the network, and
- disruption of service.

In general, LANs have revolutionized the concept of networking. Data is now easier to transfer from one system to another; for example, the speed at which the data is transferred is greater than ever before, and the risk of loss is higher than before. There are several security concerns and problems associated with LANs. These concerns and problems need to be discussed so that others may learn about them and correct them or force the industry to address them.

1.1 Overview of Subsequent Chapters

Chapter 2 will provide a background which will present the various topologies that are used for LANs, the types of media that are used for LANs, and the standards that have been developed for networks. Two widely used LANs, namely Ethernet³ and token ring, will be presented. The discussion will include

³ Ethernet is trademark of the XEROX Corporation.

comments and ideas concerning connections from these LANs to other networks via gateways, routers, and bridges.

Chapter 3 will provide a detailed analysis of specific problems associated with LANs. Attention will be given to the problems of promiscuous mode, node impersonation, and disruption of service.

Chapter 4 will discuss how different types of network protocols address the problems and concerns associated with LANs. This discussion will evolve around three network protocols:

- Transmission Control Protocol and Internet Protocol (TCP/IP),
- Digital Equipment Corporation's DECnet,⁴ and
- Open System Interconnect (OSI).

Although primary discussion will pertain to TCP/IP and DECnet, the OSI protocol will be presented and discussed as the "new" standard for future LANs to follow.

Chapter 5 will present some possible solutions to the problems and concerns that are associated with LANs. Some of the solutions are derived from information and work already presented by others and being applied on LANs today; others are from work being done but not presently in use at this time. Also, recommendations will be presented based upon the various problems and their solutions.

⁴ DECnet is a trademark of Digital Equipment Corporation.

Table 1-1 [2] illustrates some measures that may be taken to help control the possible damage resulting from these problems and concerns. And finally, the

Table 1-1: Network Protection Mechanisms

LEVEL	PROTECTION FEATURE
Most Secure	Data File Encryption
*	Hardwired Terminals
*	Network User Verification
to	File Access Restriction
*	File Passwords
*	Account Numbers and Passwords
Least Secure	Terminal Access

following ideas from the author will be presented and discussed as additional methods for LAN protection:

- management control for LANs,
- various operating system controls,
- network monitoring, and
- encryption of *all* data.

CHAPTER 2

INTRODUCTION TO LANs

As stated in Chapter 1, there are some problems and concerns associated with LANs. Before discussing these problems and their solutions, it is appropriate to define “broadcast” LANs. A broadcast LAN is a network in which all nodes can have access to all data on the network. Even though data is on the LAN, other systems have the capability, the potential, and the opportunity to capture this data. There are many examples of LANs, but the Ethernet (bus structure) and the token ring (ring structure) will be presented in this chapter.

Networks were designed and developed to allow interconnectivity among several types of data processing equipment. Being distributed and allowing such connectivity makes the data on the network difficult to protect. LANs are used primarily to provide network capabilities for users within a limited geographical area. For some LANs, a connection is made from the LAN to the main network backbone or trunk. The LAN allows users within the same area to share data and resources without flooding the primary trunk with such traffic. Local area networking was designed and implemented to fully utilize the distributed networking concept, and, as such, little or no attention was given to the actual control or protection of resources that use LANs. Several different LANs now

exist with insufficient control or protection. In this chapter, discussion is focused on the network reference model, which will provide a standard to help define some of these problem areas.

2.1 International Standards Organization Reference Model

The International Standards Organization (ISO), which has members from several standard organizations which represent various countries, has developed the Open System Interconnect (OSI) reference model to serve as the foundation for computing networks [3]. The networking model developed by the ISO has been adopted by many vendors, and descriptions of the various layers of this model will be presented. These descriptions will focus on each layer and how the other layers relate to each other. Examples of how this seven-layer network reference model relates to the Internet protocol suite (TCP/IP), DECnet, and OSI will be presented later in this chapter.

In 1977 the ISO formed a subcommittee for the sole purpose of defining the OSI. This subcommittee was tasked with developing the basis and architecture for standard network protocols. After 18 months of work, the subcommittee revealed the seven-layer OSI reference model [4].

Table 2-1 reflects the ISO/OSI reference model for networking [5]. The

Table 2-1: ISO/OSI Reference Model for Networks

Layer	ISO/OSI
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

“Physical” layer is the actual hardware connection to the network. This connection from the computer equipment to the network could be twisted-pair wire, coaxial cable, or optical fiber. This layer is concerned with the medium used to connect resources to the network. This medium will be the carrier of data to and from the computers that are attached [5]. The “Data Link” layer is responsible for establishing the communication paths between adjacent computers on the network. This layer readies the computers for transmission and receipt of data [5]. The “Network” layer contains actual data from the user and packet information that deals with routing. This layer is also concerned with network congestion and packet control [5]. The “Transport” layer is responsible for the logical link for the network users. This layer controls data flow and

system-independent services [5]. The “Session” layer is the user interface to the network. This does not imply that this layer is used to establish the physical connection between the users and the computers. This does imply that a logical link is established by this layer. Also, it provides access for a user to log in to a remote system and move data. This layer is usually the most complex [5]. The “Presentation” layer deals with remote utilities and gateway access between other systems on the network [5]. The “Application” layer is concerned with actual user applications that can be used to transfer data to and from nodes on the network. These applications can be utilities, system services, or user programs that can be utilized to move data [5].

The various vendors are now designing and implementing better security controls and features for LANs. The physical layer of LANs are generally configured in one of the following network topologies [6]:

- bus structure,
- star structure, and
- ring structure.

Figure 2-1 depicts these LAN topologies.

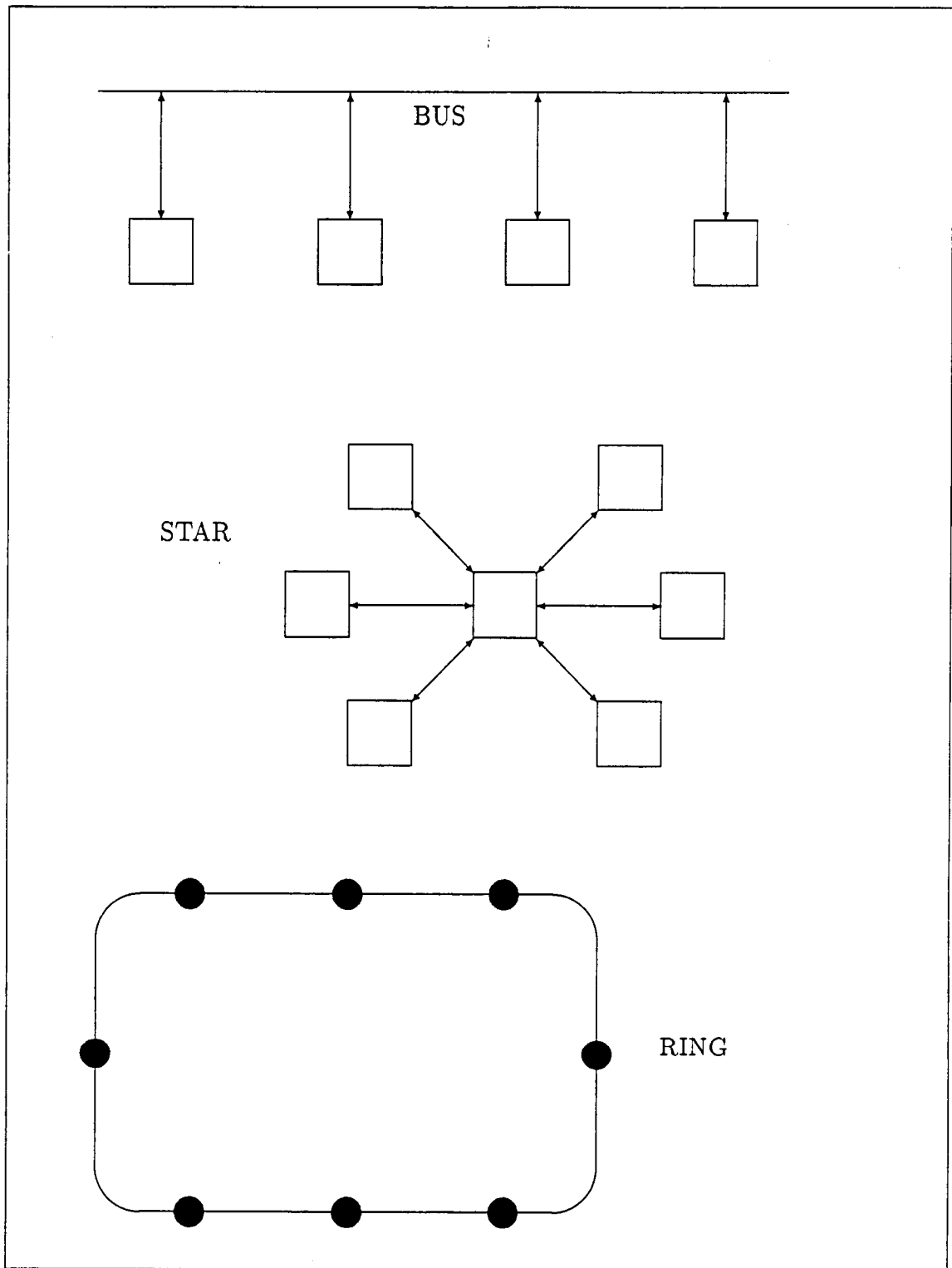


Figure 2-1: Examples of LAN Topologies

Some LAN topologies provide better control and protection of users' data, while adding some protection to equipment. The star, for example, is somewhat more difficult to access, because the controller is connected to the remote hosts via a point-to-point line. Thus, the controller may be able to detect if a physical connection has been made, because it is more difficult to impersonate other hosts on dedicated point-to-point lines. This is especially true if the original host is active. Signal loss on point-to-point lines due to taps or other problems may provide the controller with some indication of a problem. The star structure connects all network devices to a single, centrally located station. This single, centrally located station can be either a passive device or computer node. If this central station is a passive switch, then the data on the network will be "broadcast" to all other nodes. But if the central station is a computer, or other intelligent device, then the data on the network will be routed to the destination node and not "broadcast" to all other nodes [7]. The star structure can be configured in a manner that makes it more difficult to obtain all other users' data, as the controller sends/receives only data that is intended for a specific network node. Note, however, that recent developments in technology provide the use of twisted-pair Ethernet in a star configuration. In twisted-pair Ethernet, all data is broadcast outward from the star to all nodes on the network. This occurs because the central station acts as a passive switch and does not know where to route the data. This new technology will present the same broadcast problem for the star as presently exists for the other types of LAN networks. For

the purpose of this study, the focus shall remain with the first version of the star network. All data must pass through the central controller before being passed to the correct network node. If the central controller is ever compromised, then the entire LAN is vulnerable.

Both the ring and bus send data out along the LAN, thereby providing all network devices with easier accessibility to data than with the star structure. The bus structure (for example, Ethernet) allows all stations to be connected to only one cable that makes up the entire network. The data migrates out on the cable in both directions, thus the data is “broadcast” to all network devices [7].

The ring structure (for example, a token ring) connects each station to its neighbor, which in turn is connected to its neighbor, and so on, until all the devices are connected in a closed loop. The data on this type of LAN is, in most cases, passed in one direction around the ring. Most ring networks provide two network connections: one for the primary transmission connection from the node to the network concentrators (and eventually to the nodes) and the other for secondary transmission connection in the event that the primary cable becomes damaged. This allows data to reach most nodes even if the network should fail. The ring structure has the ability to send data backward around the ring in case of a network failure, thereby avoiding the damaged node [7].

Usually, LANs adopt one of the following media for their physical layer:

- coaxial cable,
- twisted-pair wire cable, or

- fiber optic cable.

One of the most frequently used media for LANs is coaxial cable. Coax is used basically because it is less expensive to install than fiber optics and provides higher data transfer speeds than twisted pair. Because of its versatility, coax seems to be the standard set by the industry [8]. There are several reasons why coax may be preferred over either twisted-pair or fiber optics. Twisted-pair lines have distance limitations which are more restrictive than those of fiber or coax, and twisted pair lines do not support as many different capabilities as fiber or coax. At present, fiber optics seems an excellent choice for point-to-point connections, but not as good for multi-drop connections [8]. The entire fiber technology is changing, the installation cost is dropping, the multi-drop connections are beginning to be real, and the expertise of using fiber is becoming commonplace. Based upon this, the choice of coax for LANs will no longer be the only choice. The choice of which type of cable to use is quite important, as it is easier to physically tap into a coax cable than it is to tap either fiber lines or twisted-pair lines. Because most LANs use coax cable, such as Ethernet, there is no need to physically break the cable; rather, a "vampire" type of connector simply clamps onto the coax cable. Tapping twisted-pair cable can be done like tapping fiber optic cable, but the ability to tap either fiber optic or twisted pair relies on the expertise of the individual to know exactly how to tap the media. It should be noted that the ease in which Ethernet cables can be tapped and "new"

nodes can be added is relevant to problems associated with LANs regardless of which media is easier to tap.

2.2 Examples of LANs

Discussion here focuses on two types of broadcast LANs; namely, the Ethernet and token ring networks. Brief discussions will be presented to define these networks and explain how they are configured. Some time will be devoted to the mechanisms that can be used to help combat certain security problems that are common to these networks. Chapter 5 provides more detail on solutions for protecting data from promiscuous mode, node impersonation, and disruption of service.

2.2.1 Ethernet Network

The Ethernet technology was developed jointly by three companies which were Xerox, Intel, and Digital Equipment Corporation (DEC). The Ethernet technology utilizes a baseband, single-channel network. This scheme allows multiple connections on the same cable, as opposed to dedicated point-to-point links, and supports several types of network devices from various vendors. It is not the intent here to cover all types of Ethernet devices or to completely analyze the capabilities, but rather to provide a basic understanding of Ethernet. New designs and developments pertaining to Ethernets have indicated that many LAN

topologies can support and accommodate it. Ethernet is only the physical and data layers of the network, and all the necessary interfaces, node computers, and other associated components actually are the network. The mainstay for the users of Ethernet, at least for today, is the bus network topology. Nodes can be added very easily to any Ethernet network. In most cases, network devices can be added by clamping onto the cable. This clamp-on technique involves a screw-in tap that pierces the cable and provides the connection from the cable back to the network device. Based on this, it is easy to add nodes and difficult to control access to the cable, thereby making it extremely difficult to administratively control and monitor the network for any newly attached devices. Ethernet was designed and developed for the speed of moving data and also for the ability to support distributed networking. But as previously noted, security issues were not adequately addressed during the effort to accommodate faster data processing and greater distributed networking.

The Ethernet standard states that each hardware device or interface from the host node to the network is assigned a unique address, which is also known as the hardware or physical address [9]. Such addresses are assigned by the Xerox Corporation to vendors who produce the hardware that interfaces to networks that utilize the Ethernet technology. The Ethernet physical address is comprised of 6 (8-bit) bytes for a total of 48 bits. This 48-bit address, in most cases, is split into groups of 2 (3-byte sections), which allows Xerox to establish unique address identifiers for companies that sell Ethernet hardware and to establish identifiers

for the hardware devices themselves. Usually, the entire 48-bit Ethernet address is never changed or modified. Most network protocols, such as TCP/IP and Xerox Network Services (XNS), use the unique Ethernet address that comes with a particular hardware device. These protocols utilize this address as part of their network routing techniques, but the DECnet protocol alters the hardware address. Discussion about why DECnet alters the hardware address is presented below.

For DEC, the Ethernet address is broken into two sections. The first section is comprised of 4 (8-bit) bytes. These 32 bits are assigned to DEC for their specific internal use as a manufacturer of Ethernet hardware. The second section is comprised of 2 (8-bit) bytes. These 16 bits are encoded on read-only memory (ROM) chips that identify the physical hardware device. In most cases, once the network is brought up, the DEC operating system enables the Ethernet interface and provides it with a valid set of addresses [9]. These DEC nodes can be configured for either Ethernet network protocol or for DECnet network protocol, but not both. Therefore, the DEC systems can have the default addresses altered, thereby allowing the interface to "listen" on the network for any data that is associated with a valid address. Once such data has been received, the interface passes it on the node for further processing. The hardware interface performs many duties, among which is to provide filtering of data packets and to act as a buffer to help control the amount of data that is passed onto a node [9]. The interface also has the capability to be configured to enable multicast (ability to

send data to select network nodes) and broadcast (ability to send data to all network nodes), to modify Ethernet addresses, and to set promiscuous mode. Discussions about how these different capabilities can be enabled or set are presented in Chapters 3 and 4. This hardware interface passes data packets onto the host if, and only if, the addresses of the Ethernet match the address for that particular host node [9]. If there is no match, the interface does not receive any data, and the process of “listening” continues.

There are several interface devices that are available from either DEC or other vendors that can interface the computers to Ethernets. Two such devices from DEC are the DEUNA⁵ and DELUA⁶ network interface devices. Both of these devices can be used on any DEC computer that uses a Unibus to connect peripherals to the main CPU components. The above devices are used in an Ethernet bus configuration, and, in most cases, the DECnet protocol is used. But other LAN protocols, such as TCP/IP, can also be utilized with these same hardware interface devices. While these devices are connected to the computer's bus architecture, the computer's operating system actually controls the devices on the bus. A privileged programmer, system manager, or any privileged user (certain required privileges are presented later) who knows the internals of the computer can develop software programs that will enable his/her node on the network to operate promiscuously, that is, to receive all data packets that are placed onto the network. There are two methods that privileged users can utilize

⁵ DEUNA is a trademark of Digital Equipment Corporation.

⁶ DELUA is a trademark of Digital Equipment Corporation.

to modify these devices which can increase the vulnerability of data on the network. One method is that privileged users can write their own device driver software that will place the device in promiscuous mode or that will modify the entire 48-bit Ethernet address. On some systems the privileged user can call or use special operating system routines to enable promiscuous mode or to change or modify the 48-bit Ethernet address.

2.2.2 Token Ring Network

Earlier, the ring topology for LANs was defined as a structure that connects each station to its neighbor which, in turn, is connected to its neighbor and so on until all the devices are connected in a closed loop. The ring is usually composed of several dedicated point-to-point links [10], rather than one single cable. The ring is composed of a primary net and a secondary net which provides redundancy. If something causes the primary net to become inoperative, a switch to the secondary net is made. In most cases, a secondary net is not needed; however, redundancy is always welcomed.

The main network is normally connected to wire concentrators, which, in turn, are connected via receiver adapters (incoming) to the nodes (e.g., personal computer, mainframe, etc.). If this adapter determines that the data is to be received, the adapter assumes control of the data. If the adapter does not recognize the data (by valid address), it passes the token to its neighboring node.

The token and/or data is sent from the node back to the network concentrator via a transmit (outgoing) wire. Once at the concentrator, the concentrator passes the token (with new data or with the original data) back onto the primary net for distribution to the next nodes. This addition to the original definition of the token ring presented previously provides detailed information that helps to explain how the ring network may continue to operate if a node is removed or if the network is damaged. In the ring topology, network concentrators can be configured to automatically adjust to reroute and reconnect the ring to avoid the damaged area on the ring. The damaged area is, in effect, isolated from the rest of the ring [10]. These network concentrators contain intelligence that allows the entire network to be monitored and to have various service functions performed [10]. This is somewhat different from the bus topology, where the entire network is stopped if the bus is damaged [10]. This is due primarily to the fact that the bus relies on a single cable that allows multiple access with no ability to control or isolate damaged components. But even if the network (bus) manages to continue to operate, host computers that are connected on either side of the damaged network will no longer be able to address each other [10].

In the token ring environment, specifically the IBM token ring, the data tokens, known as "frames," are removed from the network by the valid receiving node that recognized its address in the frame header. After the data packets (frames) are received by the valid node, the entire frame, along with data added to the frame that indicates that the data was received, is sent back out on the

ring. The node that received the data enables a received bit within the data stream that is used to signal to other nodes that the data has been received. Thus nodes that are not located between the source and destination node cannot copy the data, because the received bit is enabled [11]. This information is passed on until the original node receives the frame. The original node removes the frame from the network and places a "free" token back onto the network. The next node wishing to transmit data obtains control of the "free" token and replaces it with data frames and the process begins again [12].

Each adapter that connects the personal computer (PC) or workstation to the token ring requires initial configuration of the adapter handler. In most cases, this adapter handler is referred to as the **TOKREUI** [11]. The **TOKREUI** (Token Ring Extended User Interface) comes with the hardware adapter in the form of magnetic media that the user loads upon startup. This software allows the user to interface with several types of communications software and to perform other types of interfaces. By using the **TOKREUI**, most complex adapter capabilities are reduced, and the user can utilize previously developed communications software. Upon startup of the PC or workstation, the **TOKREUI** procedure is begun by typing **TOKREUI na0, na1, sr0, sr1** [11]. All of these parameters are optional, and default values will be input if the user decides not to configure the adapter. The **na0/na1** parameter pertains to the node address of the device that is requesting access to the network. If no value is entered by the user, or if 0 is entered, the node address for this device will be the address

that is resident on the ROM. This information is contained on a chip that is part of the adapter card. The `sr0/sr1` parameter deals with memory for the adapter. This memory allows messages, data, or other information to be stored once it is received from the network. From this discussion it appears that the ability to impersonate nodes on a token ring network is as easy as it is on the Ethernet network.

The ability to enable promiscuous mode on token rings is somewhat more difficult than other network protocols. The token ring hardware interfaces come with special electronic chips. The following discussion explains how the IBM token ring adapter hardware is used. Various sources indicate that IBM sells two distinct types of chip sets for the token ring network adapter hardware. One adapter is for the microchannel based network and the other adapter is for the 286-based systems. The IBM hardware does not allow the user the ability to enable promiscuous mode. IBM does utilize a network monitor/analyzer in the network for network maintenance (which can be used to copy all network data). This particular capability requires expertise known by IBM support personnel. The IBM adapters can be programmed by the user to override the preprogrammed node address on the PROM and a new node address can be entered (impersonation). System software will check upon start-up for the validity of the node address (this is discussed later). Therefore, it may be possible to impersonate a node, if that node is not on the network and the user knows

that nodes address. It is not possible to use IBM hardware adapters and enable promiscuous mode.

Another vendor Texas Instruments (TI), manufactures a chip set for token ring adapters that does allow the user to enable promiscuous mode. The user can develop a specific hardware adapter using the TI-based chip set and then set the adapter into promiscuous mode. This promiscuous mode is enabled with software that interacts with the TI chip set. The TI chip sets are not interchangeable with IBM chip sets or hardware. Also, the impersonation capability exists with the TI chip set and hardware. There appears to be some control here that is not available in the Ethernet environment.

There are several hardware adapters that are available for use on the token ring. These adapters are used to connect the network to the device that wishes to access the ring network. These adapters have the capability to either set unique network adapter address (48-bit node address) or, if no address has been set, the adapter will automatically assign a node address to the device. When the adapter assigns the node address, the adapter will perform tests to check the address to ensure that it is valid; however, the test does not check for address uniqueness [12]. In most cases, the adapters ability to assign their own node address can be disabled using jumpers on the adapter board. Thus, upon initialization, the adapter checks to see if a valid node address exists. If a node address is not present, the communications processor will obtain a node address

from the address module. If a node address is present, then the initialization process continues.

There are several phases that the adapter must pass through before the node can be accessed from the network. There are various diagnostics that are performed along with system tests, all of which must pass before the adapter can be used. One interesting test actually checks and verifies that the node address is valid for this adapter. There is a test that also checks for valid node addresses on the ring and, if this test returns with a duplicate node address indicator, the adapter knows that another station on the ring has the same node address. It appears that if another node already has the same address, the adapter will let the user know during the initialization and diagnostic checkout phase. Once the adapter checks out completely, the node can be accessed from the network. But the ability to impersonate nodes in the token ring network is basically the same as the other types of network environments. The user could reconfigure the adapter and select new node addresses; the user could pick a node address that is known to exist, and the user may also know that the valid node is off-line for repair work or for other reasons. Thus, the user could come back on the network impersonating the other network node. At this point, all the data that was sent to the valid node would be arriving at the node impersonating the valid network node.

2.2.3 Summary of Ethernets and Token Rings

There are certain physical controls that can be used to help protect all networks from the problems discussed throughout this paper. Some of these controls include good administrative and management policies and procedures; others require physically securing the network hardware and associated components, and controlling the number of privileged users who have access to the host computers. It has already been noted that the "broadcast" of data along both types of networks is nothing unique, and there is no absolute way to protect all the data all of the time. Some innovative ideas must be developed that will allow users to fully utilize the distributed networking and sharing of data/resources while providing some level of protection. The level of protection that must be used to ensure the integrity of the data must be implemented within the various layers of the ISO/OSI reference model. There is no one LAN that addresses these problems better than any other. Since most all of these LANs will support various types of networking protocols, the protection levels have to be incorporated within the middle layers of the standard network reference model (ISO/OSI). Discussion about this type of solution is presented in greater detail in Chapter 5.

2.3 TCP/IP, DECnet and OSI

Examples of how specific LAN protocols, TCP/IP, DECnet, and the OSI map to the seven-layer reference model are presented. The tables that follow help to point out the differences of these network protocols as they relate to the ISO/OSI reference model.

The TCP/IP protocol, better known as the Internet protocol, is presented first. The Internet protocol suite was developed by the Defense Advanced Research Projects Agency (DARPA), an agency within the Department of Defense, in the early 1970s as part of a research program [13]. DARPA was chartered to develop network protocols that would allow different physical networks to be connected to different computers [13]. The Internet protocol maps directly to layers 1 to 4 of the ISO/OSI reference model, but there is no direct map between layers 5 through 7 from the Internet to the ISO/OSI model.

As can be seen from Table 2-2, the Internet protocol suite combines layers 1 and 2. In these first two layers, the physical equipment used is specific. It is the method used to pass data from one point to another along the physical media [13]. The Internet protocol suite supports several different types of physical media such as Ethernet connections, serial line connections, token ring, and X.25 connections.

Table 2-2 [13] depicts the reference model of the ISO/OSI and how the Internet protocol suite maps to it. The Internet protocol suite provides for network-

Table 2-2: ISO/OSI Reference Model and Internet Protocol Suite

Layer	ISO/OSI	Internet Protocol Suite		
7	Application	Error Messages	NFS	FTP
6	Presentation		Servers	Telnet
5	Session			SMTP,R-Utilities
4	Transport	Transmission Control Protocol (TCP)		
3	Network	Internet Protocol (IP)		
2	Data	X.25, Ethernet, ProNet		
1	Physical	Token Ring, Serial Line		

to-network communications at the third level (“Network” layer). In this layer, the Internet protocol provides for messages to be moved across the Internet [13]. In this layer there is no mechanism to control the flow of data, data reliability, or any other system services access [13].

The Transmission Control Protocol (TCP) layer of the Internet protocol suite maps to layer 4, the “Transport” layer, of the ISO/OSI model. In this layer, TCP controls the flow of data and provides reliable communications between interconnected computers that are on the network [13].

Layers 5, 6, and 7 of the ISO/OSI model are all combined in the Internet protocol suite into an application layer. In this layer, protocols like File Transfer Protocol (FTP), remote login (TELNET), electronic mail (SMTP), and other utilities are provided [13]. Most network security features of the Internet protocol suite are provided at the application layer, which uses authentication and resource protection mechanisms provided by the host operating system. Other security options exist within the Internet Protocol (IP) itself, but these are generally never used [14].

Table 2-3 [15] depicts the reference model of the ISO/OSI, and how DEC's

Table 2-3: ISO/OSI Reference Model and DNA

Layer	ISO/OSI	DNA	
7	Application	User	
6	Presentation	Application [Others,DAP]	Network Management
5	Session	Session	
4	Transport	End Communication	
3	Network	Routing	
2	Data	DDCMP,X.25,Ethernet	
1	Physical	Physical	

internal network, called Digital Network Architecture (DNA⁷), maps to it.

⁷ DNA is a trademark of Digital Equipment Corporation.

Layers 1 and 2 (physical and data) in the DNA relate to the type of network media and the type of format used to packetize the data. Layer 3 deals with the routing of data from node to node. The end communication layer of the DNA (layer 4) provides functionality for different networks like IBM's SNA⁸. The "Session" layer bridges the end communications layer and the various logical functions of the operating systems to the communication layer. The application layer maintains all the logical paths for the network. The user layer is where the end user will access the remote resources. Such resources include SET HOST (virtual terminal), file transfer, and the downline loading of tasks and system files. The DNA in its present form maps closely to the ISO/OSI reference model. The reason for the statement "in its present form" is that DEC is in the process of modifying the DNA to more closely match the OSI protocol stack. The new DNA is scheduled for release with DECnet Phase V.

The original network data is expanded as it migrates through the network layers on its way from host to host via the network. Within the DNA, the "Data Link" data, the "Routing" data, the "End Communication" data, the "Session" data, the "Network Control" data, and other data are added to the original data as it moves through the network layers [16, 17].

⁸ SNA is a trademark of International Business Machines, Inc.

Table 2-4 [18] depicts the reference model of the ISO/OSI and how the (OSI) protocol maps to it. As can be seen from Table 2-4, the OSI network protocol

Table 2-4: ISO/OSI Reference Model and OSI Protocol Suite

Layer	ISO/OSI	OSI Protocol Suite		
7	Application	X.400	File Transfer	Future Use
6	Presentation	OSI Connection Oriented Presentation Protocol ISO 8823		
5	Session	OSI Connection Oriented Session Protocol ISO 8327		
4	Transport	Transport Protocol ISO 8073		
3	Network	Network Layer Protocols ISO 8473 and ISO 8208		
2	Data	CSMA/CD, TOKEN BUS, TOKEN RING, X.25 V.35, RS-232C, RS-449		
1	Physical			

maps to the ISO/OSI reference model almost identically through the first four layers. After these lower layers have been mapped together, there appears to some minor differences at the upper layers. Since the OSI protocol is so new, there are limited products that are available that follow all seven layers of the network reference model. Most of the effort for products has been concentrated on the upper layers of this protocol. The assumption is that there are adequate products that have been accepted as standards at the lower layers, while the upper layers have been ignored by most networking vendors.

The information presented above gives a better understanding as to how different network protocols fit when compared to the standard OSI network

reference model. The OSI does not provide any better security features than are presently available on existing protocols [10].

2.4 Connecting LANs to other networks

Bridges, routers, and gateways are used to link LANs together and to offer remote users the ability to connect to other networks and use other computing resources. Also, some of the above devices are used to allow different networking protocols to be accessed by the various users. These devices are defined and their effect on network security investigated.

Bridges are usually utilized at the data layer (layer 2 of the ISO/OSI reference model). These devices are primarily used to allow separate networks using the same data layer protocol to be connected and appear as one large logical network [19]. In such a configuration, all network devices are accessible by each other and appear to be on the same network. There are some bridge devices that can be configured to filter and to block network traffic [19]. When a network data packet is sent to a bridge, the address is checked, and if the address is not on the local network, the bridge passes (forwards) the packet to the network. If the data address is known by the bridge, the bridge assumes that the data is for local nodes and does not forward the packet. This filtering method is used to keep local data traffic from being broadcast onto the main network. In most cases, the bridge will forward all broadcast and multicast data packets from one

segment to the next. By definition, these types of data packets indicate that all nodes are to receive these packets; therefore, the bridge will forward them to the next segment.

Routers are usually configured at the network layer (layer 3 of the ISO/OSI reference model). Being at the network layer, these devices have the ability to transfer data to networks that may be using completely different physical media or connections [19]. Routers must share the same types of data and network protocols at the lower levels of the ISO/OSI reference model. Recall that the lower layers are the physical, data, and network layers. Routers can be used to pass several types of network protocols (e.g., DECnet, TCP/IP, etc.); however, the router can only pass one such protocol at a time. Routers do not care about the specifics of the upper-level protocols of the data. These devices review the data packets for valid network protocols and then map the correct destination address and retransmit the data to that node. As long as the lower layers are common, data can be passed, or routed, between network nodes. This routing can be from node to node along the same LAN, or it can be from LAN to LAN. In most cases, routers are used to pass data from LAN to LAN. Routing devices usually have the ability to determine where a packet is to be sent, but since these devices function at the network layer, they have the capability to route packets from different physical network protocols [19]. Routers do not convert protocols; rather, they evaluate the packets to determine where to send them. Routers handle different protocols by being able to break down the packets and

find the destination address [19]. Upon finding the destination address, the router knows where to send the data, at which time the router reconstructs the packets and retransmits it.

Gateways, in most cases, can accommodate any or all of the ISO/OSI reference model. These devices are intelligent in that they have the capability to link networks together and pass data; additionally, the protocols and networks can be completely dissimilar [19]. The gateway performs protocol conversion on the data and prepares a new packet for transmission under the requested network protocol. This conversion is performed by using emulating network communications controllers. This conversion allows network protocols to appear like another network protocol. For example, a DECnet/SNA (IBM System Network Architecture) gateway allows DEC computers on a network using the DECnet protocol to be connected to an IBM host that is using SNA protocol via this gateway. The gateway converts the network protocol from DECnet to SNA or from SNA to DECnet, depending upon the request. Other examples include mail gateways, such as DEC ULTRIX⁹ computers that can convert DECnet to TCP/IP for the ability to move mail messages from TCP/IP networks to DECnet networks. Gateways may also convert file transfer and virtual terminal protocols.

The security implications of these devices are described in this section. The bridge can be configured to provide no data filtering, which could result in

⁹ ULTRIX is a trademark of Digital Equipment Corporation.

all data from LANs being passed onto the main trunk. This could cause a severe disruption of service to all network users. Routers can be configured to impersonate other network nodes and thereby request that all data be passed to the impersonating node. The data can be captured by this impersonating router and then be stored for any number of uses, or the data may be modified and resent to other nodes. Gateways can be configured to convert protocols into meaningless data, modify data headers to send data to the wrong node address, or capture all network data. The point is, all of these devices have certain flaws, and those flaws can be exploited to cause damage to the network and its users.

CHAPTER 3

PROBLEMS WITH LANs

In years past, networks connected dumb terminals directly to mainframe computers. The data could be sent from these terminals to the mainframe computers, but data sent to these dumb terminals could not be stored for future use. There was no local networking, PCs, or workstations, and all data traffic was handled via dedicated point-to-point data links. The development of PCs and workstations has essentially eliminated the need for dumb terminals. These workstations and PCs can do what the dumb terminals did, plus several other functions. These new computing devices provided the capability for distributed computing, which required new networking philosophies. Networks that could transport and process data much faster were needed, as well as networks that could support additional connectivity among new devices. The new, faster network that provided the full connectivity became known as the "local area network."

Broadcast LANs create three security problems: a node may capture all the data on the network by being in promiscuous mode; a node could impersonate another node on the network; and service can be disrupted. Prior to detailed discussion of such security problems, it should be noted that there are two groups

of users: privileged and non-privileged. Privileged users are those who are either system managers on the large systems or system managers (by default) on PCs or workstations. These users have the ability to access peripheral devices and various data files (including system files) and the ability to modify any system parameters. Non-privileged users are those who use terminals or other non-intelligent devices to communicate with the various network nodes. The non-privileged users do not need system privileges to perform their daily job duties. These non-privileged users do not usually manage their terminals or workstations; rather, these devices are slaved directly to the host processor for management purposes. These users, in most cases, are tied into specifically developed application programs that provide the users with only limited capabilities. The users never have the ability to access any other portion of the system. Certain protection mechanisms that apply to one group do not apply to the other, which will become evident in the following sections.

3.1 Promiscuous Mode of Operation

Broadcast LANs present the problem of all network data being "broadcast" along the entire network. On a broadcast network, a node can "listen in" and possibly obtain data intended for other nodes or individuals. This capability is often referred to as operating in "promiscuous" mode. Promiscuous mode can be used from PCs and/or workstations, as well as the mainframes that

are connected to the networks. This promiscuous operation is, in most cases, an optional parameter setting on the network hardware interface unit that connects the computer to the network. The promiscuous ability is set by either software modification to privileged network configuration code or by system software to enable data control bits to be set. If the network hardware interface unit does not support the promiscuous mode, then the possibility of data capture does not exist.

In the past, the computers were managed and operated by a small population of trusted systems personnel who controlled the privileges of how the system was used and by whom. Today, however, with the use of PCs and workstations, the non-privileged user is now his own system manager with the ability to have any and all privileges at any time [5], which is to say that the privileges pertain to his PC or workstation and not to the centrally managed computer. Any users (privileged/non-privileged) can either buy or write software that will allow their systems to monitor and/or capture data that is broadcast on the network [20]. With the ability to view and/or capture data on LANs, there is no longer a fail-safe method to guard against unauthorized access or disclosure of data.

Recalling from Chapter 1 that there is a set of five items that describes the concerns associated with the use of broadcast LANs, the promiscuous mode problem allows the data to be compromised because the data can be viewed by any network user. The privacy of the data is in question because several users may have the ability to capture the data. And the data itself could have been

manipulated by several unauthorized users. The promiscuous mode defeats most attempts to protect the data from other users. In most cases, the promiscuous mode will be unnoticed, because network monitoring techniques are unable to detect such nodes. Once the promiscuous mode is enabled, the entire process is passive rather than active. The threats associated with the promiscuous mode are the capturing of data that may contain information that others should not see or the capturing of a valid user's password which would allow others to gain access to the originator's data files.

If the promiscuous mode is such a problem, why is it supported by the hardware and software vendors? This mode is supported because it allows network managers and systems managers to monitor the LAN. Technicians can utilize this capability to monitor bridges and various hardware interfaces and to help troubleshoot the LAN. The use of this capability must be controlled, because anyone can capture all network data while in this mode.

3.2 Node Impersonation

As mentioned earlier, the ability to fully utilize distributed computers on a LAN gave rise to certain problems. One such problem is the ability for processors to masquerade or impersonate other nodes on the network. This node impersonation may be accomplished in several different ways. In some instances, a valid network address has been made known to an unauthorized network user. Once

this user has the valid network address, this user's node can try to be recognized as a legitimate node on the LAN and obtain data that is intended for the original node. Most LANs utilize network hardware address assignments for node identification. In these cases, node impersonation is more difficult. But privileged users may still alter the hardware address and masquerade as other nodes on the LAN. Publishing schedules of preventative maintenance of valid network nodes can aid in establishing node impersonation. In most cases, preventative maintenance is scheduled so the other computers can update the appropriate network address tables. Usually, network management alerts the various network nodes as to the availability of nodes, thus the network routers are aware of the node not being on-line. This notice could provide the impersonating node with the information needed to impersonate a node on the network. Based upon the above, a node wishing to impersonate must know the valid network address and know when the valid node is not going to be on the network. This allows the impostor to connect and inform the other nodes the valid node is back and ready to process data. Different network protocols will respond differently once two nodes with the same network address are recognized on the network.

Another type of impersonation can be accomplished via a network router. If a privileged user gains access to a network router that is also a general-purpose processor, this computer can become some other node on the network. This privileged user may reconfigure the network such that all data packets will be routed through the impersonating network router. Then the router may create

its own data stream and insert another node's network address into the data stream and thus impersonate yet another node.

Node impersonation can permit access to other nodes through non-authenticated services. Routers and gateways can be used to impersonate other valid routers and gateways. In one case, nodes can become false network routers or gateways and request that all other nodes use this false node as the network gateway or router. The other case is that real, valid network routers and gateways could be configured to impersonate other network nodes. Since all data is passed through these routers and gateways, it could be easy to obtain valid node addresses and node names.

Various network protocols handle the problem of two network nodes appearing on the network at the same time differently, and a detailed description of what actually happens is not addressed here. However, a brief description of how two different network protocols handle the situation of two nodes on a LAN at the same time is presented in Chapter 4. The description provides some insight as to how difficult it is to actually catch an impersonator on a LAN.

There is another type of impersonation called "service" impersonation. Service impersonation results from users accessing one of several different types of software utilities that are provided by the operating system or application software. In most cases, software utilities require no special privileges, because the PC or workstation user controls all such utilities. Such utilities allow users to send files containing data to other network nodes. These files may contain

the user's password or access controls to other files and data that may reside on remote network nodes. However, a privileged user on a large processor can develop and implement a fake application software program that emulates many system utilities. Thus, the individual user's remote access is subject to capture. This remote access may include the user's data files, passwords, or other unique data that pertains to the user.

It is clear that nodes can impersonate other nodes, but what harm can come from this? The impersonating node can forge mail and send/receive data to or from other network nodes. This impersonating node can also read mail that the other nodes think is sent to the real node. The impersonating node automatically obtains all privileges that the other node had. If the valid node is allowed access to other network nodes, then the impersonating node also has this access. Such access may allow copying of files or sharing of files, and it may also allow the other node to log in to the other nodes. Based upon these threats, the impersonating node can be used to compromise data, cause the loss of data integrity, and allow unauthorized users to have access to data. Also, impersonation makes it extremely difficult to ensure the authenticity of both the sender and receiver of data.

3.3 Disruption of Service

The main concern about disruption is denial or loss of service to any or all components that are part of the LAN. This denial or loss of service could be intentional or unintentional. Disruption of service can result from:

- physical damage to the transmission media,
- damage to physical network components (such as bridges or transceivers),
- damage to communications hardware associated with the computer, or
- flooding the network with spurious data.

The disruption of service on LANs can be caused by several different factors. The network media may be cut, which will result in lost network service; network hardware may become damaged and may require repair; various communications interfaces may become damaged and require repair; or users may flood the network with meaningless data which may drastically affect total data throughput on the LAN. Although the denial of service may seem to be nothing more than a “glitch” on the network that never really causes any damage to any portion of the LAN, this “glitch” may result in the users of the network being required to retransmit data. And, as previously discussed, retransmission is one method used to allow some other device to “listen” or capture network data. Examples of network glitches include unsuspected network failure, unusual network

error messages, or users of requesting data wondering why the data never arrived. Therefore, any and all network problems must be taken seriously and mechanisms must be in place to detect such problems before it is too late. Such mechanisms are discussed in detail in Chapter 5.

CHAPTER 4

PROBLEMS OF SPECIFIC LAN PROTOCOLS

4.1 Promiscuous Mode

4.1.1 TCP/IP and the Promiscuous Mode Problem

The Internet protocol suite is used on broadcast LANs. The main concern is one of privileged users (system managers or non-system managers with elevated privileges) having the ability to place the computer into promiscuous mode. Usually, processors can be placed into promiscuous mode by using software code to enable certain control bits to be set. In most cases, the user can either develop code to change these control bits or utilize system routines to change these addresses.

With TCP/IP there exists various network interface devices that can be modified to enable the promiscuous mode. One example on SUN¹⁰ systems allows the “superuser” to utilize software code, known as Network Interface Tap (NIT) protocol, to capture data packets from the Ethernet. This code is a software product that is layered on top of the kernel operating system. There are two parameters that can be set when using NIT. The first allows

¹⁰ SUN is a trademark of SUN Microsystems, Incorporated.

for normal filtering, which only receives data packets that are intended for a specific processor (based upon the network address); the second is promiscuous filtering, which receives all data packets. Of course, users also have the ability to change existing software that controls the Ethernet hardware interface (This was pointed out in Chapter 3). Users can either develop software or use system routines to change the Ethernet hardware.

Another example is software that specifically controls the 3COM Ethernet interface typically found on PCs. This software, known as the Ethernet hardware configuration control program, configures the network ports on the hardware device and also contains information on the Address Resolution Protocol (ARP) address, along with the valid Ethernet address. This software is protected from the non-privileged users on large computer systems. But again, if the computer system is a PC or workstation, then the user is the system manager. Under this condition, these types of users can change, modify, or develop their own software to enable the promiscuous mode. There are other software programs that can be used to initialize the Ethernet hardware devices and connect such devices to the systems input/output controllers. The users of large systems cannot access these routines, but the users of PCs or workstations can change these routines. The users on TCP/IP networks, specifically users of PCs and workstations, can set their devices in the promiscuous mode without great difficulty. Because users of these PCs and workstations control their systems, the ability to track exactly what has been done is difficult at best.

As mentioned previously, the privileged users (such as system managers) have the privileges necessary to assume control on a particular processor for which these privileged users are responsible. Therefore, if a particular processor accepts user accounts and is also used as a router for the network, the situation exists for possible compromise of data and resources. These privileged users can capture all data that is being sent through this processor to other network nodes via routing tables. This processor could be used as the router for local LAN traffic or it could be the router from one remote network to another. This particular router may also be established as a valid and legal network router; however, a closer look indicates that non-privileged users who access the processor are not able to perform any special or privileged operations. This is exactly what should happen. No privileged access is provided, thus the integrity of the computer remains intact. Now when a user with elevated privileges (privileged user) does access this processor, the entire computer (all data, resources, and associated components) is subject to compromise, manipulation, and loss. Because a privileged user can access this computer, nothing on this processor is secure. All data that is routed by this processor is subject to compromise.

To summarize, there are two major areas of concern: promiscuous mode and data capture via routers. Both of these are problems that require additional thought and attention when a network is initially designed. This data capture problem relates to the third item on the list of general concerns (the ability to protect data from compromise) provided at the beginning of this chapter. As

previously described, it is somewhat difficult to protect data from compromise when network interface equipment and various hardware devices provide the capability for promiscuous mode and data capture.

4.1.2 DECnet and the Promiscuous Mode Problem

Several interface devices are available from either DEC or other third-party vendors that can interface the computer to a broadcast LAN. There are several different types of these devices, but two such devices from DEC are presented in the following sections. These two devices are the DELUA (Digital Ethernet Large-scale-integration Unibus Adapter) and the DEUNA (Digital Ethernet Unibus Adapter). Discussion focuses on how such devices can be configured in promiscuous mode. Detailed analysis for configuring these devices and modifying them to capture data from the network is explored in this section.

The DELUA and the DEUNA adaptors can be used on any DEC computers that use a Unibus-based backplane for connecting peripherals to the main CPU components. These devices are used in an Ethernet bus topology and, in most cases, the DECnet protocol is used. As noted in Chapter 2, however, other network protocols can also be accommodated with these devices. These hardware interfaces can be programmed to receive all data packets on an Ethernet (a broadcast LAN). While these devices are connected to the computer's bus architecture, the computer's operating system actually controls the devices

on the bus. Privileged programmers, system managers, or any privileged users (exact privileges discussed later) who know the internals of the computer can develop software programs that will enable network nodes to receive all network data packets. Privileged users can also utilize system routines that already exist to accomplish the same goal.

The DEUNA manual [21] and the DELUA manual [22] contain sections that deal with which particular bits need to be enabled to cause these devices to be placed into promiscuous mode. These manuals are intended to give examples of how the privileged user could develop software to set the appropriate bits to establish promiscuous mode. A privileged user can enable the promiscuous ability of these network interfaces by utilizing specific operating system routines. These routines in the DEC VMS¹¹ operating system are known as Queue Input/Output (QIO), and these routines are part of system functions that allow the systems managers or other privileged users the ability to fine tune the system for better performance or to allow special systems software to be designed and implemented for specific purposes. If the privileged user knows what to do and has experience in using QIO routines, the entire system “belongs” to this individual. In particular, the experienced privileged user would need to utilize the **SET MODE** operation that directly interfaces to the network hardware. In most cases, the privileged user would use the command **IO\$_SETMODE! IO\$M_CTRL** from the QIO systems services routines, which indicates that the network interface is

¹¹ VMS is a trademark of Digital Equipment Corporation.

to have certain characteristics set. To enable the promiscuous mode, the privileged user would issue the command **NMA\$C_PCLI_PRM**, which identifies the promiscuous mode. After this, a choice is made to enable or disable the promiscuous mode capability. If the choice is to enable promiscuous mode, the command **NMA\$C_STATE_ON** is used; if no promiscuous mode is desired, the command **NMA\$C_STATE_OFF** is used. The default for this particular capability is promiscuous mode disabled. In order to fully implement and modify the parameters associated with these network hardware devices from a software program, the user must have as a minimum **PHYSICAL I/O** privilege. Any of the other elevated system privileges, such as **SYSPRIV**, **CHMKRL**, **SET-PRIV**, to name a few, can be used to assign the physical I/O privilege. The example above assumes that the user is taking advantage of the system routines that are provided as part of the operating system. Next, discussion focuses on the privileged user who wishes to develop and use his/her own system software to enable or disable certain features.

As indicated at the beginning of this section, users can ignore the existing system software routines and use their own software to interface with system devices. The privileged user may choose to develop and implement his/her own system routines to enable or disable certain system features with external devices. In this case, the user would still need the necessary system privileges and must also know how develop particular system routines to implement the code that was written. The user could use the specific hardware reference manual,

which usually provides information as to which registers and which data bits need to be addressed when developing specific system software. In most cases, the privileged user would develop software that would tie directly to the Port Control Block (PCBB) [21], which resides within the internal memory of the computer. This control block directly controls the network hardware interface. In this particular case, this is either the DEUNA or the DELUA. The PCBB has information that relates to the port function of the device [21].

4.2 Node Impersonation

The following discussions pertain to the problems of nodes on these “broadcast” LANs that can be made to look like another (valid) node on the network. If a node can indeed appear to be a different node at any point in time with success, then the vulnerabilities that accompany such are quite significant and must be addressed. Contained herein are discussions on the various network protocols and how the impersonation problem relate to each. Again, the network protocols discussed are TCP/IP and DECnet. The OSI protocol is discussed later in this chapter. The impersonation problem is by no means limited only to these specific protocols. In fact, this problem is probably widespread among many network protocols.

4.2.1 TCP/IP and the Node Impersonation Problem

Computers on a LAN that use the Internet protocol suite (specifically, TCP/IP as a communications protocol) usually follow a set of simple controls. Computers using TCP/IP may generally follow any network topology. For this example, discussion centers on computers using a bus topology Ethernet configuration with the TCP/IP protocol. All Ethernet addresses are unique, which is to say that each interface hardware board is assigned a distinct and unique address. In this case of computers using TCP/IP on an Ethernet, the Ethernet addresses are obtained via the Address Resolution Protocol (ARP). This ARP is not part of the TCP/IP protocol. Each computer on the Ethernet maintains an ARP table that contains known Ethernet addresses for systems on the network. The ARP table is used to map the Internet network address to the Ethernet address. If a host needs the Ethernet address of another node then the host computer broadcasts an ARP request asking for the valid Ethernet address for the system in question [23, 24]. For example, a system with a network address of 149.0.4.200 is trying to address a system with a network address 149.0.4.100. As discussed previously, the source node will verify by looking up the correct Ethernet address for the network node of address 149.0.4.100. The 149.0.4.100 is known as the TCP/IP Internet address. The source uses the ARP table to find the related Ethernet address for the node 149.0.4.100. If the Ethernet address is found, the data packet will be sent to node 149.0.4.100.

If the Ethernet address of 149.0.4.100 is not in the ARP table, the source node sends out an ARP request in the form of a broadcast message asking for the valid Ethernet address. By design, every system on the network must listen for such broadcast data packets [23]. Therefore, every computer on the Ethernet will see the ARP request. All the computers will then determine if the ARP request is for their own Internet address. If the Internet address is for them, the computer will return a response to the source node; if not, they will ignore the request [23]. Now that this has been accomplished and the systems are conversing, the ARP watches in a passive fashion for any duplicate addresses which may appear. Duplicate addresses almost always indicates that a system is impersonating another node. If a duplicate address is found by the ARP, a monitoring capability of various alarms and logged entries will be made. These logged entries will be recorded at the requesting node console, as well as on the duplicate node console. Therefore, from the log the duplicate address will contain an Internet address and a unique Ethernet address for that node. Thus impersonators may be discovered. All systems using this scheme have an ARP table that keeps a valid Internet address and Ethernet address for each node on the network. Even if a node duplicates the Internet address, the new entry in the ARP table will have the unique Ethernet hardware address associated with the duplicate address associated with it. This will be logged at the console along with other data pertaining to the duplicate address that tried to access the LAN. The logging of such information will make it more difficult to impersonate

nodes on the TCP/IP network because the Ethernet hardware addresses are not altered.

Nodes on TCP/IP networks, whether they be PCs, workstations, or large computers have the ability to change their IP network address. Privileged users can modify the system startup file that contains the network name and associated IP address. In this manner, the privileged user can change node names and reboot the node. Thus the privileged user can configure his/her node to impersonate a valid network node by using this technique. Even if the original node comes back on the network, the impersonating node will continue to receive data. This is based on the workings of the ARP table. The ARP table maps the IP address to the Ethernet hardware for the node. The impersonating node has the valid Ethernet hardware address, and the ARP passes this information to any other nodes. The original node, when requesting access to any network node, will be sent to that other network node, but data sent from the other node will be sent to the impersonating node. No data will ever reach the original node. The original node will at some point realize that no information is being received and will initiate action. This action will be to notify the TCP/IP network manager that there are network problems; however, by the time the impersonating node is discovered, data has already been compromised.

There are some specific capabilities that exist within the TCP/IP protocols that could be exploited with impersonation. Such capabilities include the Network File System (NFS), the hosts equivalents (hosts.equiv), the remote hosts

(.rhosts), and others. These facilities use the Internet address as a method of authentication. The NFS allows other computers and associated users to access centrally controlled disks. The hosts equivalents are used by systems managers to establish a list of hosts that are to be trusted, and no formal authentication is required for access. This information is kept in a table that is protected from other users. The remote hosts is a list that a user maintains of machines that can be trusted to access each other without any further authorization being required. As seen from these examples, if a node is able to impersonate a node that has any of these features enabled or has access to any other node, all data on these nodes is no longer adequately protected. However, some Ethernet hardware permits privileged software to alter the Ethernet address. This condition is presented in the following section and indicates how DECnet operates.

4.2.2 DECnet and the Node Impersonation Problem

In the second example, computers that utilize the DECnet protocol on a LAN using a bus topology follow the same basic description given previously. The data is still broadcast on the LAN and all other nodes have the ability to capture it. With DECnet the routing concept is somewhat different. In DECnet, the data is handled by routing techniques that are used to route the data to the correct network node. But the data is still sent out on the LAN in broadcast form and, while it is intended for a specific node, it can still be captured. The

DECnet routers maintain tables that have valid Ethernet addresses for all nodes. The information in these tables deals with node names and node addresses. This network data is maintained in a DECnet configuration database which is part of the DECnet software. Privileged users can access this DECnet software and modify the entries in this database. These privileged users can change the node names and node addresses to any other network node. Thus, node impersonation via these changes is possible if the privileged user knows how to initiate and modify this table. Once the privileged user has modified this network table and to make the change become permanent, the node must be removed from the network and then rebooted back online, and then the changes to DECnet will be recognized throughout the network.

Users who utilize PCs or workstations have absolute and total control over their systems. In these cases, users of these systems may not fully understand that by changing or modifying these network nodes and addresses that they are affecting any other nodes or devices on the network. The DECnet router reviews the Ethernet packet and looks at the Ethernet destination address to see if the data should be processed or sent on. The difference between TCP/IP and DECnet as they pertain to Ethernet is that TCP/IP does not allow software to alter the Ethernet hardware address, whereas DECnet does allow privileged software to alter the Ethernet hardware address to encode the area and node address needed for the DECnet protocol. Based upon this ability to change the Ethernet hardware address, the detection of invalid nodes is more difficult.

As already indicated, the actual Ethernet physical address is assigned by the Xerox Corporation, so DEC allows the network interface boards the ability to assign a unique address that is appended to the original Ethernet physical address. For example, an Ethernet physical address of AA-01-23-45-67-FF contains two portions of information [25]. The AA-01-23-45 is the Ethernet address that is unique to DEC, and the 67-FF is the Ethernet address assigned by DEC that details the network interface board. These two addresses make up the full Ethernet address for a particular node on the LAN. In most cases, the user has no need to alter the Ethernet address directly because the computers themselves construct their own unique physical address for the Ethernet address [25]. That is to say, if DECnet is not used as the network protocol, there is no change to the Ethernet address; but if DECnet protocol is used, this Ethernet address is changed. Once DECnet is installed and loaded on a node, the entire 48-bit Ethernet address is redefined to indicate the unique DECnet node address. This DECnet node address is established from the System Generation (SYSGEN) parameter SCSSYSTEMID [26] at system startup or when the system is booted and brought online. This SYSGEN parameter contains a constant DECnet header which is always AA-00-40-00, which is the first four bytes of the Ethernet address, but in this case, this defines the network node as using DECnet. The last two bytes contain unique DECnet area number and node number. This is how DECnet knows where to route data in order to pass information from node to node. The information that is passed from node to

node as part of the DECnet routing data also establishes paths and circuits to other nodes on the network.

There are other ways that the Ethernet address will change if, for example, the LAN is experiencing difficulties with rapid node growth, hardware failures, or other such problems. New Ethernet addresses can appear on the network because, in most cases, hardware is removed and added at will. So one valid Ethernet address that was used may change due to hardware problems that require new hardware or a new network address. Once the DECnet network is halted on a particular node to allow new components to be added or removed, the Ethernet physical addresses may change. As described in the following paragraphs, privileged software does exist that allows the physical portion of the Ethernet address to be altered.

The DECnet protocol needs to know the source, the destination address, and the area and host numbers of the nodes on the LAN for purposes of area routing. Thus, by design, the DEC computers, specifically VAX¹² computers using DECnet, have software (privileged) which allows the Ethernet address to be altered to include the area and host address, which the router needs to utilize as part of the DECnet protocol. This altered address, as already indicated, is the physical address that contains data about the physical interface and node that is on the LAN. Not every user on the computer can alter the physical address, but system managers have the necessary privileges to alter this physical address.

¹² VAX is a trademark of Digital Equipment Corporation.

As explained in the previous section, methods exist that the privileged user can utilize to modify certain features of the hardware device. These methods include using existing system software routines (known as QIOs for VMS) to change or enable capabilities of these network devices. As before, if the privileged user wishes to utilize existing system software via the QIOs, the user would need to be familiar with the QIO commands that allow the Ethernet address to be changed. This particular QIO has the same initial command as was used for the promiscuous mode discussion. Referring back to the promiscuous section for a moment, the reader will recall that the SET MODE section of the DE-UNA and DELUA present callable system QIO routines for enabling features of these network devices. Changing or modifying the Ethernet address (all 48 bits) can be done by using the following QIO. The user issues the command **IO\$_SETMODE!IO\$M_CTRL**, which indicates that the controller portion of the network device will be changed. After this command, the user issues **NMA\$C_PCLI_PHA**, which is the parameter that controls the physical address for the device [26]. Once this QIO is issued, the user must decide what action to take next, as there are three unique parameters that exist and the user must choose one of them. One of these parameters allows the new 48-bit address to be set by **NMA\$C_LINMC_SET**, the other parameter allows the entire 48-bit address to be cleared by **NMA\$C_LINMC_CLR**, and the last parameter allows the default DECnet address to be set by **NMA\$C_LINMC_SDF** [26].

As before, these are examples of how the existing QIO system routines could

be used to change the 48-bit Ethernet address. But an alternate method exists that the privileged user can utilize. As stated in the promiscuous mode section, the user can ignore the existing QIO system routines and develop unique software that can be used to change the 48-bit Ethernet address. The user will still need to have elevated privileges to perform the necessary functions that allow this 48-bit address to be modified. The user would develop software that interfaces with the PCBB, specifically PCBB function codes 4 and 5, indicating that the physical address (48 bits) is to be modified [21]. Several different combinations can be set to enable or disable certain address capabilities. Some of these capabilities include the ability to read the default address, to write the default address to the device driver, and to change the physical address.

There are third-party vendors who manufacture components that can be configured to utilize the DECnet protocol. These components emulate DEC hardware and, because of this, the privileged software allows these emulators to alter the Ethernet address and to use all the functionality of the DECnet protocol.

Once the source and destination computers recognize each other, a "dedicated" link is established among them on the Ethernet channel, and the node that is impersonating the legitimate node may never be discovered. Although the network will realize by the use of network monitoring tools that there may be two nodes with the same DECnet destination and source address, data will continue to be placed on the Ethernet.

If a node on the DECnet network could impersonate another node, what damage could be done or what other data could this "new" impersonating node have access to? Whatever the original node had the authority to do, this impersonating node can also do. If proxy accounts were used on the network to gain access to this particular node, then the impersonating node could utilize proxy accounts also. The use of proxy accounts allows users to access remote nodes and files without using a password. To copy files without the use of a proxy account, the syntax is `$ COPY node_name"uid password": file_name.ext *.*.` The key element here is the data in quotation marks: the user ID and associated user password. This data is in "cleartext," and users with sufficient privileges can "listen" on the Ethernet for this type of data. With proxies, the same copy command has the syntax `$ COPY node_name::file_name.ext *.*.` The difference is that no user ID or password is present; therefore, the data cannot be compromised. The proxy account is added to the specific user authorization files (UAF). The system manager on host B runs the system authorization utility procedure and adds the proxy information to the user files. The syntax for adding the proxy capability is [27]:

- **SET DEFAULT SYS\$SYSTEM**
- **RUN AUTHORIZE**
- **UAF> ADD/PROXY host A::FRED DEC**
- **UAF> EXIT**

The proxy account, as seen in the above example, must know three things about the user [27]:

- the remote nodename,
- the remote username, and
- the local username.

From the example above, the remote nodename is host A, the remote username is FRED, and the local username is DEC. Based upon this, anyone who is logged in as FRED on host B can access through DECnet over Ethernet the account of DEC on host A without providing a password. While the use of proxy accounts is good and stops the compromise of "clear text" passwords, there are problems that allow the impersonation of the network nodes. Using the example from above, if host A is removed from the network for maintenance, a privileged user of another network node can modify his/her DECnet nodename and node number to be that of host B. This privileged user could then add a known proxy account (in this case, FRED) on his/her host, for he would know the password. Then this privileged user could log in as FRED to his/her host and gain access through the network to the remote host (in this case, host B) to which DEC has access. For this reason, the use of proxy accounts with privileges should never be allowed.

If the network nodes used this particular node for special services, like file service and print service, then all data would be sent to this impersonating node

as well. Thus the integrity of the data is no longer valid. The data itself may be compromised, and the ability to protect data from unauthorized access can no longer be accomplished.

The impersonating node also has the capability of going into promiscuous mode, which raises a double concern. Not only can this node address other processors while impersonating a node, this same node may now also listen to all data that is being broadcast on the Ethernet.

4.3 Disruption of Service

Earlier in the thesis, reference was made to several areas that were cause for concern in the disruption of service on LANs. Some of the areas that can cause disruption of service are out of the user's control. These would include the loss to network components, loss of communication hardware, or, in some cases, loss of the transmission media itself. But if users wish to cause damage to the network by using the items below, a great impact on the network and its users would result:

- loss by physical damage to the transmission media,
- loss to physical network components (such as bridges and transceivers),
- problems with communications hardware related to the computer, or
- flooding the network with bogus data.

There may not be any unique mechanisms that can be used for any of the specific LAN protocols that have been discussed throughout the thesis, but there are some methods that can be implemented to combat this particular problem. Some discussion will be directed toward the different types of LAN protocols that have been presented, but it is worthy to note that this particular problem is not restricted to certain LANs. The methods used to cause disruption of service may be different, but the result is the same.

4.3.1 TCP/IP and the Disruption of Service Problem

As mentioned previously, the privileged users (such as system managers) have the privileges necessary to assume control on a particular processor for which they are responsible. Therefore, if a particular processor allows user accounts and is also used as a routing node, which in TCP/IP is how most messages are distributed from node to node via a router, disruption can become commonplace. The routing node is a very powerful device and, as discussed, can be used to capture network data, or it can be used to cause disruption of service for the entire network. These privileged users can capture all data being sent through this processor to other network nodes via routing tables. Such users can also flood the network with bogus data. This type of attack may not be as damaging as destroying data files or capturing data, but the side effects are worse in many cases. If the network is so bogged down with bogus mail or network traffic,

there is no way that valid data can be transmitted and processed properly. While this type of disruption is more apt to happen on small LANs, because they are easier to obtain access to, one must remember that LANs are usually connected to other LANs via other routers. Based upon this, these processors could be used as network routers for not only local LAN traffic, but also could be the router for other networks. The point needs to be made again that when a user with elevated privileges (privileged users) does access this processor, the entire computer (all data, resources, and associated components) is subject to compromise, manipulation, and/or loss. Because a privileged user can access this computer, nothing on this processor is secure. All data that is routed by this processor is subject to compromise. This could allow any or all data to be viewed, stored, modified, and deleted at the direction of the user who now controls the computer.

Non-privileged users of nodes on TCP/IP networks can also disrupt service on the network. These types of users can consume all network ports to the network nodes. This can be accomplished by using the File Transport Protocol (FTP) to transport files to all nodes on the networks. In TCP/IP, FTP is configured to allow access to all network ports, and users can cause the entire network service to be affected. These users could flood the network with meaningless data, which would require all TCP/IP nodes to respond. Thus, all nodes would try to respond at the same time. This would cause several collisions on the network and service of the network would be affected. These users could

utilize electrical devices that may cause the electronics of the network to fail. The ultimate service problem is physical damage to the cable.

4.3.2 DECnet and the Disruption of Service Problem

The major condition that can disrupt DECnet traffic on a broadcast LAN is the flooding of data to all known nodes on the broadcast LAN. Such a flooding would require that all the receiving nodes send data back an acknowledgment that the message was received. If all went as planned, all the responses sent on the network would drastically reduce the data throughput because all nodes would be responding to the initial message and flood the entire LAN with data.

There are several other ways in which the network service could be disrupted. If a privileged user gained access to the DECnet node that was used for the network router, the damage that could result is uncontrollable. This routing node could be removed from the network, in which case certain traffic may not be transferrable. The routing node could change or modify the network routing tables that are part of the DECnet software, which could redirect traffic to different DECnet nodes, or the traffic could be removed from the network before it has a chance to reach its destination. This idea was presented initially in Section 4.2.2. This concept indicates that the LAN problems cross over into several aspects of network control. If not properly managed and controlled, a network could change in size by the addition or deletion of nodes on the LAN

and thus affect the entire network. If nodes or devices are allowed to connect without any control, the size of the network will contribute to factors like slow throughput, inability to find the correct routing nodes, and the possibility of duplicate node names and addresses.

4.4 OSI and Various Security Problems

As indicated earlier, most attention was directed to the TCP/IP and DECnet protocols because of their popularity and current utilization as the standard in LANs today. Discussion now is directed towards the OSI protocols and what they are going to offer to the LAN technology.

From various sources and technical publications, it appears that all the OSI products are being developed for the upper layers of the ISO/OSI reference model. Specifically, the OSI products are being developed for users to utilize several different application types of software. This will allow greater flexibility since the type of network the users are on will make no difference. Once fully implemented, networks that adhere to the OSI network standard will be able to share information among all types of networks. This ability to share information will not be dependent upon common hardware or software but, rather, upon the implementation of the OSI. Table 3-3 depicts the OSI as it relates to the ISO/OSI reference model. The only real difference is that the upper three layers ("Session," which is layer 5, "Presentation," which is layer 6, and "Application,"

which is layer 7) provide several different methods which allow users to pass data from node to node. The lower layers follow, or can follow, previously developed and established standards for access. The Physical and Data layers, both referred to in Table 3-3, have already been developed and, in most cases, are used today. This entire discussion assumes that no attention has been given to protecting data. These newly developed and designed OSI products must provide greater protection of data or no control has been obtained. By utilizing various application-specific products to pass data on LANs, there may be less concern about network-specific protocols. Thus, a distributed network will be truly distributed across many different types of networks. These networks will be using several different types of protocols, and the users will be able to exchange data among all these various types with little or no problem. But the fact remains that as long as there are broadcast LANs, the ability to "listen in" or to impersonate another node will exist.

4.4.1 Summary of OSI

There are several possible reasons for efforts being concentrated on the upper layers of the ISO/OSI reference model. One is that all the necessary standards for the various network components and control protocols are already established at the lower layers. Such standards include all the physical methods for networks, such as Ethernet, token bus, and token ring, as well as others. The other layers

of the ISO/OSI reference model also have settled and are being established as the standard for networks. The OSI protocol will be taking advantage of ISO/OSI network standards, and, thus, OSI-specific products will be able to be implemented on most computers. The entire process and purpose of the OSI strategy is to allow different computers to interact with each other [28]. Today, this ability exists within various test facilities, and there are a few such applications in use in the customer environment. One such application is the use of a gateway that allows IBM and DEC computers to access each other. This gateway converts IBM network protocol to DEC network protocol so the DEC computer can understand what was sent. The gateway also converts DEC network protocol to IBM network protocol so the IBM computer can understand what was sent to it. There are several vendors producing these types of gateways for use between various computer systems. This is a small piece of the total picture. In the future, OSI standards will allow (as the ultimate goal) all types of computers to access each other without the use of gateways to convert network protocols. When the OSI standards are finally in place and the ability exists to access several different types of computers via one standard network protocol, the ability to protect information will be much greater than it is now. At present, the OSI standard for security is only concerned with providing protection of the communications route and the data that is transmitted [28]. The security on the computer system is not part of the OSI standards, but that type of system security is still very important. The system security is the basis that is used

to protect the data while it is resident on the computer. Several OSI security methods can be and usually are combined with existing security controls to provide greater security and control. Some of these security controls are [28]:

- encipherment (encryption),
- digital signature (electronic),
- access control,
- data integrity,
- authentication,
- traffic control,
- routing control, and
- notarization.

From this list it is interesting to note that several of the items are used on computers today to provide protection to data on the systems. Further discussion on the above items will provide some support as to what each item can provide in the way of additional protection. These items are part of the ISO/OSI security architecture for the interconnection to other systems. Many of these items cannot be utilized on all types of computer systems, but these items do provide a foundation that should be considered for all systems when they are being designed and built.

The use of encipherment (encryption) can be used to provide additional protection against users trying to access the networks or systems and users trying to modify data. It can also provide data integrity. Encryption is discussed in detail in the final chapter as part of solutions to LAN problems. The point to understand here is that encryption can be used to enhance the total security of the network/system [28].

Digital signatures can be used to protect a certain type of data or they can be used to allow certain users to access specific data elements. In most cases, digital signatures are used to allow a specific set of users to create data and send this data to designated recipients. In this type of arrangement the data is encrypted and the keys for encrypting/decrypting are known only to the authorized users for the digital signatures. These encryption algorithms allow both the sender and receiver of data to know exactly where the data came from and who created it [28].

Access controls are used to limit access to resources to those individuals who have been authorized for such access [28]. Some types of access controls could be unique user passwords, access control protection on both files and directories on the system, and the establishment of access rights to data. In most cases, computers can accommodate the use of access controls with very little changes to system configuration.

The concept of data integrity is closely related to encryption, but the use of checksums can be helpful to ensure that data has not been altered. Also, oper-

ating systems can set specific data/control bits to help provide data integrity. Certain checks can be performed on data as it is received from the network; these checks are performed before the data is stored or used by the user.

Authentication exchange refers to the use of passwords between systems. These passwords are used to identify users to remote systems. The use of passwords will not protect against individuals who are intent on doing damage to the system. The use of mutual authentication could also be used to ensure that each system knows who is accessing whom.

Traffic padding can be used to help reduce the intermittent data traffic on the network. If the network is successfully tapped (passively/actively) the unauthorized network listeners can learn what type of data traffic is on the network. And if a pattern develops, these unauthorized users could analyze the traffic, wait for network transmissions to cease and then assume control of the network. Some systems can pad the data packets to a constant size so such unauthorized monitoring will not be able to obtain data on traffic flow [28].

Routing control can be used on some systems to control the routing of data. This control can be used to ensure that data is sent/received over trusted network routes. These routes are established for the sole purpose of sending/receiving data that is sensitive or data that requires additional protection.

The use of notarized control is based upon some third party being used to valid, authorize, and authenticate users. This type of control can be used provide data integrity, protection of data, and various other information that can help

to log data on users and systems or other aspects of network use. Detailed discussion on such a notarization technique is presented in Chapter 5.

It is apparent from this discussion that the OSI standards are going to be the standard for all future computer components; however, it will take a significant amount of planning and migration to implement this new network protocol. There will be no choice for some in this matter. It has already been announced that all U.S. government agencies are required to implement the Government Open System Interconnect Profile (GOSIP) in the near future. The GOSIP standard is based upon the OSI protocol and is intended to allow all the various government facilities to communicate directly with each other. The concept is to allow distributed networking but to no longer have the need for dissimilar networks using dissimilar protocols. A significant amount of time is needed to properly plan and migrate to the GOSIP, because many facilities or agencies will be phasing out specific networks and protocols while phasing in GOSIP. There is no real difference between GOSIP and OSI, except that the U. S. government wants to make the OSI protocol internally mandatory. There are other underlying problems related to the implementation of GOSIP. All networks will be able to communicate with each other over many different types of network medias. This poses a severe problem for facilities that are dependent upon proprietary networks in order to function. The problem is that it may be difficult to obtain a waiver for *not* using GOSIP.

It is evident that OSI is going to be used and will, in all likelihood, become the standard for networks. Based upon this premise, what are some products that are (or soon will be) available for use with the OSI protocol? There will be great emphasis placed upon the capability to exchange mail messages among all network users; therefore, initial work is being focused on Message Handling System (MHS); File Transfer, Access, and Management (FTAM); Remote Operations Service Elements (ROSE); and Common Management Information Service (CMIS), as well as other network management control products. Recent updates to the OSI protocol stack indicate that all of upper layers (4, 5, 6 and 7) have been finalized or are in the process of being finalized [29].

CHAPTER 5

SOLUTIONS TO LAN PROBLEMS

Up to this point, all discussions have centered on the various problems associated with LANs. As identified previously, these problems are promiscuous mode, node impersonation, and disruption of service. Discussions of these problems focused on how they can be exploited and how possible damage to data may occur. The following sections provide information that may be helpful in combating inherent LAN problems.

Recalling from Chapter 1, Table 1-1, the set of network protection features, the following discussion focuses on each of those protection features. The first entry is terminal access, which involves controlling and restricting access to the terminals. All terminals should be logged off by the user, and these devices should be powered off when not attended.

The second feature is establishing valid account numbers and passwords for all users. All accounts and passwords must be unique to the individual user. Users should not be allowed to change their own passwords. In most cases, users tend to pick trivial passwords that can be guessed easily by other users. Policies and procedures should be developed to control the generation, distribution, and updating of all passwords for all users. All passwords for users must be randomly

generated and must be controlled. Also, some form of encryption must be used so that the entire password file can not be displayed from the system. Associated closely with all of the above is the caveat that all individual system access be suspended when any user terminates from the project or company.

The third protection feature is utilization of unique passwords to protect files. This feature helps to protect each user's files from attempted access by all other users on the computer. Unless other users know the password, they cannot get access to another user's files.

The fourth protection feature is file access restriction. The restriction allows certain users access to another user's data files. But other users who are not part of the privileged access group are not able to access the data. This is similar to an authorized user group that has been granted access to other users' data.

Network user verification is the fifth protection feature. This technique requires the user to be verified as a valid user on the network. A network controller contains unique data on all valid network users. This data is encrypted to provide protection in case something happens to the controller. Such data may be the user's account number and password. The user logs onto the network controller. The controller compares the user's data with the data in its tables. If the data matches, the user is allowed access to the systems on the network. If the data does not match, the user is denied access to the network and the controller records this information in a log.

Hard-wired terminals make up the sixth protection feature. Under this tech-

nique, all network terminals are physically wired to specific computers on the network. Access for users is provided by dedicated point-to-point links. Terminals are dedicated to only one computer system, and users are authorized for access to only this computer.

The last protection feature is encryption of all data files on the computer. There are several different types of encryption that can be used to protect data as it goes from computer to computer via the LAN. Examples of these encryption methods are presented later in this chapter. There is no doubt that the encryption of data provides the greatest protection of the data, but in most cases, the cost to encrypt prohibits its use. The cost includes hardware components, time required to transmit/receive encrypted data, and ability to manage the encryption equipment. But as techniques and technology advance, the ability to encrypt will become more commonplace.

Several different philosophies are being touted as the "fix" for such LAN problems. Some experts say the only true measure to protect data is to encrypt every data element from the moment it leaves the sending node until the receiving node obtains the data. While this type of solution may be acceptable for some environments, it may not be acceptable for all. Therefore, there must be alternative solutions that can reduce the risk, while still allowing users to utilize the distributed network concept. Some of these other alternatives include administrative control over network connections, physical protection of computing equipment, and education for all users of computing resources. There may be

other alternatives that can be used to protect information besides encryption. Some of these other methods are presented and discussed later in this chapter.

In the first section, attention is focused on various techniques that can be used to provide general protection of LANs. Subsequent sections are devoted to discussing other methods that can be used, like encryption, to protect information from the problems and concerns presented in Chapter 1. There are groups of computer scientists who either have designed or are in the process of designing and implementing mechanisms that can help reduce the risk and combat the problems that are associated with LANs. There are probably several such mechanisms, but discussion here focuses on the most widely known and used:

- general protection of LANs and
- encryption of data.

Recommended solutions are presented in the final section.

5.1 Protecting LANs and Data

5.1.1 General Protection of LANs

There does not exist today any method to absolutely guard against the broadcast of data on a broadcast LAN to all network components; however, steps can be taken to help control this broadcast of data. Such actions include limiting the actual geographic size of the LAN and restricting the number of devices that are

allowed to attach to the LAN. The smaller the LAN, the easier it is to manage and control administratively. Along these lines, it is easier to control, monitor, and manage a LAN that has configuration control established [2]. Configuration control includes the physical protection of the hardware connected to the LAN, as well as the LAN itself [30]. The hardware (file servers and disk servers) should be placed in facilities that lock, and access to these facilities should be controlled. The PCs themselves should be controlled in a similar manner. These microcomputer devices should be fitted with keyboard locks, CPU locks, and power lock-out switches. Also, a physical inventory of all system components should be made for future reference [30]. A physical inventory should be made periodically, and the findings should be compared to the original list to determine if any components have been added or removed. The LAN should be installed in such a manner that it can be inspected easily for unauthorized connections, or, in cases of sensitive data on the LAN, it should be placed inside protective conduit [30].

Several features can be employed to provide some level of protection. For example, the operating system of the mainframe computers can be configured in such a manner to separate privileged and non-privileged users. Most computer operating systems allow for a partition between user and privileged system storage. Non-privileged users cannot access the system space. This is, of course, not the case on PCs or workstations. There is no separation because the user is in control of the entire system. Such separation would help to protect the various

system parameters from the non-privileged users. Also, the privileged users who have access to the computer must be kept to a minimum. In some operating systems, the ports that are accessible from the LAN can be configured to be privileged, and only certain ports can be used by certain users. This helps to control which users can access the computers via predefined ports.

Bridges and routers can also be used to provide isolation and separation from other LANs. Many bridges and routers can be configured to filter and/or block addresses or specific network protocols. Monitoring of the LAN with special network software can be helpful in finding "new" nodes that have been added to the network. Monitoring may detect impersonating nodes and unusual activity, such as new nodes appearing and then disappearing, or changes in LAN addresses. Monitoring techniques look for unusual activity on the LAN, and the addition/removal of nodes may trigger unusual activity. The monitoring concept may also detect any LAN tapping (especially on twisted-pair and fiber LANs). This does not work as well on coax cable because, in most cases, monitors will be able to detect the loss of signal that results from taps of either twisted-pair or fiber cables. But, depending upon the type of coax LAN, the monitor may or may not detect taps. For example, Ethernet uses the "vampire" clamp-on technique for connecting devices to the LAN. If the connected device has a valid address, the monitor will not be concerned about this new node. But if the coax is tapped in such a way that the continuity is affected, then the monitor would record a loss of signal and report a possible problem.

5.1.2 Encryption of Data

Several types of encryption are available that can be implemented on most computers fairly easily. For example, the Data Encryption Standard (DES) algorithm can be implemented either through hardware or software. The hardware utilizes preprogrammed chip sets that contain the DES algorithm, and this interface usually exists as an interface between the actual device and the connection to the network. Thus, the encryption actually is performed by this hardware interface before the data is transmitted over the network. The other method is to have the DES algorithm reside in software inside the computer itself. Once the data is ready to be sent, the encryption algorithm is accessed and the original data is placed into unreadable ciphertext. The correct node with the matching key will be the only node that can decrypt the ciphertext and access the data. This is a simplistic view of how the data is moved along the network in encrypted form. More descriptions and discussions of various encryption techniques are presented later. Some general background information is now presented.

There are at least two major encryption techniques used by most vendors who supply encryption products [31]:

- end-to-end encryption, which means the data is encrypted and stays encrypted until the data is received and decrypted by the correct node, and
- link-to-link encryption, which utilizes encryption up to the next communi-

cations link, where the data is decrypted and then encrypted back before being transmitted out to the next communications link.

The use of these two types of encryption is limited somewhat because of problems in the implementation of the encryption techniques. All the necessary hardware and software components exist for utilizing the encryption techniques, but the ability to use the encryption on specific networks has led to some problems. If end-to-end encryption is used, the total number of network nodes needs to be as small as possible. If the network is large, the distribution of the encryption keys to the network nodes will take too long. There must be an effective key management system, because if the keys are ever compromised, all encrypted data is no longer adequately protected. Also, the time that it takes to distribute new keys for all associated network nodes can pose a severe problem to the users of the network. In the case of the link-to-link encryption, the fact that the data is encrypted as it travels from node to data switching center appears to provide sufficient control. But, as pointed out in the definition above, once the data arrives at the data switching center, it is decrypted, which means that the data is vulnerable and could be captured by an unauthorized user. In many cases, time will be lost due to the encryption/decryption of data at the switching center.

There are several different types of encryption mechanisms that can be used to encrypt/decrypt information [32]:

- one-way encryption,

- symmetric encryption, and
- asymmetric encryption.

The one-way encryption algorithm is just what it implies. The data is encrypted and there is no easy way to decrypt it. Usually, passwords for access to computing resources are one-way encrypted. If the users lose or forget their password, a new one must be generated because there is no easy method to determine what the password was. Symmetric encryption almost always uses the same encryption key to both encrypt and decrypt data. As an example, the DES algorithm relies on this type of encryption. Asymmetric encryption usually utilizes two distinctly different keys, one for the encryption and one for the decryption of data. One of the keys is used to encrypt the data itself (known as the primary key), while the other key (usually derived from the primary key) is used to decrypt the data [32]. An example of the asymmetric encryption is that of the public-key encryption algorithm. In this algorithm, users know the public key and use it to send data to another user, but only the correct individual user can decrypt the message. Discussion of to how these various types of encryption schemes actually work, or how the keys are determined, is beyond the scope of this thesis.

As indicated earlier, there are several different types of encryption methods, and there are several different ways in which these methods can be utilized. For purposes of this research, three different encryption techniques and their implementation are presented:

- BLACKER,
- VISA, and
- KERBEROS.¹³

BLACKER

The BLACKER technique, for all practical purposes, has the least amount of openly published literature available. There is insufficient data or material for public review that defines how it works or what it is composed of. Little is published about BLACKER because it is intended to be used by the federal government, specifically the military. Therefore, work on BLACKER is probably being scrutinized and reviewed before being released for public use. But BLACKER does exist, and several pieces of general data have been presented. General comments about how it may be used on different networks have also been presented, but no detailed or specific data concerning its use or implementation has been openly revealed as yet to the public. Blacker utilizes link encryption which is used on TCP/IP networks, specifically the Defense Data Network (DDN). The Blacker device is physically located between the DDN network and the host computer. Incoming requests are held by the Blacker front-end device, while the Access Control Center (ACC) is checked to determine if the request is valid. Also the Key Distribution Center (KDC) is also contacted for the encryption/decryption keys [33]. Basically if the request does

¹³ KERBEROS is a trademark of the Massachusetts Institute of Technology.

not check out, the host is denied access by the Blacker device and a log record is made to indicate such an access attempt was made.

There are, on the other hand, several references that discuss the two other encryption techniques, VISA and KERBEROS. These two techniques are new, and there are not many facilities that have implemented them. Literature on the VISA technique indicates that a release for public use maybe forthcoming. However, KERBEROS is now available for public use. As a result, several computer facilities will probably obtain a copy to see if it will actually perform as well as the literature states. The use of the above encryption techniques may solve some of the problems, as well as create new ones. Potential areas of concern are raised and reviewed here.

VISA

The VISA technique was developed primarily to control the flow of data between various organizations. The assumption is that not all aspects of the organization are located in the same facility; therefore, a method is needed that will allow data to flow from any, or all, of the facilities to any other facility. The real purpose is not to establish simple network connections between organizations but, rather, to provide valid authorization to allow the flow of data between organizations. The need for such valid authorization techniques is directly related to the enhanced connectivity that LANs have provided. Earlier, LANs were used to connect computing resources to each other, but there was a phys-

ical limitation on number of devices and distance of the network. Based upon this, there usually was no need to worry about different groups or users trying to utilize LANs from remote facilities. However, with the advent of increased connection capability and advanced network hardware, the LAN soon provided remote facilities and users the ability to connect to almost any other facility. As a result of the advanced capabilities of networks, several organizations are taking advantage of the ability to interconnect to other facilities (either their remote facilities or other organizations). Along with the ability to provide greater network access and various other network capabilities, the need for network control and protection has also increased. Several mechanisms have been developed that provide security at the various layers of the ISO/OSI network reference model. While most of these mechanisms work on the individual protocol layers, there appeared to be a need to provide control while still appearing to be transparent to the user. These other mechanisms work well, but in most cases they tend to decrease performance, response, and access to remote nodes on the network. With this in mind, the concept of providing a mechanism that would place a noncounterfeit seal on network data was designed. Such a mechanism came to be known as VISA. The concept of VISA is similar to the visa needed to travel outside of the United States. A valid visa is necessary on the user's passport in order to cross the border; likewise, the user's VISA must be on the network data, and the VISA must be valid before the users may cross from one network to another.

Within VISA, the concept of trusted systems in networks carries new merit that is illustrated by the following discussion. If systems can be trusted and the algorithm that encrypts the data is secure, then the trusted systems can communicate with each other. The underlying criteria that enables this trust is data packets on datagram networks which contain information about the computers as well as data. On other types of networks information about the computer is implied and is not part of the data. For VISA to work, all nodes must be able to utilize and understand the visa mechanism.

All packets that use visas to communicate from various systems to other systems carry two visas. One visa is for leaving the originating network, and the other is for valid access to the destination network. The two visas are part of the total data packet. One is required for the originating node to be able to send data, the other is required by the destination node to receive data. The visa for leaving the originating network is validated by an access control system (ACS) that is part of the originating network hardware. Before the data can leave, the exiting visa must be valid. After this, the data may or may not pass through a gateway controller. The gateway controller is used to check all valid ACS devices to ensure that the visa was checked by a valid ACS before it was sent. The gateway controller (if used) will allow the data to pass onto the destination ACS. The destination ACS will check the destination visa, and, if valid, the destination ACS will allow the data to pass to the destination node [34].

The VISA technique can use either public-key or private-key encryption for

protecting data on the network. The use of VISA does not enhance or address every aspect of security on computers, but it does provide protection that proves security can be achieved on computers with positive results. Several other system features must be enabled to provide tighter control and security. These other features include restricting privileges, access control on users, and proper separation between the operating systems and users, to name a few. The VISA technique does not presently address the ability to control the privileged user. But if VISA is used in conjunction with other security controls and mechanisms, a more secure system may be the result.

KERBEROS

KERBEROS is a software package that is usually added on top of the existing operating system. Most computer systems require users to be identified before granting access to the computer system. This identification process in most cases requires the user to supply information that identifies and authenticates the user. This identification and authentication is usually in the form of a user ID and password. In several cases, specific network services are used to authenticate the user. Most of the time, this type of validation is acceptable. This is specifically true if the user does not have any method by which to change or modify network addresses and system privileges or to effect changes on his remote device. This type of control is useful if all remote users are using terminals that cannot be modified or changed to allow these remote users to bypass the

network services. However, if these remote users are using PCs, workstations, or other intelligent network devices, additional control must be provided. These types of remote devices can be configured to impersonate other network nodes and also change network node addresses. For this reason, the network service previously mentioned no longer provides adequate protection of resources. Thus, new controls must be developed that provide additional protection. One such control is KERBEROS, which is intended to provide another layer of control if the network service for validating remote users is not sufficient.

The objective of KERBEROS is to provide control mechanisms that cannot be bypassed by unauthorized network users. KERBEROS also provides accounting data about users who have passed through its validation either incorrectly or correctly [35]. At this point, the only operating system that it has been designed and developed for is the UNIX¹⁴ environment. Since it is a layered software module, several existing UNIX and TCP/IP commands must be modified to work under KERBEROS. For example, the *login* capability, *ftp*, *telnet*, and other system and network functions would need to be modified to utilize the KERBEROS software. The KERBEROS software is basically a software programming tool. The programmers can develop and utilize various software subroutines within KERBEROS to allow users to encrypt specific applications between computers and users for the sole purpose of protecting these applications and associated data from other users. When KERBEROS is installed, it provides a standard

¹⁴ UNIX is a trademark of AT&T Bell Laboratories.

set of programming tools and capabilities that allow programmers to develop mechanisms to protect certain software applications.

To see how encryption can be used to protect data on a LAN, KERBEROS is examined in detail. First, the objective of KERBEROS in relation to the specific problems of LANs as described in Chapter 1, (promiscuous mode, node impersonation, and disruption of service) is stated. Second, the components that make up KERBEROS are discussed, as well as how they interact with each other. Third, discussion about how KERBEROS can be used to solve the problems on LANs is presented, along with discussion as to what KERBEROS cannot do. KERBEROS will not stop or contain any disruption of service problems; however, KERBEROS can be used effectively on the promiscuous mode and node impersonation problems.

Discussion is now directed to the various components that make up KERBEROS. Below are items that compose the KERBEROS systems. Information about each of these components and how they function is also provided. Items that make up KERBEROS are [36]:

- KERBEROS applications library,
- encryption library,
- database library,
- database administration server,
- authentication server,

- database propagation software,
- user programs, and
- various applications.

The “applications library” provides software routines to read specific authentication requests or to create totally encrypted data messages. The “encryption library” uses the DES encryption algorithm. The encryption library has been designed and implemented as a independent software module. This will allow variations of the DES algorithm or other encryption algorithms to be used within KERBEROS. The “database library” is also removable and could be replaced with other database management systems. The only requirement for the database system is that the data for the authorized users be kept current and reflect the current configuration. Some of the entries for the database include valid entries for each user which contain name, private key, expiration date of the user and other information pertaining to the user. The “database administration” programs are used by the KERBEROS-authorized users to maintain the database and also to enter the information into the database. The “administration server” provides the proper interface to the users on the network who access the database. This interface is read-only from the user side and writable for users who maintain the database. The “authentication server” allows for the authentication of users and the generation of keys for the protection of data or for the services requested. The “propagation software” allows for additional

copies of the database to be on several different processors along with copies of the authentication software for the remote slaved processors. The designated master KERBEROS will provide updates to these slaved processors. Finally, the "user and application programs" provide the interface for the users who login to the authentication server, change accounting information, or either view or cancel tickets for specific services.

These components are the minimum needed to utilize the KERBEROS system. There are three features that KERBEROS provides for the protection of data on LANs: user authentication, data and message verification, data and message encryption. Some basic information is provided here concerning how KERBEROS functions. The first step in validating a user is to ensure that the user requesting access is known by KERBEROS. The second feature, the verification of data and messages, is performed by various KERBEROS utilities to ensure that the data and messages remain correct. This verification is done by utilizing encrypted checksums to help protect data from tampering. The last feature, data and message encryption, is performed by using the encryption algorithm modules that are part of KERBEROS.

The first aspect of using KERBEROS requires that the user be registered within the database of the KERBEROS authentication server. If users are not established as valid users within the authentication server, they will not be allowed to utilize KERBEROS services. Users must supply information to the KERBEROS authentication server. This information is usually the user's name

(for access purposes) and password. The privileged systems programmers access the KERBEROS master database and add the information about the user to the master files. The user's private key is encrypted along with the KERBEROS authentication server's master key. This provides additional security in that if the user's private key is compromised, the unauthorized user still needs to know the master key to obtain access to the user's data that is part of KERBEROS. Also maintained in the KERBEROS authentication system are entries of all approved services for the various users. These services include systems that require KERBEROS authentication for *rlogin*, using remote printers, using file servers, and other types of services.

In order to use these various services, the user must obtain valid tickets from KERBEROS. Tickets are granted if, and only if, the KERBEROS authentication server contains information about the user in its master database. Initial tickets for services are granted when the user logs on initially or through the use of KINIT (explained later) if login has not been modified to use KERBEROS. At the time of login, the user is asked to supply his/her username. Once this is entered from the remote device, a request is made to the KERBEROS authentication server. The authentication server checks the KERBEROS master database for information concerning the user. If this check results in a positive match, information containing the ticket for the ticket-granting service and information about the user is sent back to the user. This information from KERBEROS to the user is encrypted using the user's password. Once this information arrives

at the user's device, the user is asked to input his/her password. The password is used to decrypt this information, and the information for the ticket-granting service is stored at the user's device (maybe a PC or workstation) for future use.

Tickets may be granted by using a routine called KINIT. The KINIT routine is provided by KERBEROS and can be used in conjunction with the login routine of the system. The KINIT is used if the login session lasts longer than the timestamp of the ticket-granting capability or if login has not been modified to support KERBEROS. If this should happen, the user can use KINIT to receive new tickets for the ticket-granting service. The KINIT routine performs in a manner similar to the login sequence described above.

The user now has all the necessary information that can be used to request from the ticket-granting service specific tickets for services on the LAN. The user obtains from the KERBEROS authentication server a ticket that is used to request, from the ticket-granting service, a specific service that the user wishes to utilize [36]. This request from the user to the ticket-granting service is comprised of information such as the name of the server, the ticket-granting ticket, and the authenticator that validates the user. The ticket-granting server will validate the authenticator and the ticket-granting ticket. Once this validation is completed and there is a match, the ticket-granting server will supply a randomly generated session key that will be used by the user and the specific server. Next the ticket-granting service develops a ticket for the service that has been requested. This ticket contains the user's name, the server name, the user's Internet Protocol

(IP) address and the session key. The ticket-granting server transfers this data back to the user.

Now that the user has obtained tickets for services, how is the user able to utilize these services? The user sends the ticket to the server for which the user is requesting access. This ticket is encrypted using the server's key. The server decrypts this data and compares it with data resident with the server. If all the data matches, the user is allowed to access the requested service. If there are areas that do not match, the service is denied to the user.

With the session key, a user and server can utilize a mechanism to send/receive "safe" messages, which use encrypted checksums from source to destination. The source (user) will encrypt checksum data at the end of the data stream, and the destination (server) will decrypt and compare the checksum. This will add some protection to the data stream. The user and server can also utilize a mechanism to send/receive "private" messages. These messages use encryption on the entire data stream using the session key, which is known only to the server and user.

The above is a broad overview of how KERBEROS can be used to authenticate users for specific services. But what does the privileged systems programmers need to do in order to fully implement and use KERBEROS? The user never realizes that there is a validation/authentication mechanism in place. This aspect is totally transparent to the end-user. The system programmer must, in many cases, modify existing applications in order to fully utilize KERBEROS. The example that follows discusses some of these modifications that

are necessary in order to use KERBEROS with the *rlogin* utility. The *rlogin* utility allows terminals from local hosts the ability to connect to remote hosts over a TCP/IP network. Without the use of KERBEROS, this capability is provided via the *rhost* mechanism. The *rhost* mechanism allows hosts that are trusted to access each other without any additional authorization. To provide enhanced security, KERBEROS can be used to authenticate these remote users and determine if they have valid remote access accounts.

In order to allow for this use of KERBEROS, the system programmer must make some modifications to the existing *rlogin* network application software using routines that are a part of the KERBEROS libraries. Software code must be added to *rlogin* (client) to call the KERBEROS library routine to request authentication for the desired service. The information resulting from this authentication is passed onto the *rlogind* (server) responsible for the service. The *rlogind* (server) program must be modified to receive the authentication data from the user and also call the KERBEROS library routine to verify the authentication data. Examples of the KERBEROS library routines that are used for this are *krb_mk_req()* on the user side and *krb_rd_req()* on the server side. The *krb_mk_req()* is the KERBEROS routine that requests a ticket for a service. This request contains information about the users authentication, the service to be requested, and checksums on data if required. This data is passed from the user to the KERBEROS system, the KERBEROS system gathers the information from its master database and sends authentication/ticket data back to the

user. The user then sends this information to the server. The server, once the information is received, makes a call to the KERBEROS library routine called *krb_rd_req* and decrypts the ticket with its server key and verifies the authentication information. If these checks are confirmed, the user is allowed access to the service. If not, the user is denied access.

The *rlogin* software was modified to utilize *kcmd()* in place of the *rcmd()* to provide valid authentication and service to the remote *rlogind* server. To provide encryption, the *-x* option was added to allow users to invoke the DES encryption/decryption for sending/receiving data over the network. On the *rlogind* (server) side, code was added to allow for the authentication of the received data from the user. The *krb_recvauth()* is used to verify the data from *rlogin*, and the *kuserok()* is used to verify the user is authorized for the service.

The example above indicates how KERBEROS can be used for protecting information from local nodes to remote nodes by modifying the *rlogin* and *rlogind* programs to use KERBEROS. As another example of how KERBEROS can be used to enhance network security, an example of how routing services can be protected is discussed. In order to use KERBEROS for network routing services, the network nodes, as well as the routing nodes, must be registered with the KERBEROS master database as previously described. There are two examples of how routing services could be protected using KERBEROS. These examples are discussed below.

If the network routing service is provided over point-to-point logical links

(TCP), then the ability to use KERBEROS is basically the same as explained in the *rlogin/rlogind* example above. In this situation, KERBEROS “safe” messages are used to ensure the integrity of the message. These “safe” messages use encrypted checksums to safeguard the data. The network routers (servers) and the network router (users) utilize the *safe* message subroutines *krb_mk_safe()* and *krb_rd_safe()* from the KERBEROS libraries.

Now if the router broadcasts the routing updates (e.g., *routed* and Berkeley UNIX), the existing KERBEROS routines and services are inadequate. Previous discussion pointed out that the use of fraudulent routers can result in data capture or node impersonation. If the network routing users can be trusted, KERBEROS could be modified to provide multiple tickets with one common session key. However, if these routing users cannot be trusted, they can impersonate the network router very easily because they know the key. This key would be used to pass the routing updates to all network nodes for network routing services.

Public key encryption could be used if the network nodes cannot be trusted. Each network user node could request a ticket for the network router, and a common session key would be used for each network node. The network router would use the common session when broadcasting routing data. The user nodes would then use the key to validate that the data has not been altered. To support the use of common session keys, the KERBEROS master database would be modified to indicate that a service uses a common session key. This key would

be kept available so that it could be reissued when a user requests a routing ticket. The various KERBEROS database programs and server programs must be modified to utilize the common session keys. Based upon recent data from the KERBEROS project team, this concept is being considered as an extension for future KERBEROS releases.

There are certain areas for which KERBEROS cannot provide additional protection for data. If the PCs or workstations using KERBEROS services are public devices, as opposed to private devices, KERBEROS is not applicable. If these PCs or workstations are used by several different users who can modify the system files and other users' data files, there is no way KERBEROS can be used to protect this information (because the server key is stored in clear text on the server). Another area of concern deals with the user wishing to access other remote hosts after the user has been granted access via KERBEROS to the first remote host. Under the present KERBEROS implementation, the ability to provide authentication forwarding does not exist. Therefore, the user cannot use KERBEROS services from any of these remote hosts until the user validates himself explicitly to that host. When this occurs, the user's password is transmitted across the network in clear text. These are known problems that are being studied by the KERBEROS project team. Finally, KERBEROS provides only authentication; authorization must be handled by each individual server.

The following problems can be solved, or at least modifications can be made to address them. In some cases, existing KERBEROS or existing network ap-

plications need to be modified in order to fully utilize the KERBEROS services. These modifications will require time, programmers, and funding in order to install and implement KERBEROS. The use of full encryption for all data will result in a dramatic loss of response time and increased compute load because the computer must encrypt all data and send it; then the computer must be able to decrypt data that is sent to it for processing. To ensure the protection of the users' passwords as they are stored on the KERBEROS system, a valid mechanism must exist to provide protection of these passwords. Such protection is necessary because the KERBEROS manager enters the user's password into the KERBEROS master database in clear text and the entire password file is two-way encrypted (which allows the KERBEROS manager the capability to decrypt the password file). This protection must not allow these passwords to be exposed to any other user; therefore, the present password-handling scheme may need to be reviewed to determine if it is adequate.

5.1.3 Recommended Solutions

Good configuration and administrative control can provide a strong foundation on which control of LANs be based. Relying on such methods by themselves will only provide a false sense of protection. In most cases, if the size of the LAN is small and the number of users is also small, these methods may be adequate. However, once the LAN expands, the ability to protect resources

and data are no longer trivial. The ability to manage an expanding LAN will, in most cases, require more resources than management is willing to provide. This raises a paradox when management requests that resources and information be protected. But when more resources, such as manpower and equipment, are needed to provide additional protection, management wonders why. By the same process, if the organization ever loses data or resources due to the lack of protection, management cannot understand why more resources were not requested to provide better protection. In these cases, no action is taken until the network has been compromised. If management had planned properly when the systems and networks were installed, maybe the loss of such data and resources could have been mitigated. The obvious point is that all aspects of computer systems and networks must be included at the planning and budgeting stages, as opposed to being an afterthought.

Encryption of data can be used to provide additional protection of data. As already noted, encryption prevents unauthorized users from stealing data that is in plain text or not protected at all. But the encryption technique seems to be the most cost-effective over dedicated point-to-point data links [37]. The cost for utilizing encryption over LANs and multi-connected networks is still quite high [37]. Data encryption will indeed stop users from stealing data via taps, listening devices, or other methods that can be used on LANs to obtain data, but the encryption technique does nothing to control access to the computer itself. If the user provides the correct sequence, the computer assumes that the user is

valid and access is granted. Good access controls must be coupled with data encryption, configuration, and administrative controls to adequately protect data and computing resources. Each organization must understand the requirements they must follow to protect data, and then they must implement such controls. For example, the need to protect data and resources in the academic environment will be less than the protection for classified data at a government facility.

Another vulnerability discussed throughout this document is the ability of users to change or modify the Ethernet address or the network physical address. Maybe the time has come for the vendors to design and develop hardware device components that do not allow the network physical addresses to be modified or changed. Often, these same network hardware devices support the promiscuous mode. This type of support should no longer be an option on the hardware. As an example, the DECnet hardware should not allow or support the ability to change the Ethernet address (as discussed in Chapter 4). In cases where the addresses need to be changed, maybe the user should be required to swap hardware. Or, in other cases, users may not be allowed to change predefined addresses at all. The ability to change addresses at will in an uncontrolled manner is not acceptable any longer.

Encryption has been discussed and presented as the ultimate protection of data on a LAN; however, technology and components must become advanced to the point that high-speed, affordable data encryption will be usable for all systems. At present the use of encryption is not meant for high-speed data

transfer and does not handle large data files easily. But given time and relatively small data files, present data encryption performs quite well. Once again, technology is forcing the issue. What was adequate is no longer acceptable by today's standards.

There is another method that can be used with encryption to provide better security for data on LANs: physical separation between different types of networks that process various types of data. Most of the time, the only true separation that exists is a logical one, and this type of separation provides little isolation. By isolating the various types of data onto separate networks, better control, protection, and administration can be provided to the networks that process classified or sensitive data. The use of separation may cost more than other presently used methods, but the ability to protect data in this manner is more readily realized. The use of this method, coupled with data encryption, could provide an effective mechanism to increase the overall integrity of data that is processed via LANs.

Additional control can be placed on users by requiring that no access be allowed to routers/gateways. All users will be granted access to the various nodes on the network, but generic users have no need to access communication processors directly. Systems personnel will need access to these types of processors, but this set of privileged users can be kept to a minimum; therefore, monitoring who has access will be somewhat easier. It will be more difficult to effectively control these privileged users, as their privileges allow them to perform certain

system functions; however, system personnel should not be allowed to access all gateway/router processors. They should be allowed access only to certain individual systems.

There are monitoring capabilities that are very useful in the LAN environment, and such capabilities can aid in detecting LAN problems, but there are no monitoring tools that are an integral part of the computers on LANs. All monitoring is provided via add-on software and hardware to existing systems. Monitoring tools must be built into systems, and such tools must be able to perform many more functions than are presently available.

LIST OF REFERENCES

LIST OF REFERENCES

1. Branstad, Dennis L., "Considerations For Security In The OSI Architecture," *IEEE Network Magazine*, [USA], Vol. 1, No. 2, April 1987.
2. Madron, Thomas W., *Local Area Networks - The Second Generation*, John Wiley and Sons Inc., 605 Third Avenue, New York, NY., 10158-0012.
3. Tanenbaum, Andrew S., *COMPUTER NETWORKS-Second Edition*, Prentice Hall Incorporated, Englewood Cliffs, New Jersey, 1988.
4. Zimmermann, Hubert, "OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection," *IEEE Transactions on Communications*, Vol. Com-28, No. 4, April 1980.
5. Tanenbaum, Andrew S., *COMPUTER NETWORKS*, Prentice Hall Incorporated, Englewood Cliffs, New Jersey, 1981, pp.15-21.
6. McMaster, David S., "IBM's TOKEN RING: What Are The Alternatives?," *Mini-Micro System*, [USA], Vol. XIX, No. 11, September 1986.
7. Southard, Robert K., "LANs: STATE OF THE UNIONS," *Mini-Micro Systems*, [USA], Vol. XX, No. 9, September 1987.
8. Stallings, William, *LOCAL NETWORKS: An Introduction*, Macmillan Publishing Company, 866 Third Avenue, New York, New York, 10022, 1984.
9. Comer, Douglas E., *Internetworking With TCP/IP Principles, Protocols, and Architecture*, Prentice Hall Incorporated, Englewood Cliffs, New Jersey, 1988.
10. White, David W., "By This Token, 802.5 TOKEN RING HITS THE BIG TIME," *LAN: THE LOCAL AREA NETWORK MAGAZINE*, September 1988, page 56.
11. Haugdahl, Scott, J., *Inside the Token Ring*, [USA], Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN., 55424, April 1986.
12. International Business Machines Corporation, "RT PC Technical Reference Token-Ring Network Adapter," *Publication No. SK2T-0291-1*, International Business Machines Corporation, Dept. E02, P.O. Box 12195, Research Triangle Park, North Carolina, 27709-9990.

13. Connor, Nancy, "TCP/IP Training," *LAN Magazine*, [USA], February 1988.
14. Draft Revised IP Security Option, *Network Working Group*, Request For Comment (RFC) 1038, St. Johns, January 1988.
15. DECnet/VAX Internals, *Seminar Notes*, Digital Equipment Corporation, Maynard, Mass.
16. DECnet, *Digital Network Architecture (Phase IV) General Description*, Digital Equipment Corporation, Maynard, Mass.
17. DECnet Protocol Analysis Seminar Notes, *Protocol and Flow Control*, Digital Equipment Corporation, Maynard, Mass.
18. International Standard Organization, "Information Processing Systems - Open Systems Interconnection Basic Reference Model," ISO 7498, October 1984.
19. Hunter, John J., "Bridges, gateways open to confusion," *NETWORK WORLD*, Vol. 4, No. 40, October 5, 1987.
20. Dunigan, Thomas H., *Network Security*, January 19, 1987.
21. DEUNA User's Guide, Digital Equipment Corporation, [USA], *Manual Number EK-DEUNA-UG-001*, 1983.
22. DELUA User's Guide, Digital Equipment Corporation, [USA], *Manual Number EK-DELUA-UG-002*, Second Edition, April 1986.
23. Hedrick, Charles L., "Introduction to the Internet Protocols," Computer Science Facilities Group, Rutgers State University of New Jersey, 1987.
24. Plummer, D.C., "An Ethernet Address Resolution Protocol," RFC-826, Network Information Center, SRI International, November 1982.
25. VAX/VMS Networking Manual, *Order Number: AA-Y512C-TE*, [USA], Digital Equipment Corporation, Maynard, Mass., April 1986.
26. VAX/VMS I/O User's Manual: Part II, Digital Equipment Corporation, [USA], *Manual Order Number: AA-Z601A-TE*, September 1984.
27. VMS: Guide to VMS System Security, Digital Equipment Corporation, [USA], *Manual Order Number: AA-LA40A-TE*, April 1988.
28. International Standard Organization, "Open Systems Interconnection Basic Reference Model," Part 2: Security Architecture, ISO 7498-2-1988(E), 1988.

29. Hungington, Jill, Ann, "OSI-based net management: Is it too early or too late?," *Data Communications*, Vol. 18, No. 3, March 1989.
30. Strehlo, Christine, "The Well-Protected Network," *Personal Computing*, [USA] January 1988.
31. Turn, Rein, "Encryption for Data Security," Chapter I-3-5, *Handbook Of Communications Systems Management*, James W. Conrad, Editor, Auerbach Publishing Inc., Boston, Mass., 1988.
32. Schwartz, Michael, "Understanding Public Key Encryption," Chapter I-3-6, *Handbook Of Communications Systems Management*, James W. Conrad, Editor, Auerbach Publishing Inc., Boston, Mass., 1988.
33. Bellovin, S.M., "Security Problems in the TCP/IP Protocol Suite," *ACM Computer Communication Review*, Vol. 19, No. 2, April 1989.
34. Estrin, Deborah, Mogul, Jeffrey, and Tsudik, Gene, "Visa Protocols for Controlling Inter-Organizational Datagram Flow," *DRAFT VISA PLAN*, Unpublished technical document, University of Southern California, Los Angeles, California, and Digital Equipment Corp., Western Research Lab.
35. Miller, S.P., Neuman, B.C., Schiller, J.I., Saltzer, J.H., "Kerberos Authentication and Authorization System," *Project Athena Technical Plan*, Section E.2.1, Massachusetts Institute of Technology, December 21, 1987.
36. Neuman, Clifford, Schiller, Jeffery, I., Steiner, Jennifer, G., "Kerberos: An Authentication Service for Open Network Systems," *Kerberos Overview*, University of Washington, Massachusetts Institute of Technology, March 30, 1988.
37. Sharma, Ranjana, "Unlocking the Secrets of Network Security Devices," *NETWORK WORLD*, Vol. 6, No. 2, January 16, 1989.

VITA

Chad Matthew Pfleger was born in Sturgis, South Dakota, on January 11, 1956. He graduated from Brown High School in Sturgis in May 1974. Mr. Pfleger has an Associates Degree in Electronics from the Northwest Vocational Technical school in Rapid City, South Dakota. He also has a Bachelor of Science degree with a major in Computer Science from the University of South Dakota, Vermillion, South Dakota.

He received the Associates Degree in Electronics in May 1976 and graduated from the University of South Dakota in December 1980. In May 1981, he accepted a position as a computing analyst with Union Carbide Corporation, Nuclear Division, operators of government installations for the United States Department of Energy in Oak Ridge, Tennessee.

Mr. Pfleger entered The Graduate School of the University of Tennessee, Knoxville, in March 1983. He received the Master of Science degree with a major in Computer Science in August 1989.

Mr. Pfleger is presently employed by Martin Marietta Energy Systems, Inc., operators of government installations for the United States Department of Energy in Oak Ridge, Tennessee, since 1984.