

Computer Network Security for Large Scale Networks

**A Faculty Scholars Report
(Half-Time Stipend)**

**Rick Kovacic
CSMM Department
Eastern Oregon University**

submitted Spring 2004

Table of Contents

COMPUTER NETWORK SECURITY FOR LARGE SCALE NETWORKS

ABSTRACT	4
<i>Objective</i>	<i>4</i>
<i>Challenges</i>	<i>4</i>
<i>Recommendations</i>	<i>4</i>
<i>Use of this Document</i>	<i>5</i>
<i>Acknowledgements</i>	<i>5</i>
SECURITY TECHNOLOGIES OVERVIEW	6
<i>Cryptography</i>	<i>6</i>
Symmetrical Encryption	7
Asymmetrical Encryption	7
Message Authentication	7
Public Key Encryption	8
<i>Network Security Applications</i>	<i>9</i>
Authentication Applications	9
Kerberos	9
X.509	10
Electronic Mail Security	11
Pretty Good Privacy (PGP)	11
S/MIME	11
Internet Protocol Security	12
IPSec	12
Internet Protocol (IP)	13
Web Security	15
Secure Socket Layer (SSL)	15
Transport Layer Security (TLS)	15
Secure Electronic Transaction (SET)	16
Network Management Security	16
Wireless Network Security	18
WWAN	20
Data Transmission	20
Equipment Security	21
WLAN	21
Access	22
<i>Network Security Systems: Host</i>	<i>26</i>
Network Intrusion	26
Intruders	26
Passwords	27
Malicious Software	29
Malicious Programs	29
Software Viruses	30
Advanced Virus Protection	31
Generic Decryption	32
Digital Immune System	32
Behavior Blocking Software	32
Network Firewalls	32
Trusted Systems	33
DMZs	35
Virtual Private Networks	39
Honeypots	40
Intrusion Detection Exchange Format	41
<i>Network Security Systems: Client</i>	<i>42</i>

Remote Connections: SSH, sFTP, WebDAV	42
SSH: The Secure Shell	42
Open SSH	43
sFTP: secure FTP	43
webDAV: web-based Distributed Authoring and Versioning	44
Virus Protection	45
TESTING	46
<i>Port Scanning</i>	46
Overview	46
Comments and Conclusions	48
<i>Network Sniffing</i>	49
Overview	49
Comments and Conclusions	50
<i>Exploits</i>	51
Overview	51
Comments and Conclusions	56
<i>Rootkits</i>	57
Overview	57
Comments and Conclusions	60
<i>Spoofing and Denial of Service</i>	62
Spoofing	62
Denial of Service (DoS)	66
Comments and Conclusions	68
<i>Computer Viruses</i>	69
Workstation Security	69
Network Security	70
Comments and Conclusions	71
Further Study	71
FOOTNOTES AND BIBLIOGRAPHY	72
REFERENCES	73

Abstract

Objective

Computer network security has evolved dramatically in the post-9/11 world, the result of a growing community of software hackers and computer 'terrorists', and the increasing number of remote users accessing a variety of information for business, education, and personal use. Security is of a concern for both clients (users of a network) and the administrators trying to protect sensitive data located on protected areas of their network.

There are a variety of different aspects of network security to consider: protecting proprietary or restricted data; retrieval and updating public data; access and retrieval of data from remote locations, allowing access to sensitive or confidential information, and the like. This Study has researched the technical aspects of the variety of network security solutions being developed for modern day networks, how these solutions affect network users and administrators, and recommend solutions that could be implemented on EOU campus network and its subnetworks (if not already done so).

Challenges

There is a proliferation of information regarding network security, available electronically and in print form. Gaining access to security used on various networks and subnets is, however, unobtainable in most instances. A major challenge of this Study is to create a realm or environment in which to research and test security technologies without compromising any the existing security mechanism in place. Thus, decisions were made to **1)** not install or conduct any tests on the EOU network inside the main firewall, and **2)** to create an alternative operating environment for testing purposes, outside of the university campus, that would be representative of real-world use.

Another challenge encountered was the amount of technology available to research versus the chronological scope of this Study. With the advance of network technologies into high-speed Internet access using different technologies; upgrades in network switching and routing hardware; the proliferation of network threats; and advances in wireless networking numerous new standards have been implemented dealing with these various networking schemes and applications regarding security. Overviews of many technologies are provided here, with more specific information and testing limited to those seen as most useful to a campus network environment; the list of technologies here is in no way all-inclusive.

Recommendations

Comments are interspersed through this document related to possible solutions to problems that may appear on the local campus network. Some are general and common sense in nature, others are specific to network devices, server operating systems, or some other particular technology either existing or emerging. For a quick reference, visit the **Comments and Conclusions** area of the various technologies and threats considered in the Testing section of the document, starting on page 46.

Use of this Document

This Report is an overview of network security technologies and applications for wide area networks at its present state. It is written as a reference document for persons not familiar with computer security technologies or administration; this is not considered to be a technical report nor a network administrators guide to network security.

Concepts and technologies are summarized and presented for easy reading. In-depth computations, encryption codes, and the like have been translated and condensed to enhance the subject matter.

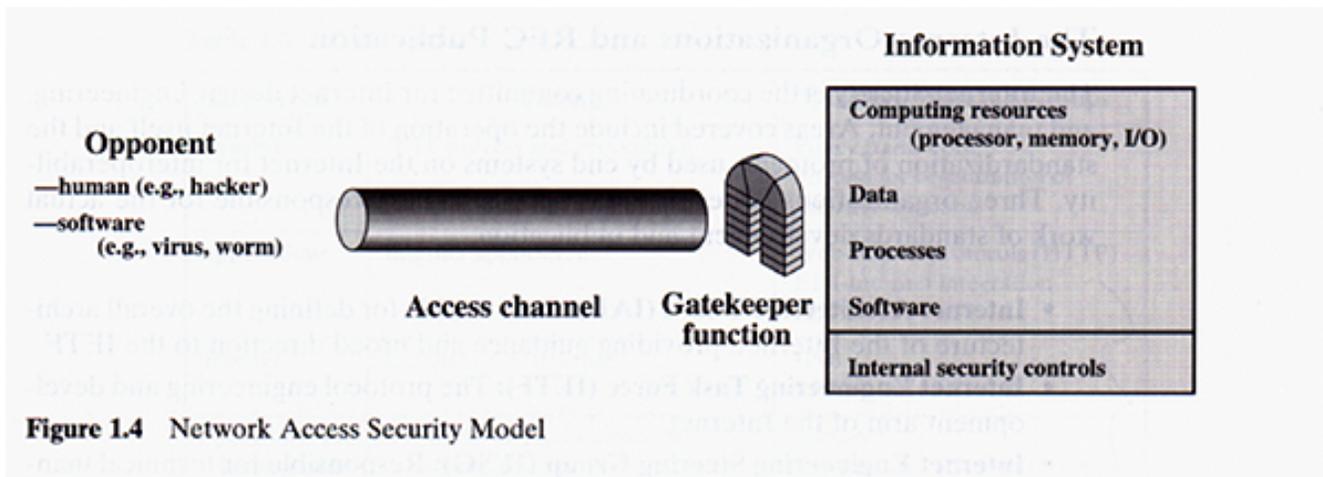
Acknowledgements

The Author would like to acknowledge the following parties for their contribution and support of this research project:

- members of the Faculty Scholars committee for providing the research opportunity
- author William Stallings, for his excellent writings on network security
- the various contributors, both personal and public, to network security standards
- colleagues and friends at EOU that believe in constructive use of technology.

Security Technologies Overview

There are two main trends in computer usage and technology which emphasize the need for security on networks. First, the explosive growth in computer systems and their interconnections via networks has increased the dependence of both organizations and individuals on the information stored and communicated using these systems, Second, the disciplines of cryptography and network security have matured, leading to the development of practical, readily available application to enforce network security.



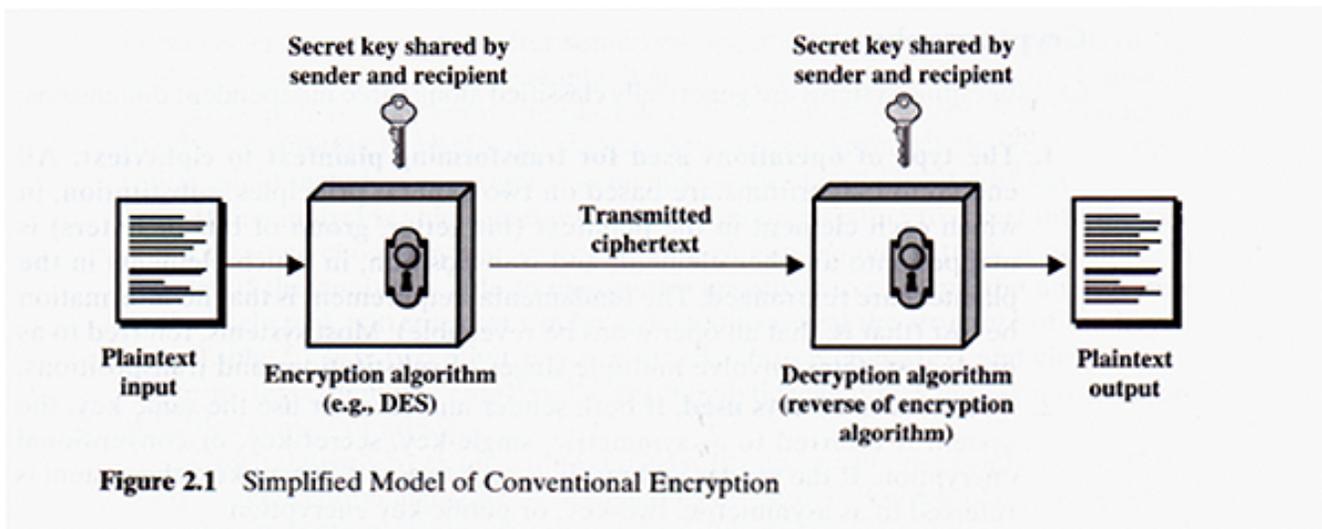
Below is an overview of technologies that deal with these trends and issues. The application of selected aspects of these technologies in commercial network products are explored in the Testing section of this Report.

Cryptography

def.: the science or study of secret writing, especially code and cipher systems. The procedures, processes, methods, etc. of making and using secret writing, as codes and ciphers.

Cryptographic systems and applications are generally categorized according to one of three different aspects or dimensions:

1. the operation used for transforming the original data into some sort of encrypted form
2. the number of keys used by the parties sending and receiving the data, and
3. the manner in which the original data is processed.



Below are different schemes that utilize most or all of these aspects of cryptography.

Symmetrical Encryption

Referred to as conventional or single-key encryption, Symmetrical Encryption was the only type of coding used prior to the development of key encryption in the 1970s. This type of encryption is used in a variety of commercial products, from simple messaging to complex e-commerce systems. This encryption scheme has five main aspects or ingredients:

- the original message or data to be encrypted
- the mathematical encryption algorithm that transforms the data into some sort of code
- a secret key that allows the data to be encrypted and decrypted
- the scrambled message produced by the algorithm, called ciphertext
- a decryption algorithm that deciphers the text back to its original form.

There are two main aspects to this kind of encryption: an algorithm is used to scramble the original text into a meaningless form, and both the sender and receiver of this ciphertext hold keys to creating and decoding this encrypted data. This provides for data that is computationally secure, and resists passive attacks over a network.

Asymmetrical Encryption

See “Public Key Encryption” below.

Message Authentication

Encrypting messages for email and other purposes protects against possible invasion of privacy in communication. With the amount of text messaging occurring on most networks, authentication of users and corresponding message content is beneficial to both senders and receivers.

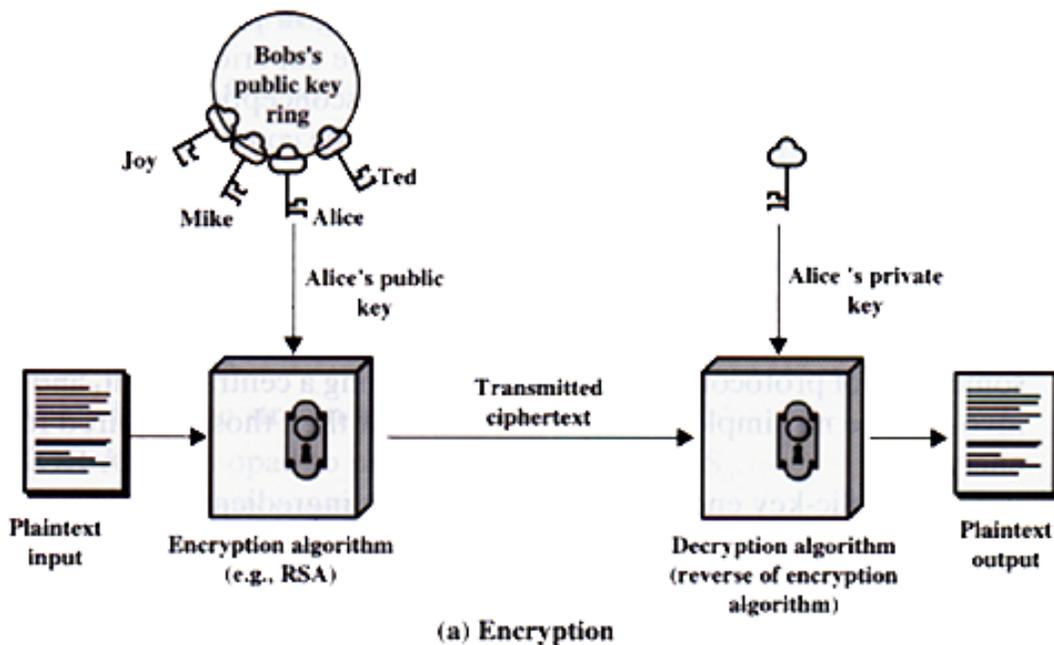
Parties trying to read or interrupt message flow do so in one of two general methods:

- passive attacks: eavesdropping on message flow across network channels is the most common method of message hacking;
- active attacks: generally falsification of data in message content and transactions. Also included in this category is false representation of sender and/or receive of text messages.

Guarding against passive and active message attacks is referred to as Message Authentication. Conventional encryption can provide for authentication of senders & receivers in email messaging and content; other approaches can provide user authentication without the network and data overhead that encryption requires, including use of a Message Authentication Code (MAC); 1-way Hash functions; and a hybrid of MAC using a Hash function called HMAC. Many of these technologies are utilized in commercial email and other communication software applications.

Public Key Encryption

Key encryption involves the use of user 'keys' to identify and authenticate senders, receivers of messages and other data. Public key encryption is asymmetrical; that is, more than one key is used by each party to authenticate their identity on the network and to access contents of messages. Public key encryption is based on complex mathematical formulas, which provide more security than simple hashing patterns. Thus, PKE requires more processor time and can create delays in sending & receiving data over a network.



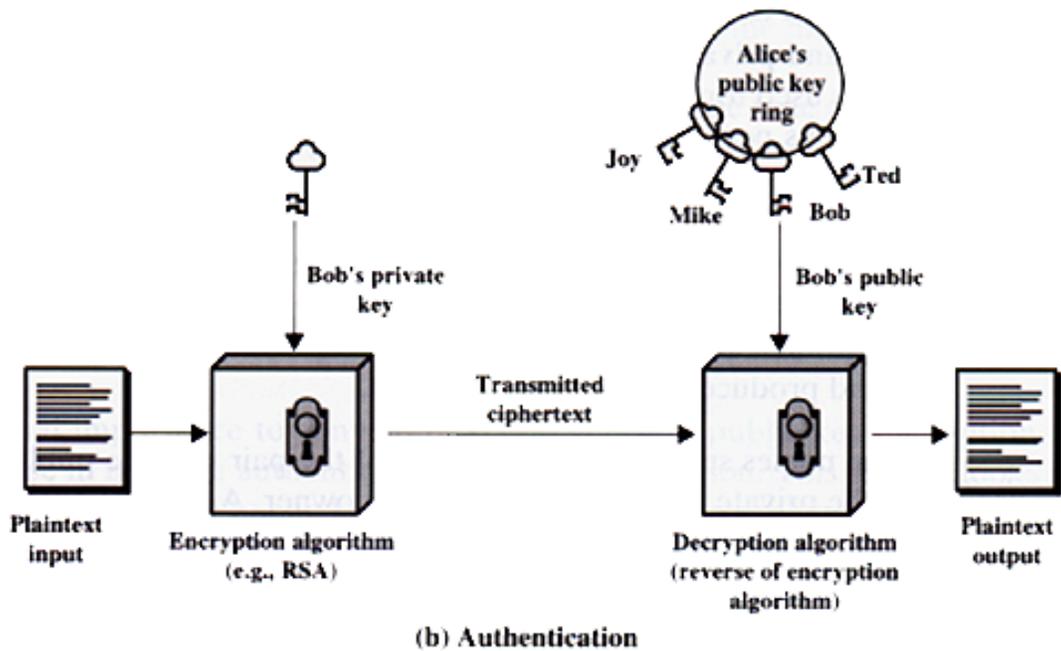


Figure 3.7 Public-Key Cryptography

When utilizing PKE, a neutral 3rd party may be employed to generate the keys for users on a network; this 3rd party may also be used to store often-used keys available to users of a large network. Using this 3rd party may potentially create a security issue for network administrators who do not want to generate or store public keys internally on the network on which the keys are used.

Public key encryption is not designed as a substitute for other forms of encryption, and is often utilized in online transactions where the exchange of currency is involved. PKE may be considered overkill for regular text messaging on an intranet where confidentiality of information is not a major concern.

Network Security Applications

Authentication Applications

Kerberos

Developed through a research project at MIT, Kerberos is designed as an authentication service deployed on open, distributed network environments. Specifically, Kerberos is used to identify users on a network, and to regulate access to servers and data on the network for each user.

There are two versions of Kerberos currently in use: a) version 4, which is widely used on large scaled networks, and b) version 5 which is being proposed as a new Internet Standard as RFC 1510. The original requirements for Kerberos included the following:

- it had to provide a degree of security so as no passive attack (eavesdropping) could obtain any information by impersonating an authorized user of that network;

- it had to be reliable enough to work in a distributed environment even if supported network services experienced technical problems;
- a degree of transparency to the user had to exist so users were not aware of the authentication processes taking place;
- it had to be capable of scaling up or down, supporting differing numbers of users and servers on a network.

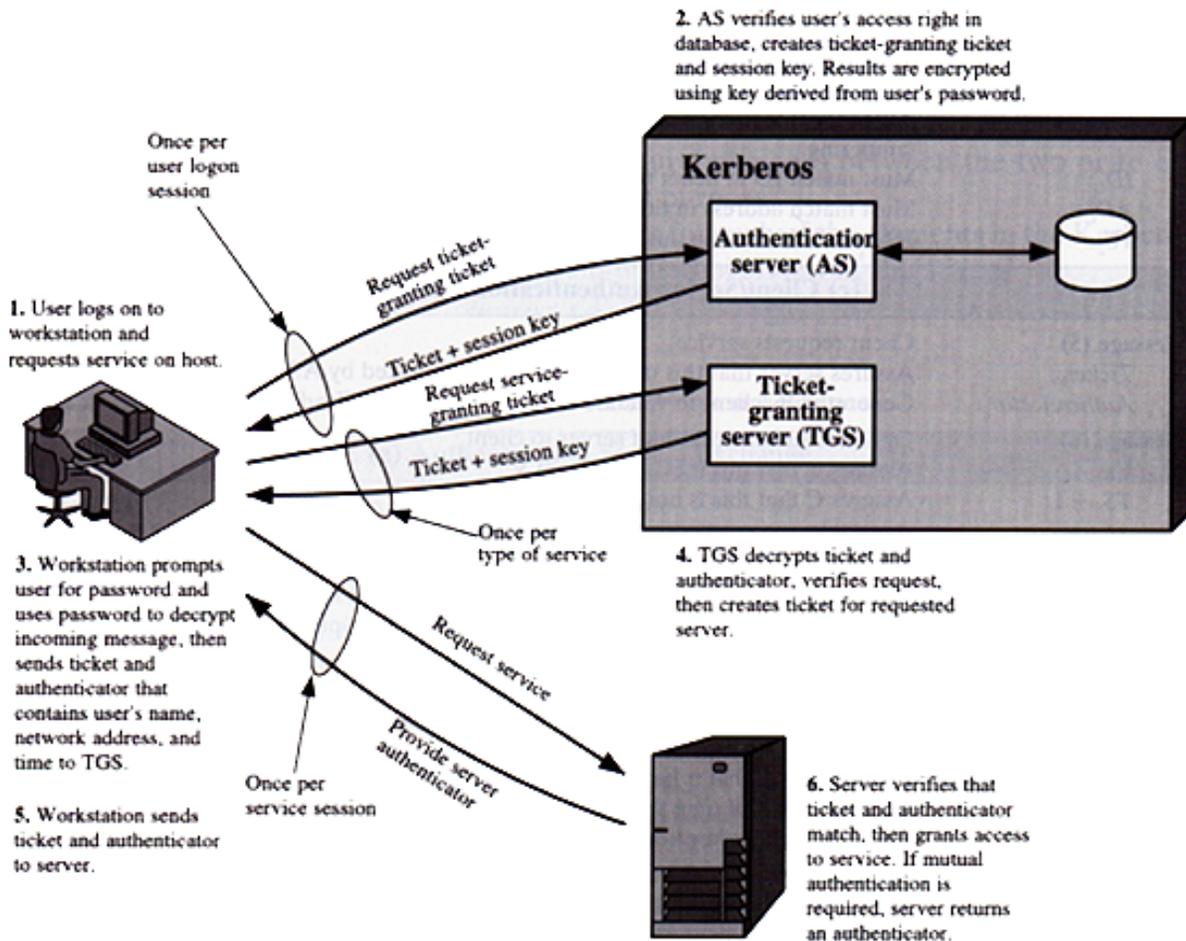


Figure 4.1 Overview of Kerberos

Supporting these requirements means Kerberos acts as a trusted 3rd party on the network, authenticating users and processes on servers concurrently. This scheme is secure if the server housing the Kerberos protocol(s) is itself secure.

X.509

ITU-T recommendation X.509 is a directory service framework for provisioning authentication services by the directory to its users. This directory may provide processes as a repository for public key certificates; it may also define parameters of protocols for authentication based on the use of public key certificates.¹

The X.509 protocol is applied in a variety of processes, including authentication of electronic text messaging, security for web-based information retrieval, and secure transactions over public networks. The protocol is based on the use of public-key cryptography and digital signatures (not yet mentioned). Recommendations are made as to what signature algorithms should be used to insure integrity of network processes, and assumed a specific hash algorithm will be used in authentication processes. Initial recommendations in 1988 have been updated in 1993 revisions to enhance hash algorithm security issues.

Electronic Mail Security

Pretty Good Privacy (PGP)

One of the first publicly-available security packages was PGP, which provides authentication and confidentiality for electronic mail and file storage applications. This is a widely used protocol incorporated into various email and network processes consisting of five processes:

1. user authentication
2. confidentiality
3. data compression
4. email compatibility, and
5. segmentation of mail length for long messages.

PGP is and always has been freely available for use by anyone, with commercial versions now being incorporated into products offered by various vendors including Network Associates (formerly Viacrypt). PGP has been ported to variety of platforms including UNIX, Windows, Macintosh and others.

PGP incorporates very secure cryptographic algorithms into a general-purpose application that is not platform-specific, and is based on small sets of commands for efficient operation.

The algorithms used are secure and have gone through substantial public review, including algorithms for public-key encryption, symmetric encryption, and hash coding. PGP is now on an Internet standards track RFC 3156 for personal email applications.

S/MIME

Secure/Multipurpose Internet Mail Extension is an enhancement to the MIME format standard used throughout the world for text messaging purposes. It is being considered as the primary industry standard for secure messaging in commercial and organizational applications. Parameters of S/MIME were original described in a number of documents, as RFCs 2630, 2632 and 2633; these have now been superceded by RFC s 3369 and 3370.

Text messaging was one of the first uses for the early versions of what we now know as the Internet. ARPANet used some simple standards (SMTP) defined in 1982 in RFC0822 to define message headers and content formats; the MIME standard used today evolved out of this original protocol and confronted issues in the following areas:

- SMTP's inability to transmit executable files and other binary objects
- non compatibility with language characters not based on the 7-bit ASCII code
- size limitations on mail messages
- mapping issues on SMTP gateways resulting in data translation problems

- message formatting issues as placement of carriage returns, truncation of line characters, removal of white space, padding of messages, and tab character conversion.

MIME parameters are outlined in RFCs 2045 through 2049.

S/MIME is designed to add security to the existing MIME protocol with a number of functions and features being developed. It is similar to PGP in its ability to add signatures to and encrypt text messages using existing and proven cryptographic algorithms. It adds features as Enveloped Data; Signed data; Clear-signed data; and Signed and Enveloped data functions. It uses 3 public-key encryption algorithms, which can be chosen or determined by the Sender depending on network configurations and receiver abilities. It also expands the number of content types as described in MIME, enhancing its ability to deal with digital signatures, encrypted data, and message objects.

Internet Protocol Security

In the early 1990s there was a call for security technologies for distributed systems using the Internet. In a report issued by the Internet Architecture Board some of the security concerns listed included unauthorized monitoring and control of network traffic, secure end-to-end user communication, user authentication schemes, and encryption mechanisms. The technologies listed below address these concerns.

IPSec

IPSec allows for secure communications on a variety of network scales, including LANs, private and public WANS, and the Internet. IPSec can encrypt and/or authenticate all traffic on the IP level, allowing it to be used in a variety of applications.

IPSec can be implemented in a variety of ways on a network:

- installed on a network firewall or router, IPSec can secure all traffic in & out of the network device
- on a users system, it can provide transparent security for peer-to-peer and network traffic
- can be implemented for remote users requiring access to confidential areas of a larger network
- also can be used in the overall routing architecture for data flow on a large network.

IPSec is described a several documents, RFCs 2401, 2402, 2406 and 2408. The features describe are supported in the existing version of Internet Protocol and in proposed versions currently being evaluated.

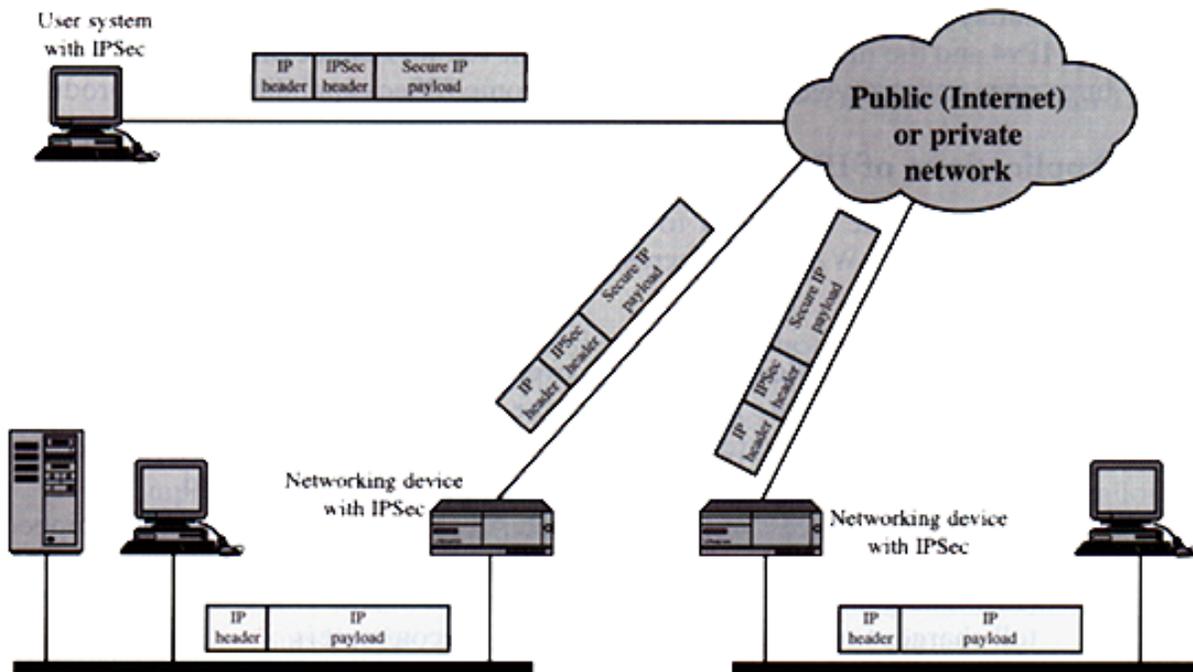


Figure 6.1 An IP Security Scenario

IPSec allows for selection of security protocols and encryption algorithms on a network, and the use of any cryptographic keys to be used in communication. Security associations can be established, where by combinations of security and encryption processes may be combined for specific network or communication requirements. IPSec allows different modes of data transmission, including 'standard' transport of data and tunneling between users or systems network devices. It also allows data to be encapsulated in secure 'envelopes' for transmission over public networks. Lastly, IPSec allows the use of different security keys for users and systems that have specific or desired security key requirements.

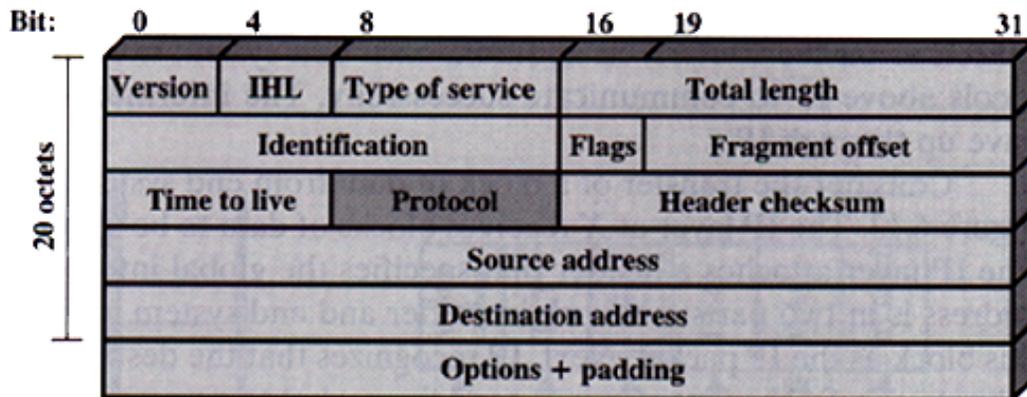
Internet Protocol (IP)

Although not (strictly) a security mechanism, an overview of Internet Protocol is included here, as it is utilized by many of security mechanisms currently in use or proposed for future implementation.

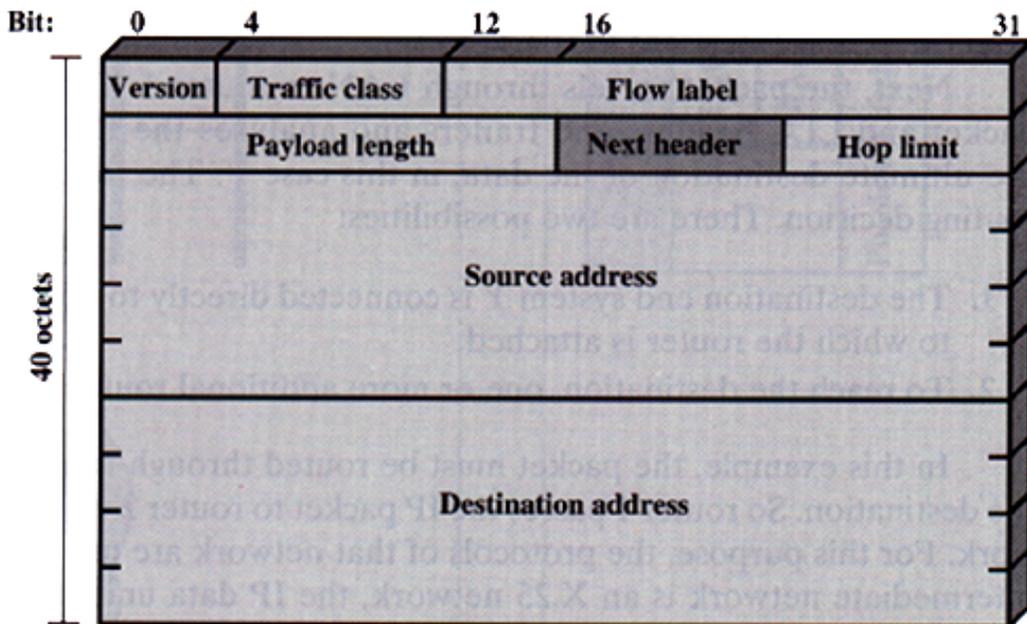
IP is implemented on the Network layer of the OSI Reference model for data transmission across networks. The role of IP in networking is to provide the functionality for interconnecting end systems in intranet- and Internet-based network environments. Ipversion4 is the current standard that has been in use since the establishment of the Internet as we know it at present. IPv4 architecture includes a header appended to each packet of data flowed in & out of a network with a variety of information, allowing this packet to traverse different kinds of network devices and systems on its way to a final destination. One of the specifications in IPv4 is the use of a 32-bit Addressing Scheme for the final destination address of the data packet; this scheme restricts the total number of possible IP addresses to a number now considered restrictive, considering the explosive growth of servers and web sites on the Internet in the last decade. Also, the IPv4 header consists of 20 octets (160 bits) of information contained in 12 different fields, adding (potentially) significant payload to the size of packets being

transmitted across a network. IPv4 is considered an unreliable protocol, in that it cannot guarantee the delivery of all packets it transmits to their final destination.

A new IP specification, version6, has been proposed to address security and addressing concerns of IPv4. IPv6 includes a 128-bit addressing scheme, allowing for support of over 3.4×10^{38} possible IP addresses. It also promises to operate faster by reducing the number of fields in its header to 8 in a larger 40-octet field, increasing processing speed of packets by routers and other network devices forwarding data. IPv6 also allows for extension headers to be added to packets for additional functionality.



(a) IPv4 header



(b) IPv6 header

Figure 6.14 IP Headers

Implementation of IPv6 universally is expected to take years, if not decades, and is a large scale implementation challenge for network developers, equipment manufacturers, and network administrators.

Web Security

A variety of technical and cultural events have taken place that have brought network, specifically Internet and World Wide Web, security to the forefront. One is the large amount of electronic commerce and financial transactions taking place online, requiring authentication of users and integrity of data across networks. Another is the large amount of network attacks taking place by knowledgeable computer 'hackers' for malicious purposes.

Two main general-purpose schemes have emerged for web security, discussed below.

Secure Socket Layer (SSL)

SSL is a set of protocols created by Netscape that operate above the Transport Control Protocol (TCP) in the OSI Model. It can be embedded in an underlying suite of protocols on a network for transparent use by clients, or embedded in commercial applications like a web browser. Both Netscape Navigator and Microsoft Internet Explore have SSL as part of their browser package, with a variety of security options available for users to define in their Preferences.

SSLv3 is the accepted version that has undergone public review and consists of two layers of protocols, whose purpose is to make TCP a reliable end-to-end secure service on public networks. The protocols include: Handshake, Change Cipher Spec, Alert, and Record protocols. SSL defines a Session state and Connection state for each client/server request, and utilizes different encryption algorithms and user keys, dependent upon the technical abilities of the client and server configurations.

Transport Layer Security (TLS)

TLS is the Internet standard that was proposed after the review and adoption of the SSLv3 protocol by the Internet community. The first working version of TLS is the equivalent of SSLv3.1, initially defined in RFC 2246 in January 1999.

A number of differences exist between SSL and TLS, including the header fields which are employed and the way in which data is encrypted in the TLS protocol. Numerous additional alert codes are included, the MAC algorithm is different, there is the use of a Pseudorandom function to expand data block for use in key generation and validation, and message padding is executed differently. These differences highlight the need defined for more secure transactions for both clients and merchants online, and the ability to insure data integrity in complex transactions on the Internet.

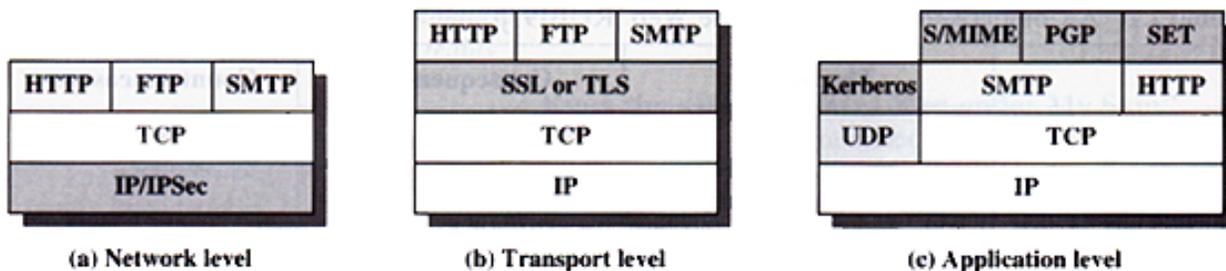


Figure 7.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

Secure Electronic Transaction (SET)

- SET is mentioned here as an outgrowth of the developments of SSL and TLS for secure credit card transactions on the Internet. SETv1 was developed in response to requests in 1996 for security standards by the Mastercard and VISA companies, and involved a number of computer-related companies in the initial specification including IBM; Netscape; Microsoft; and Verisign. SET was issued in May, 1997 and described in a lengthy specification consisting of three books and a total of 971 pages (!).

SET is a set of security protocols that allows secure communications between all parties involved in an online transaction, as buyer, seller, certificate authority, financial institution and transaction processor. SET guarantees confidentiality for both buyer and merchant, integrity of data transmitted during the transaction, credit card and merchant authentication. The protocol can operate over a 'raw' TCP/IP connection but does not interface with the other security protocols listed above.

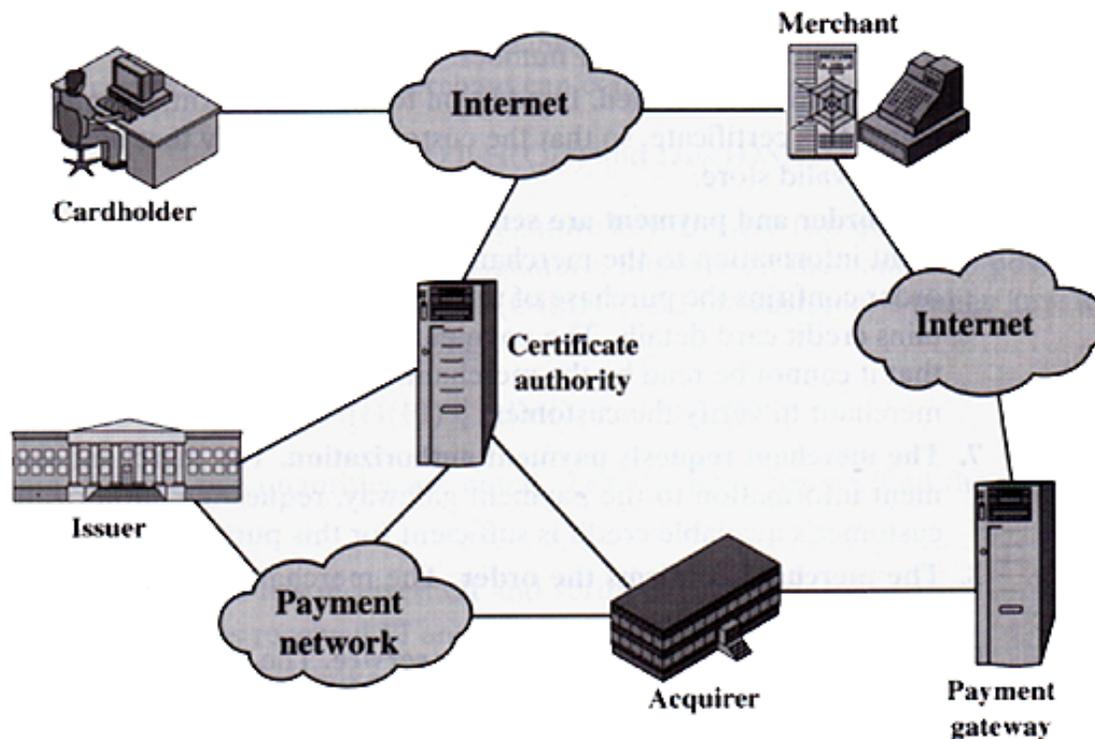


Figure 7.8 Secure Electronic Commerce Components

One major difference between SET and SSL/TLS is the use of a single 1024-bit cryptographic algorithm, due to its application using a single set of security requirements. Unique to SET is the use of a Dual Signature process to link messages to their respective recipients, eliminated the chance for fraud during a transaction. SET is utilized by numerous financial institutions, credit card processors and merchants for verifying online transactions, due to its global interoperability and support for a variety of payment schemes including credit card, debit card, and chip cards.

Network Management Security

Security of local area networks and larger scale distributed systems requires more than human monitoring and other efforts; specific network-based tools are needed for automation of security

tools. Standards for network management have been developed that include services, protocols and management functions – the most widely used standard is the Simple Network Management Protocol (SNMP).

Version1 of SNMP was introduced in 1988 and has since gone through two revisions. V2 incorporated enhanced functions to the basic SNMP protocol, and the current v3 included security enhancements for today's sophisticated network architectures. To summarize, SNMP is designed to provide overall network monitoring and security services by utilizing specialized software residing on host computers and communication processors on the network. Feedback from each network element to a central monitoring center computer(s) allows the network to be viewed & monitored as a unified system, with each active element constantly providing feedback on its status.

The key elements of this management scheme are:

- Management station(s)
- Management agents
- a Management Information Base (MIB)
- Network management protocols.

A management station may be a single administrator's computer located at a central site that interacts with 'agents' installed on various network components. These agents may be located on routers or other network communication devices, satellite manager workstations, or single user computers located on the network. The MIB consists of a collections of managed objects that contain data concerning all aspects of each managed object; this MIB may be located on a central machine, or be decentralized and distributed across a network at key access points. The management protocols provide links between the managements station and its agents that are providing feedback on the status of network components, located primarily on network communication devices.

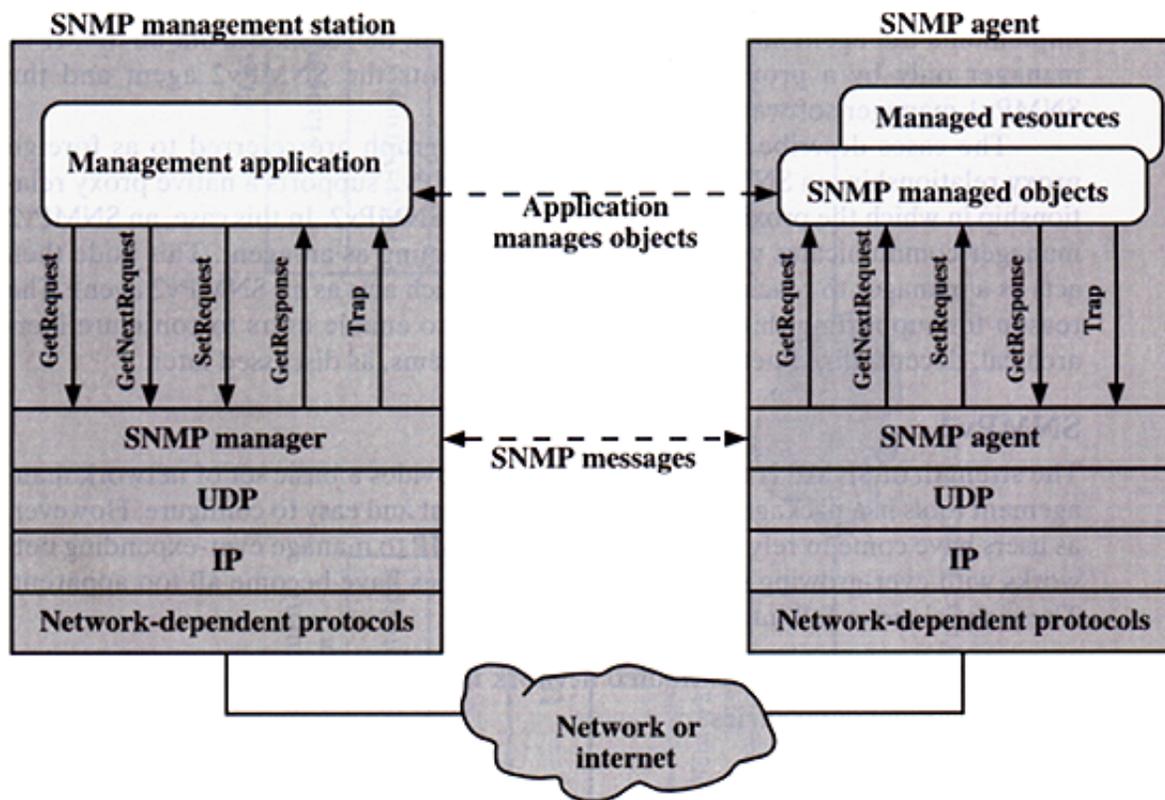


Figure 8.1 The Role of SNMP

SNMP is designed to be an application-level protocol, operating in concert with the TCP/IP protocol suite. It also operates over UDP (User Datagram Protocol) and a variety of network-dependent protocols as Ethernet, FDDI and X.25 . Components that do not use TCP/IP are managed using proxy devices on the network, utilizing SNMP agents for communication and monitoring. This allows management and security to be extended and scaled to a large degree.

SNMPv2 expands the protocols over with SNMP can be implemented, allowing greater management of network configurations that utilized various network components. Specific network issues v2 was designed to address include lack of support for management on a distributed network, functional and security deficiencies.

SNMPv3 addresses many security issues relevant to large scale networks that utilize the Internet for network traffic. v3 is not a stand-alone replacement for previous versions of SNMP, but instead defines security capabilities to be using in conjunction with v1 or v2. Thus, the functions and services of earlier implementations of SNMP can be utilized with the enhanced security features of v3 on different aspects of network functions or architecture. In an application sense, "SNMPv3 can be thought of as SNMPv2 with additional security and administration capabilities".

Wireless Network Security

There are two general classifications of wireless networks: Wireless Local Area Networks (WLAN)

and Wireless Wide Area Networks (WWAN). These networks are used for Internet access for home and remote office locations, and for sharing data between offices dispersed through a local or metropolitan geographical area. Wireless deployment is attractive as a cost-effective alternative to landline networks, and provides greater range than many other network technologies, especially in rural areas.

As with landline networks, security measures are required to insure that data integrity is preserved for both users and network hosts. The security issues center around:

- data transmission, and
- wireless equipment security.

Most forms of fixed wireless Internet access use the Radio Frequency (RF) spectrum to transmit data. This is a broad range of microwave frequencies used by many devices, and requires line-of-sight from the antennae to the receiver to function. The two most popular forms of data transmission are Multichannel Multipoint Distribution Service (MMDS) and Local Multipoint Distribution Service (LMDS). Both can operate at a range of frequencies, licensed and unlicensed; these ranges of frequencies are designed to allow a start and end frequency range the data can be transmitted through without interference. MMDS has been primarily deployed for rural television subscribers, although it is restricted to limited channels and thus a small subscriber base, but can be used for a wireless LAN effectively. LMDS operates at higher frequencies, thus offering more bandwidth, and is primarily deployed for two-way communications as telephone, television and Internet access.

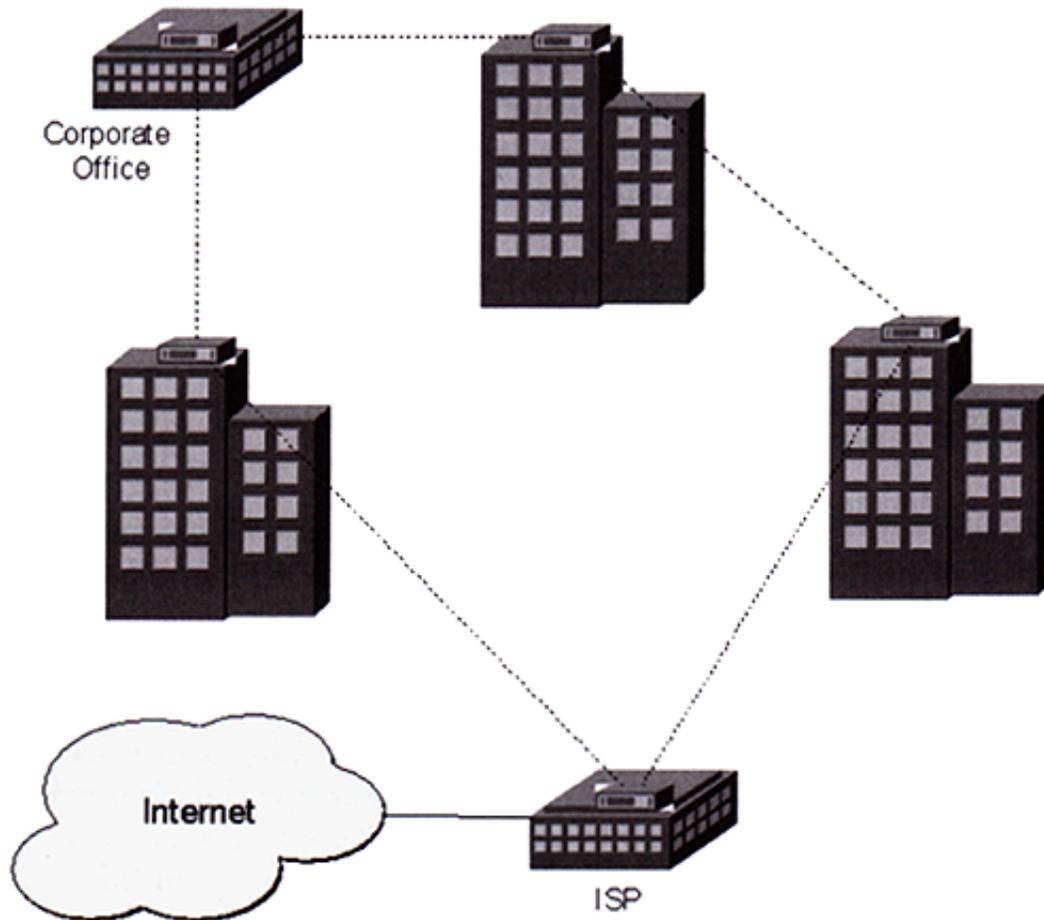


Figure 8.2 Using multiple MMDS antennas to provide network redundancy

WWAN

Data Transmission

Wireless data transmission occurs through airwaves, making sniffing and intrusion inherently easy. Wireless data encryption, therefore, is one method of insuring data integrity between sender and receiver. Many wireless equipment manufacturers have developed systems of encryption based on the Data Over Cable Service Interface Specification, called DOCSIS+ or wireless DOCSIS.

DOCSIS+ supports several types of key management encryption including X.509 digital certificates, RSA public key encryption, and 3-DES encryption. An ISP using wireless to provide Internet access sets the encryption policy used at its wireless modem, and users are required to comply with whatever standard is used. This type of deployment a) prevents intruders from randomly intercepting traffic, and b) prevents unauthorized users from attempting to access the ISP's Internet backbone through the wireless connection. The drawback to DOCSIS+ is its current interoperability between manufacturer equipment; as wireless technology is further developed this interoperability issue should decrease and disappear.

Additionally, Spread Spectrum Technology (SST) is commonly deployed for data encryption over fixed wireless connections. SST is a wideband RF technology that converts narrowband signals to

wideband signals for transmission to receivers modems; this wideband signal is then converted back to narrowband and the packets pieced together on the receiving end. This data conversion provides for a scrambled signal when transmitted, and would appear as noise to an intruder randomly scanning wireless signals. Thus, a conscious effort is required by an attacker to gain knowledge of the frequency, equipment and encryption technology being used to intercept and decipher SST-altered data.

There are three types of commonly deployed SST:

1. Frequency-hopped spread spectrum (FHSS)
2. Direct-Sequenced spread spectrum (DSSS)
3. Code-Division Multiple Access (CDMA)

FHSS provides for a signal to be hopped across different frequencies at predetermined intervals to make interception difficult; commonly frequency switching intervals are no longer than 400 millisecond intervals. If switching intervals are known by both the sending and receiving wireless modems, this method is effective in securing data transmission.

DSSS is similar to FHSS, but performed in a different manner. In DSSS, the signal from the originating modem is combined with a higher rate bit sequenced called a chirping code; this code is then spread across the transmission spectrum in a manner similar to FHSS. This chirping code provides for data redundancy and makes transmission less susceptible to atmospheric interference. DSSS is generally used for higher-speed fixed wireless connections, while FHSS is deployed for lower speed connections.

CDMA is used extensively in cellular phone networks. This technology distributes the signal across frequencies as FHSS, but uses fewer frequencies. Security is increased with CDMA by adding noise to the signal and by digitizing the data, allowing for direct transmission between sending and receiving wireless modems. These two aspects of CDMA allow modem manufacturers to offer higher bandwidth than the other two types of SST.

Equipment Security

Security of wireless equipment is mainly location-oriented; wireless modems that are placed in locked closets near network routers and other network elements can be better controlled than one easily accessible on a roof or other location. An attacker who has direct access to a wireless modem and obtain its type and model may be able to more easily gain access to the device and data it is transmitting. Additionally, if a coaxial cable is used to connect the wireless modem to a network device this provides a potential point-of-entry for an attacker who has access to the modem location.

As with other network equipment, default passwords on wireless WAN equipment should be changed upon installation.

WLAN

Wireless Local Area Networks are generally deployed as an extension of an existing wired network, providing network access to multiple users in remote areas not easily wired or cost-effectively configured for standard network access. There are some fundamental security issues that arise when

extending a network with wireless technologies, as data integrity and secure access.

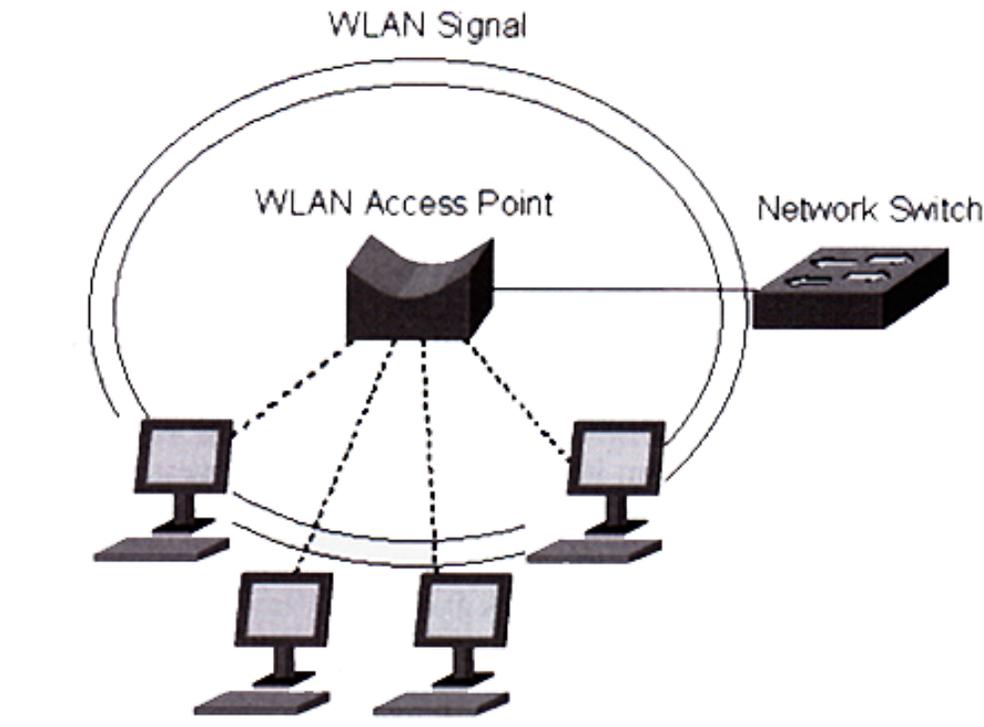


Figure 9.1 A typical WLAN design. Multiple workstations connect to an access point, which plugs into the network. The access point forwards traffic to and from the workstations and the rest of the network.

Access

The generally accepted standard for WLAN connections is the IEEE 802.11 guidelines. There are currently three different standards in 802.11, operating at different frequencies and data transfer rates. 802.11 is based on 802.3, allowing WLANs to be implemented as extensions of existing networks. 802.11 primarily defines specifications for the physical layer and MAC address portions of wireless Ethernet.

Access security issues of WLANs are the same as wireless WANS: an intruder does not have to be a part of a network physically to gain access to data and potentially secure parts of a network. "Drive-by hacking" can be done outside of a building or across the street from buildings that contain wireless network components (depending on the coverage area of a wireless antennae). Data on a wireless LAN can be sniffed using tools readily available, allowing an attacker to bypass traditional firewalls and other security features to get onto the network.

Unlike microwave, wireless antennae coverage is omni-directional, emitting traffic in a 360 degree circular pattern. An intruder outside a building with a laptop and wireless network card can extend the reach of his equipment through amplified omni-directional antennas; this allows intrusion into the network outside the physical premises the network is contained in. Security for a wireless LAN should start with this initial point of access on a network.

Access points generally rely on HTTP, Telnet and SNMP protocols for establishing connections and transmitting data. As Telnet and HTTP transmit in clear text, these should be disabled on remote access points. SNMP also has security issues and should be replaced with secure protocols as HTTPS or SSH.

There are a variety of layer security features that can be utilized for wireless LANs:

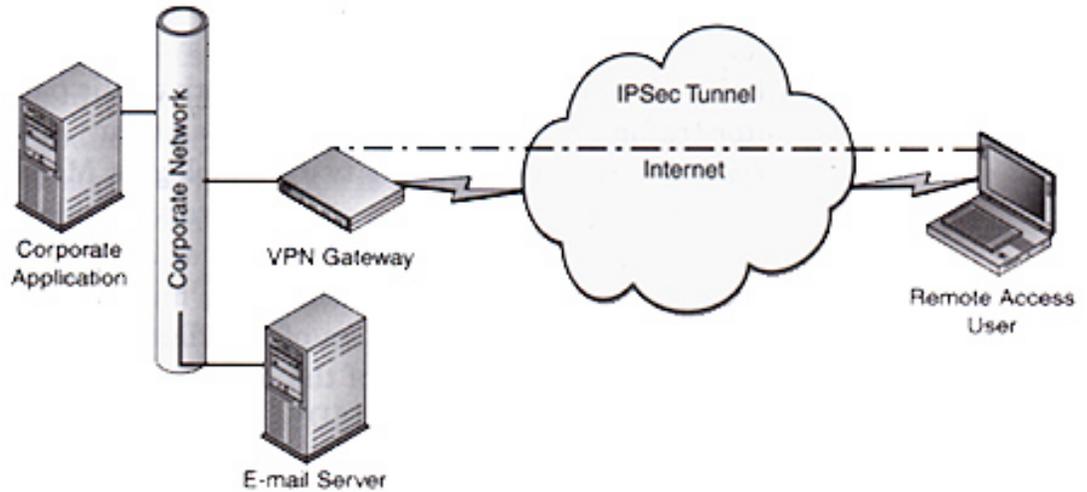
- **SSID - Service Set ID**
a 32-character unique ID that is attached to the header of packets sent over a wireless network; acts as a password when a wireless user attempts to connect to a set of wireless stations on a network.
- **WEP - Wired Equivalent Privacy**
a cryptographic mechanism designed to provide confidentiality and authentication for data transmitted over wireless networks, considered to be flawed in its implementation.
- **MAC Addresses - Medium Access Control Addresses**
MAC addressing is often used on wireless networks at access points; by generating a list of authorized network components that can access a wireless LAN, an administrator can monitor network traffic for unauthorized traffic by component prefix.
- **RADIUS - Remote Authentication Dial-in User Service**
it is possible to implement RADIUS username & password authentication for each access point on a wireless LAN, in addition to SSID and WEP; this provides an extra layer of security for wireless LANs. This service is offered commercially by several vendors but is technically challenging and usually expensive.

WLAN VPNs

Wireless tunneling works extremely well in many environments¹¹. This would allow a remote, wireless user to create a secure connection to a main land-based network and pass or access sensitive data without fear of intrusion. In some cases this can be accomplished by adding a wireless NIC to the VPN gateway on a network, or changing rules to firewall configurations. IP addresses must remain the same for the system to work in a secure manner.

There are several commercial software solutions supporting this technology for remote users of Palm and PocketPC devices, including ColumbiTech Wireless VPN. Their "Session Resume technology maintains the same session and VPN tunnel even when the wireless signal fades out, eliminating the need to re-login into another VPN tunnel. Transaction Recovery technology enables the wireless device to pick up an interrupted download right where it left off before interruption. This is executed with maximum security through authentication, strong encryption, and PKI support. Features such as compression, data reduction, and seamless roaming enhance performance and convenience further.¹² "Recent tests show a small increase in load being placed on the network when using a wireless VPN versus a terrestrial line VPN.

VPN architecture



VPN Network architecture

802.11i

802.11 is a series of protocols designed to increase security on wireless networks before layer 3 protocols (such as IP) are set up¹¹. This technology is not specific to wireless networks and can be implemented on other topologies as Ethernet and Token Ring. The various implementations of 802.11x vary in their network data rates, frequency spectrum and the data distribution technology used in wireless applications.

802.11 has some serious security issues, some discussed previously in this paper. A 802.11i Task Force was established to research ways to reduce its security vulnerabilities in wireless applications. New encryption key protocols for 802.11a and 11b are provided the improvements included in the Temporal Key Integrity Protocol (TKIP). TKIP is the next generation of WEP, the Wired Equivalency Protocol, and provides 128-bit temporal keys, fast packet keying, and key management in an attempt at fixing the flaws of WEP.

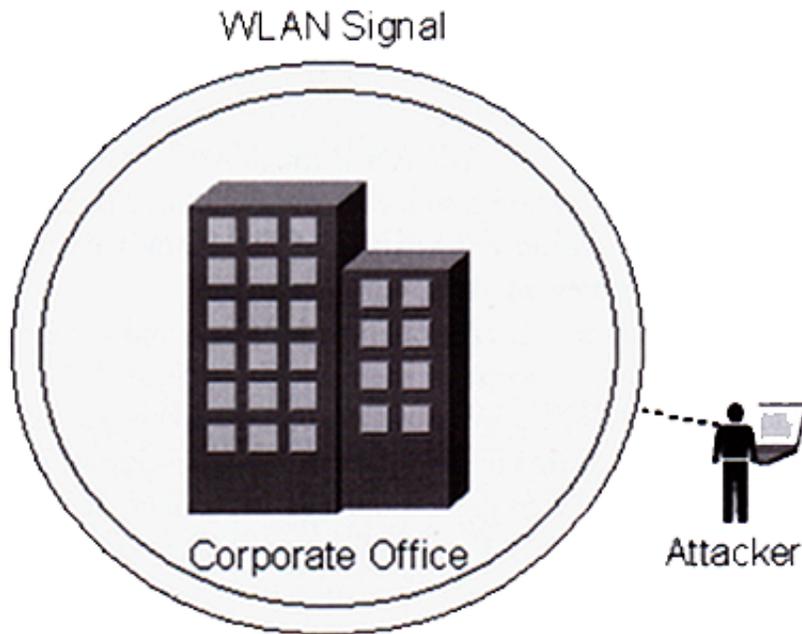


Figure 9.2 An attacker is able to use a WLAN sniffer to monitor data on the wireless network, even though the attacker is not attached to the network

TKIP is considered a short-term or temporary fix for wireless security that is compatible with existing standards. Its testing period has been shorter than normal in an attempt to provide a working product quickly. Further wireless security in 802.11i is expected with the addition of Advanced Encryption Standard (AES) data encryption; this implementation of 802.11i will require an upgrade in network hardware compatible with the AES standard.

802.1x is an attempt to allow for automatic updating of access codes by wireless network administrators, provided a higher level of security against data and connection sniffing. The WiFi Alliance of companies is actively supporting the use of this protocol for wireless devices.

Network Security Systems: Host

Network Intrusion

One of the most significant security threats to emerge is that of hostile attacks on computer systems connected to the Internet. User trespass can take the form of unauthorized access, data modification or retrieval; software trespass can take the form of introduction of virus, worm and other malicious software.

Intruders

User threats have become highly publicized; three classes of intruders can be defined:

- Masquerader: a user who exploits someone else's network access privileges to access system controls and data, usually from outside the system.
- Misfeasor: an authorized user who misuses existing system privileges to access confidential data and resources, usually from inside the system.
- Clandestine user: a user who either seizes control or gains administrative access to a system for malicious purposes, either from inside or outside the system.

Adversary	Goal
Student	To have fun snooping on people's email
Hacker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military strength
Terrorist	To steal germ warfare secrets

Fig. 7-1. Some people who cause security problems and why.

One of the major uses of a system by an intruder is to route information or gain access to another network through unauthorized access of a primary network; this allows the user to hide their identity and location. Another use is to gain access to password and other confidential files, allowing the intruder to further access and extend their privileges on other parts of the system.

Intrusion detection and defense are two methods of dealing with would-be hackers and crackers. Password files can be encrypted and/or placed in highly secure areas of a network difficult to attack, restricting access and making them difficult to decipher. Intrusion detection and monitoring is more difficult to achieve; there are numerous methods developed and strategies that can be taken to detect unauthorized access.

Motivation for and benefits of intruder detection include:

- collection of information about intruder techniques and patterns that can help in detection processes
- establishment of a deterrent for future attacks
- creation of a quick-detection process that allows for intruders to be identified and ejected from the system before doing damage.

Porras⁷ defines two main approaches to intruder detection: **a)** using statistical anomalies to observe system user behavior and recognize legitimate versus malicious behavior, and **b)** defining a set of rules for system behavior that can be applied to users, allowing intruders to be identified based on use or misuse of these rules. In reality, a combination of these two approaches is most effective in combating unauthorized access to a network system.

Passwords

The password system is the primary line of defense against unauthorized access on a system. The login procedure usually consists of two components: a) a unique name or identifier, and b) a secret password associated with that username. These two components serve to authenticate a user on a system, and authorize them to use the system based on privileges applied to the user account.

Many login schemes use encryption methods to scramble password input, then store these encrypted passwords in a file along with a 'salt' value used for encryption. This encryption helps to deter intruders from using guessing attacks when trying to hack into a system. Some of these schemes are based on original DES algorithms, with newer implementations of the algorithm increasing password security and decreasing the amount of time it takes for the system to create and store the password. The increase in speed of host system hardware also aids in the encryption and storage of dynamically-generated passwords when users login.

On many systems using Unix-based machines, password vulnerabilities exist that can be exploited by an intruder. First, a user that has gained unauthorized access to a legitimate user account might run a password guessing program to crack into encrypted password files. If this program utilizes more sophisticated decryption algorithms than the host machine uses for encrypting passwords, then password files can be compromised in a reasonably short period of time. Secondly, if a password file can be accessed and copied to another machine outside of the network, an intruder can utilize sophisticated decryption tools that are undetectable to the network being attacked, allowing for deeper intrusion after the password file has been deciphered.

Length of user passwords and password forms are directly related to the level of security of user login schemes. Passwords of 6 or more alphanumeric characters are much harder to decipher than shorter passwords. Also, many users created easily-guessed passwords that contain names, numbers and other characters associated with that user or popular; if an intruder has access to some confidential information about that user, then a simple or common password can be guessed (often

manually) by the intruder. Password guessing programs that contain commonly-used passwords in their reference file often guess the correct password in reasonable amounts of time afforded to the intruder.

Table 9.4 Passwords Cracked from a Sample Set of 13,797 Accounts [KLEI90]

Type of Password	Search Size	Number of Matches	Percentage of Passwords Matched	Cost/Benefit Ratio ^a
User/account name	130	368	2.7%	2.830
Character sequences	866	22	0.2%	0.025
Numbers	427	9	0.1%	0.021
Chinese	392	56	0.4%	0.143
Place names	628	82	0.6%	0.131
Common names	2239	548	4.0%	0.245
Female names	4280	161	1.2%	0.038
Male names	2866	140	1.0%	0.049
Uncommon names	4955	130	0.9%	0.026
Myths & legends	1246	66	0.5%	0.053
Shakespearean	473	11	0.1%	0.023
Sports terms	238	32	0.2%	0.134
Science fiction	691	59	0.4%	0.085
Movies and actors	99	12	0.1%	0.121
Cartoons	92	9	0.1%	0.098
Famous people	290	55	0.4%	0.190
Phrases and patterns	933	253	1.8%	0.271
Surnames	33	9	0.1%	0.273
Biology	58	1	0.0%	0.017
System dictionary	19683	1027	7.4%	0.052
Machine names	9018	132	1.0%	0.015
Mnemonics	14	2	0.0%	0.143
King James Bible	7525	83	0.6%	0.011
Miscellaneous words	3212	54	0.4%	0.017
Yiddish words	56	0	0.0%	0.000
Asteroids	2407	19	0.1%	0.007
TOTAL	62727	3340	24.2%	0.053

^aComputed as the number of matches divided by the search size. The more words that need to be tested for a match, the lower the cost/benefit ratio.

There are a number of strategies that can be deployed to increase protection of user passwords:

- denying access to centralized password files – this places the burden on system security and has a number of vulnerabilities, including the use of the same password for access to multiple systems. This gives the intruder multiple chances to discover a user password.

- increasing security via password selection – establishing password selection schemes that avoid the common pitfalls, as passwords that are too short, increases security dramatically. User education is one key aspect of increasing password security.

Malicious Software

Better known as viruses, malicious programs are those that exploit vulnerabilities in a network system. Included in this overall grouping are applications programs, utilities, editors and compilers. Viruses can be grouped into two general categories:

- malicious programs, and
- software viruses.

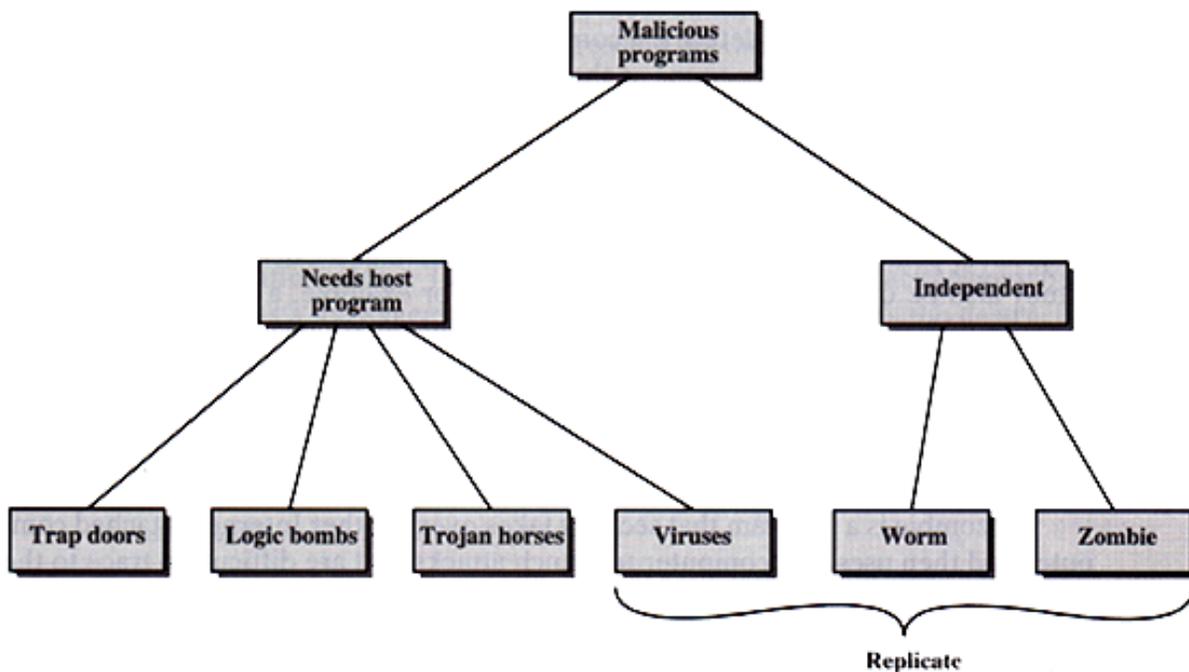


Figure 10.1 Taxonomy of Malicious Programs

Malicious Programs

Software threats can take two different forms, depending on their functionality: a) those that need host programs to propagate, and b) those that are independent executable applications. Additionally, there are those programs which replicate or spread, and those that are standalone programs that work locally on a user's machine.

Malicious programs can be categorized into the following areas:

- **Trap Doors**
These are secret entry points into programs, utilized by software engineers and network administrators to monitor and manage password records, access lists, and the like. Trap doors

are used on a regular basis by programmers and analysts to test and debug programs easily. These doors provide potential security risks, as intruders may hack a user's system to find a trap door, which might allow an unauthorized user access to sensitive material or gain entry to system for some other purpose.

- **Logic Bombs**

These are some of the oldest and simplest kinds of program threats. Bombs are code residing in legitimate programs that are set to implement or 'explode' when a set of conditions exist on a user's machine, as a particular date; the presence or absence of certain files; or a particular user running an application. Bombs may cause files to be deleted, render applications useless, or cause a machine to cease functioning.

- **Trojan Horses**

A piece of apparent useful procedure or computer code inside of a program might in fact be a Trojan Horse masquerading as a legitimate piece of software. One example is a compiler that unsuspecting inserts code into a program being developed by a user that is copied to a hard drive or propagated when the final program is launched on user's machine. This might then allow a malicious user access to any machine the legitimate piece of software was installed on.

Trojan horses can also be used to destruct data on a user's machine a malicious user has gained access to. A small program or utility containing malicious code could be downloaded and used for some purpose by a user; when implemented this code could then be used to destroy sensitive or necessary files on that user's machine.

- **Zombies**

Much of the email spamming and denial-of-service attacks in recent years utilize Zombie programs. Such a program is introduced into a networked computer through a download or email attachment, then copies itself to multiple machines connected to this host on a network. This Zombie allows a user to enter a network system, and send data through or out of the network through one of the machines connected to the host. Thus, the identity and location of the potentially malicious data is masked, hiding the location and identity of the intruder.

Software Viruses

Viruses work by infecting or modifying other programs, affecting computer operating system functions. Viruses reside on hard drives and other storage mediums, and propagate by copying their malicious code to other programs they come in contact with. Viruses can get passed to multiple machines on a network when users exchange files, either over a network or by exchanging disks. Accessing files or programs on other machines over a network is a perfect environment for viruses to spread.

Virus programs typically have four phases in their lifetime:

1. Dormancy – preoperational idle time prior to being activated;
2. Propagation – spreading of malicious code by copying itself to other programs;
3. Triggering – activation through a trigger the software recognizes and responds to. This could be

the launching of a program the malicious code is attached to, the number of times it has been copied, or some other particular event;

4. Execution – the actual function the software is designed to perform. This may be erasing files, copying itself to other sections of a hard disk, or searching for other files to infect.

There are several different categories of viruses, with new additions added by malicious software developers on a continual basis:

- Parasitic – the most common form of virus, a parasite attaches itself to an executable file (as another program) and replicates itself to other executables when the host program is launched.
- Memory Residence – this virus resides in the main memory of the host machine and infects each program that is launched.
- Boot sector – a virus residing in the master boot sector or record of a disk can launch itself and infect files it comes in contact with when a machine is booted from this disk.
- Stealth – this type of virus is designed to hide itself from virus detection software, thus making it hard to find on a host machine.
- Polymorphic – a virus that mutates each time it is replicated, making its signature extremely difficult to detect or trace.

In recent years, there have been many forms of sophisticated viruses spread through compute networks. The Macro virus was developed to infect Microsoft program files that utilized the Macro function available in many of the Office Suite of products. Since macros are used to replicate common tasks, a virus attached to a file using a macro would execute whenever that particular macro were invoked by the user. This type of virus could attached itself to a word processing or spreadsheet file, and be transferred to different machines by any person using that file.

Email viruses usually come in the form of an attachment, some executable file that would launch and execute when the receiver opens the attachment. A new form of email virus, or malware, is one that uses features of the local email client to execute; such a virus could propagate by spreading to all the addresses contained in the email client address book, sometimes within a matter of hours. This type of virus is hard to prevent due to its rapid propagation.

A Worm is a virus that spreads by network connections, usually in a email attachment, and then replicates itself on each destination machine it arrives at. A worm could also execute itself remotely, by making a copy to another machine with a know address across a network, or automatically perform a remote login to machines on a network by performing a user name & password guessing routine, then spreading itself to machines successfully accessed. These types of viruses are also very hard to detect, due to their sophisticated nature and stealthy appearance.

Advanced Virus Protection

Recent developments to combat malicious network software attempt to detect and eradicate viruses of unknown origin & structure before they enter a system.

Generic Decryption

GD technology attempts to detect and eradicate sophisticated polymorphic viruses by creating a contained operating environment on a central or dedicated machine in which to identify and explore the potential virus. Such a package would contain a cpu-level emulator that would run the malicious executable file, protecting the underlying operating system and processor functions from infection. In this environment, the module scans the suspect code, looking for known virus signatures, and controls the execution of the code in the contained operating environment. In a network environment, valuable resources may be consumed by this process and slow down user productivity on a system.

Digital Immune System

Developed by IBM, this is a comprehensive approach to dealing with the propagation of Internet-based viruses. This system builds on the Generic Decryption approach by providing an emulation environment in which to examine malicious code, but at an increased speed. When suspect software is introduced into a network, this system automatically captures the virus, analyzes it, adds detection and shielding for it, eradicates the code, and then informs all other machines on the system about the virus through its IBM AntiVirus package, effectively eliminating the virus's ability to replicate through a system⁵.

Behavior Blocking Software

- Unlike the previous two examples, behavior-blocking software integrates with operating systems of machines on a network and operates in real-time to monitor suspicious behavior. This approach negates the need for sophisticated heuristics or fingerprinting techniques, and allows the software to watch for any malicious event, as opening, modifying or deleting files; reformatting of disk drives or operations; modifications to executable files, scripts, and OS-critical files; alteration to text messaging clients to generate or send executables; and initiation of suspicious network communications.
- The drawback to this approach is its defensive posture; a virus may be able to invade, replicate or alter seeming innocent files on a local system before it is detected by the blocking package.

Network Firewalls

Firewalls are hardware and software applications that allow traffic to outside networked resources but restrict and monitor traffic flow into a local area network. A firewall acts as a centralized routing, monitoring and detection system that allows network traffic control to be implemented from one or more centralized points. Firewalls exist between public networks (the Internet) and secure areas of a LAN or WAN, and acts as an exterior perimeter security system to protect confidential data and resources. A hardware firewall may be a single computer or series of machines working together to audit and control network flow. Individual network components may have their own software applications that act as firewalls to inhibit suspect traffic in&out of the component.

Firewalls can be described by function, in four general categories which are described below:

- Service control: the primary function of most firewalls; determines the services and areas of a secure network a user may access.
- Direction control: determines the direction of data traffic a specific authorized service may use on the network (either in, out or within a network).

- User control: outlines the services a user may access on a network, primarily from inside the firewall; also applies to remote users accessing parts of the network located inside the firewall.
- Behavior control: determines how particular services may be used on the network, as email or server-based programs.

Because network intrusion can take multiple forms, effective firewalls are not usually one single device based at the perimeter of a network, monitoring & controlling all traffic in and out of a network. Rather, a system of network components and workstations may work together to provide a variety of firewall-type security features to accommodate the various types of network attacks.

There are three common types of firewalls: a packet filtering router, application-level gateway, and circuit-level gateway. The packet filter allows traffic to flow in & out of a network based on information contained in individual data packet header, as source and destination address and protocol used. An application level gateway, sometime referred to as a proxy server, acts as a relay for network traffic on an application level, as a Telnet or FTP application; the gateway acts as a mediator or relay point from source to destination to route the packets on the network in a secure manner. A circuit level gateway does not permit end-to-end connections from source to destination, rather the gateway sets up individual TCP/IP connections between source and destination to route traffic between authorized users.

There are numerous configurations and implementation schemes of firewalls on a network, depending on security needs, the equipment and services utilized on the system. Some configurations are elaborate, others less complex and costly. Regardless of complexity, basic perimeter security using firewalls is one of the best defenses against invasion of secured resources on a network.

Trusted Systems

The concept of a trusted system is a different approach to network security; many of the protocols and methods discussed are applicable in protected individual pieces of data or information. A Trusted System, on the other hand, is a way of providing security based on levels of access and permissions granted to users on a network. That is, confidential data may only be accessed by certain members of an organization; more general use data is available to a larger group of users; and so forth.

This multilevel security is based on the notion that no confidential information will flow 'downward' in the permissions tree to any unauthorized user. There are two fundamental rules a multilevel secure system must enforce:

- no Read Up – a user can only read an object of less or equal security level (the Simple Security Property)
- no Write Down – a user can only write into an object of greater or equal security level (the *-Property).

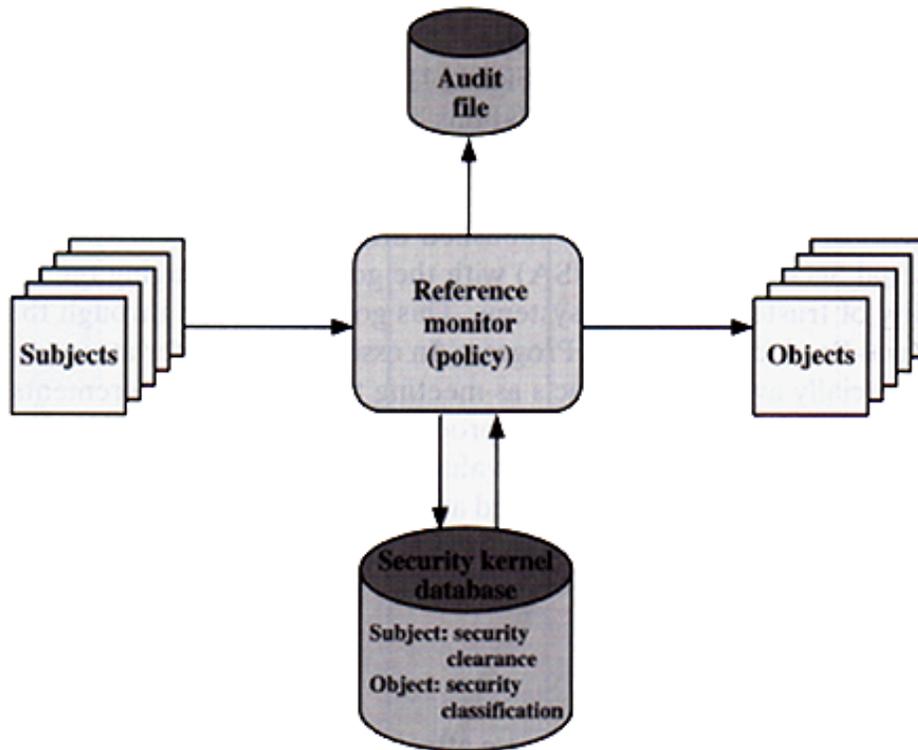


Figure 11.4 Reference Monitor Concept

The Reference Monitoring concept is a popular approach to providing multilevel security in a data processing system. A central file, the security kernel database, provides access parameters for users on a network and the objects they can access. This reference monitor enforces the security rules and has the following properties:

- complete Mediation – security rules are enforced for every action on a network, not single actions as opening a file;
- Isolation – the database and monitor are secured entities, protected against unauthorized modification;
- Verifiability – it must be possible to prove the correctness of the monitor mathematically, to insure security rules are being enforced on the network.

A system that enforces the security rules using the described properties is referred to as a Trusted System. These properties are effectively obtained using both software and hardware solutions on a network; use of software alone to enforce security rules can place too high a processing load on the network for practical use.

The National Security Agency (NSA) provides information and guidelines for commercially available security products through its Commercial Products Evaluation Program.

DMZs

A DMZ is essentially a perimeter network that separates public data from secure data. The DMZ uses a variety of firewalls on different layers of the network to isolate public traffic areas from the secure (private) data. There are a variety of configurations for DMZs, that relate to the topology of the existing network and the security needs of the servers on this network. The common use for a DMZ is to isolate the secure part of the network from that part that is open for public access, and allow different security rules to be applied to each network segment.

There are debates about the best way to implement a DMZ; it is commonly agreed that separating public and private data is a good idea. The question of how to do this in a cost-effective manner without imposing a high security load on the network is at the core of the debate.

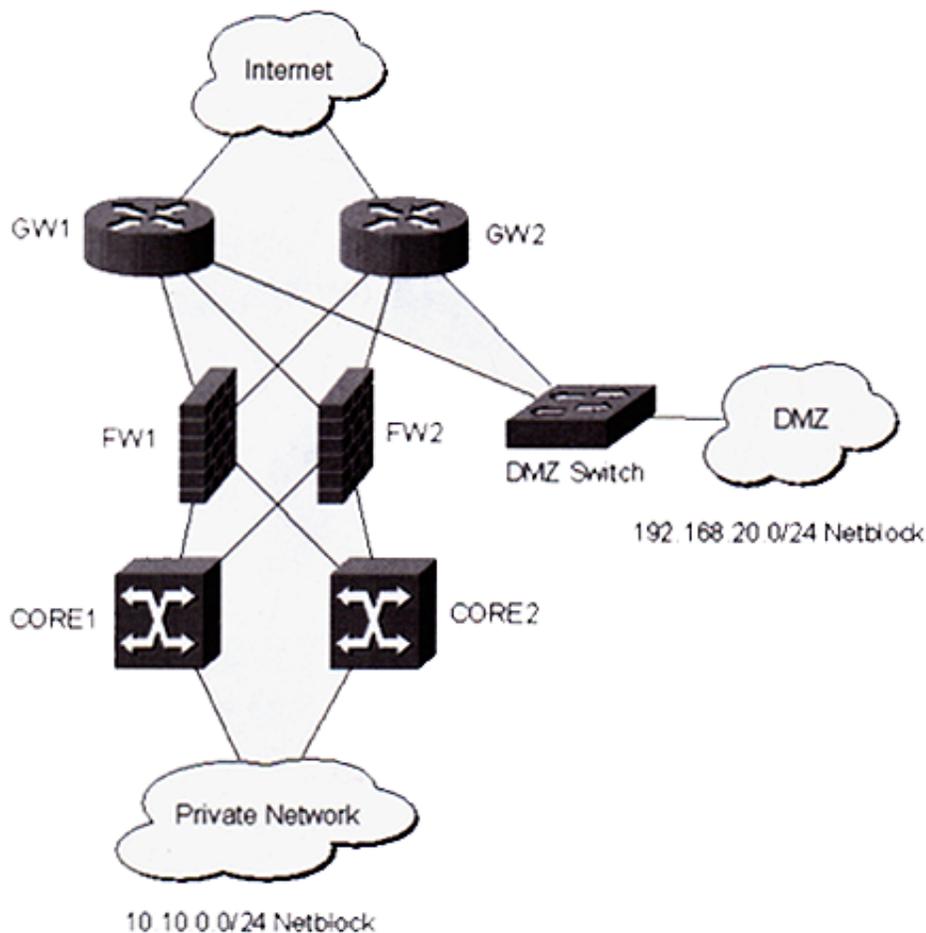


Figure 11.1 A traditional DMZ design: The DMZ network terminates at the router, and none of the DMZ servers are protected by the firewall

In a traditional DMZ, public server access is gained through a separate switch which directs all traffic to the DMZ area. The DMZ would contain servers for web traffic, for instance; incoming traffic is controlled by the two main gateways, which allow all http traffic destined for port 80 to be routed to the DMZ switch. Thus, all web traffic essentially bypasses the firewalls protecting the private segments of the network.

Though easy to implement, the security of the public servers in this scheme is thin, relying solely on security rules that can be implemented on the DMZ switch. The advantage of this configuration is the simplified rule set and monitoring of traffic on the public servers. And intruders breaking into the public network have no access to secure aspects of the network; there is no direct connection between the public and private sides of the network.

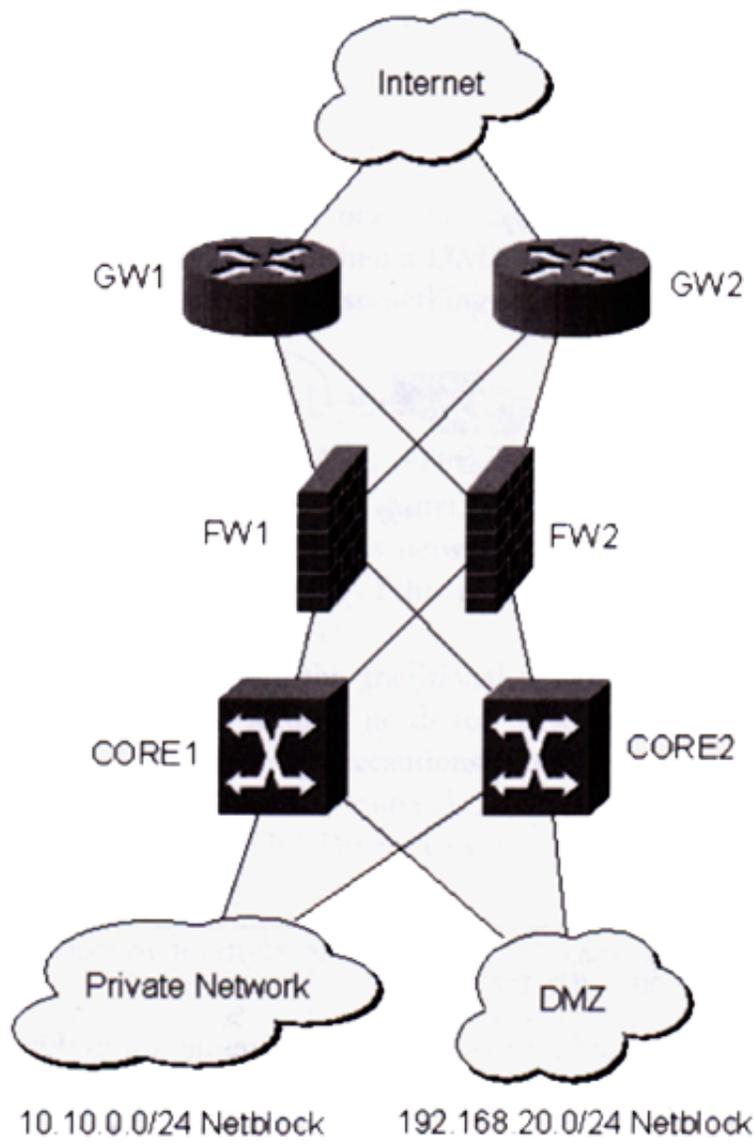


Figure 11.2 A DMZ design using the firewall as the terminating point for the network. The firewall secures and isolates the DMZ traffic, which travels within a separate VLAN from the private network.

Another method of implementing the traditional DMZ scheme is to terminate the public network on a second firewall interface. This configuration allows for the public servers to be protected by firewall security, allowing traffic to a web server, DNS server, and mail server to be isolated from other secure data but still controlled with basic security schemes. The problem with this scheme is the same as

above, there is no way of sharing data between the private and public portions of the network. This is a problem if information on the secured database server is shared with the web server, for instance; there is also no way of securing data that is passed on the network between these two servers.

One resolution to this problem is to put another set of network interface cards in each of the servers, and to direct traffic between the servers through these cards. However, this traffic bypasses the front-end firewall thus negating any security schemes in place on the firewall.

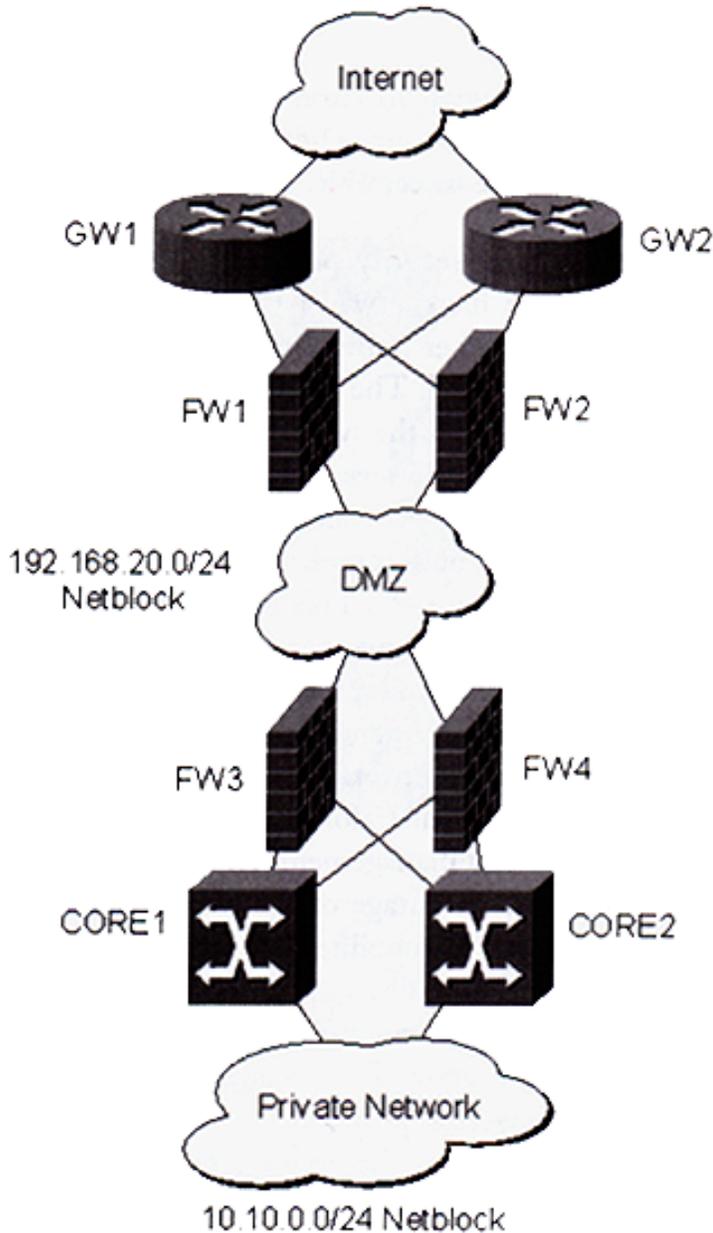


Figure 11.4 Increase network security by adding a second firewall and creating a DMZ that is truly isolated from the network

Other DMZ implementations include dividing the network into different firewall layers, allowing traffic through initially security schemes to public areas of the network, and isolating private data

behind secondary firewall layers. This allows public traffic on the top layer efficiently with little security overhead, and allows further access to private data via secondary network cards installed on specific machines in the DMZ. This allows servers in the DMZ and private network to share and update data in a secure manner, and allows for more robust security to be implemented on the private network. Additionally, intruders gaining access to the public network must go through an additional layer of security to access more sensitive data. This scheme is most effective when different types of firewalls are used on the two layers.

A further implementation of the separate firewall layer scheme is to use a secondary firewall back up to a management network, with all network traffic passing through an initial firewall layer and traffic between the private and public networks passing through the secondary firewall layer.

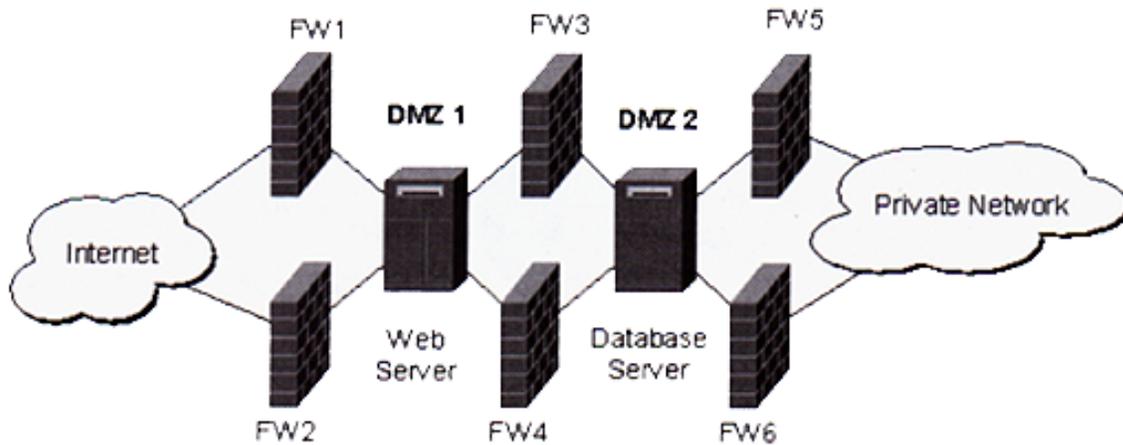


Figure 11.6 A multiple DMZ network, in which the database server is isolated from the private network. To ensure the integrity of the database server, a separate set of firewalls is added, and a DMZ is created.

Finally, the multiple firewall scheme can be extended further, separating servers on the network behind individual firewalls. This scheme would allow control of incoming network traffic, and allow for servers to share data in a secure manner. Public servers with less traffic restrictions can be located near the perimeter of the network, with more sensitive data located further into the firewall layers, with the most restrictive data located on the private network at the 'bottom' of the firewall scheme. This design allows for individual firewall security management, and multiple types of firewalls with different security schemes to increase protection of sensitive data.

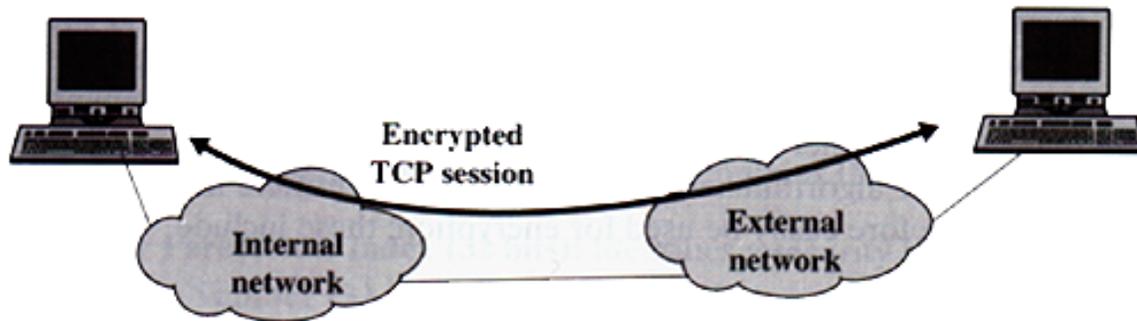
The disadvantage to this design is the complexity of the routing and security rules, as well as the physical management of multiple firewalls. The complexity of this type of scheme also creates the potential for security breaches and holes if not configured correctly. Additionally, the load placed on the network to reach highly secure data on the private network is increased, resulting in reduced access speeds.

Another implementation of this multiple firewall scheme creates separate DMZs for individual servers, isolating public servers on their own private network. This scheme guards against intrusion into multiple public servers through their isolation, but creates the need for complex firewall configurations and security rules.

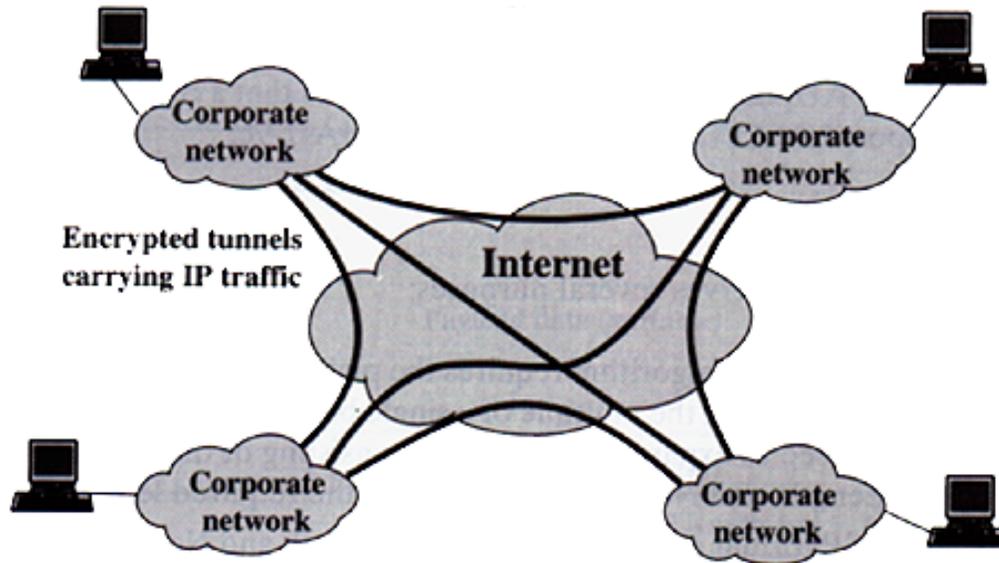
Virtual Private Networks

IPSec authentication can be used to create secure connections to a network for remote users needing access to confidential data. There are two general methods of accomplishing virtual circuits using IPSec:

1. Transport mode – authentication between a server and client workstation is accomplished using a protected secret key;
2. Tunnel mode – a remote workstation authenticates itself to a network firewall, gaining access to network resources.



(a) Transport-level security



(b) A virtual private network via tunnel mode

Figure 6.8 Transport Mode versus Tunnel Mode Encryption

These two methods make use of the authentication header information used by IPv4, which allows for authentication of all packet information that is transferred between workstation and the network. The new IPv6 will also support VPNs, although the way in which the authentication header is inserted and used is slightly modified from v4.

VPNs are an excellent method of allow remote users access to protected areas of a network, using public networks. This reduces the cost of using dedicated lines for remote users, while allowing authentication of users and protection of data. VPNs can be established for remote users through the use of protocols and permission settings on a network and a users workstation.

Honeypots

A recent innovation in intrusion detection is the use of decoy systems, part of a network left open intentionally to lure potential intruders away from other critical parts of a network. An effective honeypot would consist of fabricated data that appears valuable but not normally accessed by users of the network; this system would also contain extensive monitoring and logging functions to allow monitor access to this system and to collect data about intruders. This in turn allows administrators to learn more about security holes in a system, and methods used by attackers on their system.

Honeypots are designed to:

- divert attackers away from critical data and systems on a network
- collect information about attackers and their activities
- encourage an attacker to stay logged in to a system for extensive periods, allowing for collection of data and monitoring of their activities.

Initial research in honeypot usage involved a single computer using a fixed IP address to attract

intruders. More recently, large enterprise-type of systems have been configured to resemble networks used by corporations and educational institutions, sometimes with actual data and simulated traffic to convince an attacker of the authentication of the system. Once logged in, the behavior of the intruder can be observed and logged, providing information on how to prevent intrusion into the 'real' network system and ways to counter intrusion methods in the future.

There is an obvious expense in setting up and maintaining a honeypot system that may not be cost-effective for small networks. For larger enterprise-type of systems containing sensitive data, however, the use of a honeypot on the outside of a secure network may be a cost-effective way of deterring intruders and providing an initial buffer against attacks on critical systems.

Intrusion Detection Exchange Format

The IETF Intrusion Detection Working Group is developing standards for intrusion detection systems operating on different platforms and system environments. These standards would provide for common data formats and exchange procedures for sharing information on intrusion detection and response systems, and ways to manage the systems that interact with intruders. These standards are to include:

- a common intrusion language specification, describing data exchange formats
- a requirements document outlining functional guidelines for communication between intrusion detection systems and management systems
- a framework document that identifies protocols best suited for communication between different intrusion detection systems.

These standards are currently in an Internet-draft development stage.

Network Security Systems: Client

Remote Connections: SSH, sFTP, WebDAV

SSH: The Secure Shell

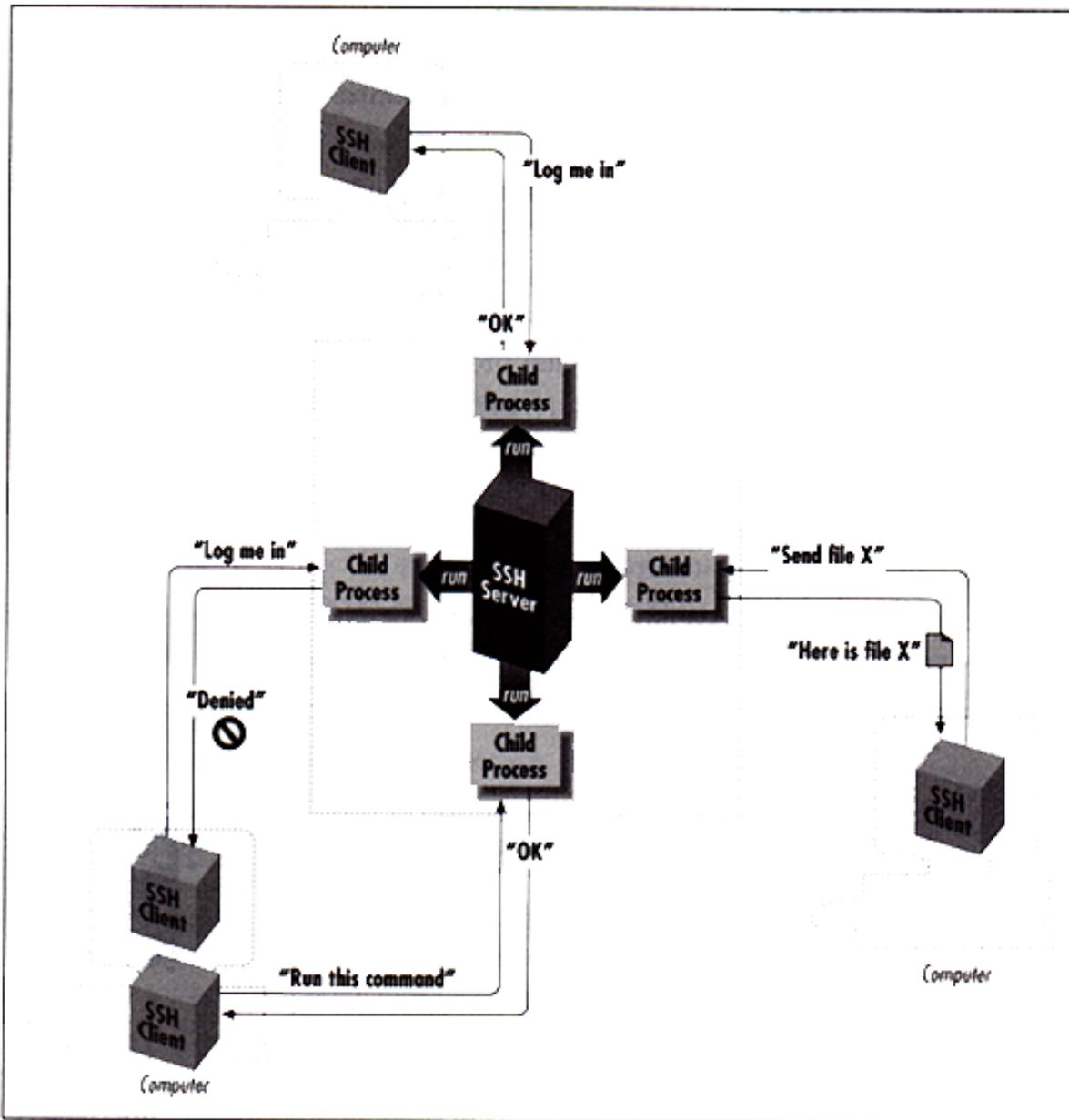


Figure 1-1. SSH architecture

SSH is a software solution for securing the transmission of data across a network, and can be deployed for use by remote users needing access to secure data on a network. Its major features include:

- a secure client/server protocol for encrypting and transmitting data over a network
- authentication of users by password, host or public key, and optional integration with other

authentication systems including Kerberos, PGP and others

- add security to unprotected network applications as Telnet and FTP¹⁰.

SSH is a protocol, not a standalone commercial product. SSH operates transparently to the user, and can be implemented on many operating systems including Windows, Linux, Unix and Macintosh. There are several versions, with the current version designated as SSH-2.

SSH consists of a client and server component, allowing for secure connections to be established between user and network machine. When logging in, a connection is first created between the client and the server using usernames and passwords. Usernames and passwords can be authenticated in an encrypted manner, and then data transmitted in a secure manner once user authentication is verified. This type of security is useful when uploading files to a server from a remote location, i.e. remote access and file transfers to a web server.

SSH protects against password detection or sniffing via a known-host mechanism; both the client and the server authenticate each other to establish a secure connection. This prevents a 3rd party from masquerading as a SSH host or intermediary ISP and intercepting passwords sent by the client. This is done through the use of a host key or ID unique to each SSH server, which is copied to a client's computer when logging in to the server. Each time the client connects to the host this key is checked and verified before a connection is established.

SSH utilizes both private and public keys for identification. The private key is unique to each client, the public key known to all clients and placed on the SSH server. During authentication, the server and client verify each other's keys to provide identities.

Open SSH

Open SSH is a free implementation of SSH-1 and SSH-2 in open source code, available for downloading. It was originally designed for the BSD Unix operating system, and has been modified to include platform support for Linux, Solaris AIX, IRIX, HP/UX, FreeBSD and NetBSD. A portable version implemented on various Unix platforms is available and tracks the development of the main OpenSSH development team. Open SSH requires Open SSL and the zlib package to operate; both are available online.

sFTP: secure FTP

sFTP is a separate file-transfer tool layered on top of SSH¹⁰. Developed by SSH Communications Security, it was originally available only in SSH-2 with other implementations appearing since its release. Some of these sFTP products run over SSH-1 and other utilize SSH-2, depending on their design and application.

Using sFTP for remote file transfers is advantageous for several reasons:

- it utilizes the SSH secure channel for data transfer
- a single sFTP session can include multiple file copying and modification commands
- sFTP can be scripted using the ftp command language
- sFTP will sometimes substitute for unsecure FTP applications running in the background on networks.

webDAV: web-based Distributed Authoring and Versioning

WebDAV is a set of extensions to the HTTP protocol included in several web authoring and upload utilities³³. Use of webDAV allows groups of users to collaborate on web development using a web server as a file server; authors can login securely to volumes on a server and perform modifications to files in a secure manner. This technology is platform independent and requires both a client and server component to operate.

WebDAV provides numerous server-side working tools for the web author including the ability to lock a file when being worked on so duplicates are not created; it also provides for secure 'namespace operations' on the server including the ability to copy and move files from a remote location.

WebDAV provides security and authoring protection in several manners; webDAV leverages HTTP header parameters to provide functionality. In HTTP v1.1, method parameter information is exclusively encoded in the HTTP headers. Conversely, webDAV encodes method parameter information either in an Extensible Markup Language (XML) [REC-XML] request entity body, or in an HTTP header. The use of XML to encode method parameters was motivated by the ability to add extra XML elements to existing structures, providing extensibility; and by XML's ability to encode information in ISO 10646 character sets, providing internationalization support³⁴. Current specifications are included in RFC 2518.

Several commercial products leverage webDAV security including the Dreamweaver MX web authoring tool, and the Goliath upload utility. Goliath release 1.0 provides enhanced SSL support including Client Certificate Support, support for adding trusted CA certificates, and updated to OpenSSL 0.9.6e for maximum security. The Goliath server component supports a number of server platforms and components including Apache mod_dav, Apple iDisk, Microsoft IIS5, and WebStar V.

Virus Protection

The best virus protection is to not introduce malicious software into a system; this is very hard to accomplish on a working network, with files and other constant data flow. Most viruses are introduced unknowingly to a system, and then can spread unnoticed.

A success approach to virus protection for individual users of a network is three-fold:

1. Detection – the ability to detect the existence of a virus and eradicate it before replication;
2. Identification – a system of periodic inspection and identification of suspicious data on local systems;
3. Eradication – once identified, a process for removing malicious software and related code from a system, essentially stopping the replication process.

There are four different kinds of anti-virus software a user can utilize on a local system, categorized by their chronological development and sophistication:

- 1st generation: a simple scanner that requires a virus signature for identification; they may also try to detect viruses through common code lengths.
- 2nd generation: a scanner that does not rely on a virus signature; rather, heuristic rules or some discovery process is used to search for possible virus presence. An example is looking for known code samples used in viruses on a local system.
- 3rd generation: these software packages reside in memory on a users machine, and look for known or suspicious activity to detect possible virus presence.
- 4th generation: sophisticated software packages that contain a variety of components and actions for virus detection; these packages may contain detection, eradication, and access control to identify and contain virus replication.

Selected aspects of these technologies and their application in network environments are explored in the following section.

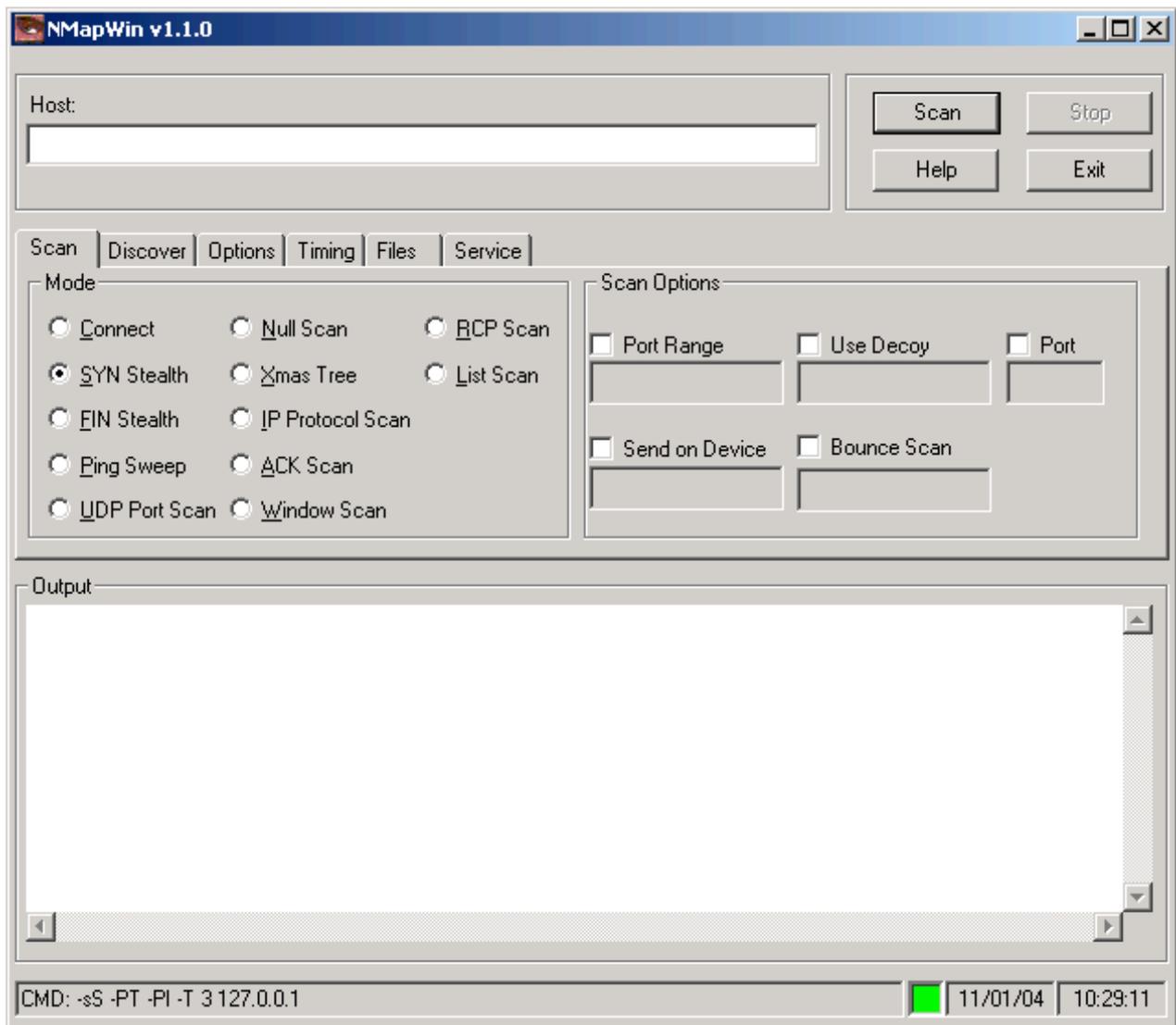
Testing

Port Scanning

Overview

Scanning or sniffing a network allows an intruder to determine what the state of vulnerabilities are that can be exploited. Often times a particular port or part of a network device is open to exploitation, but must first be found by the intruder. There are a variety of methods to do this, including writing small custom programs for this purpose, or using an already compiled program for sniffing.

One such program is Nmap; this program is available for a variety of platforms and has various options for scanning networks from remote locations. This program is used by network administrator to monitor their own network and discover any vulnerabilities that might exist for outside users. The program operates on IP addresses or URLs, and allows a variety of scanning options including UDP or IP Protocols; one can also do a Ping Sweep and ACK scan, and specify a range of port numbers on the network to scan.



Nmap will perform a scan on a destination IP address, and output information concerning open ports and the protocols used on those ports. The output below is a UDP port scan on a Macintosh computer on a local area network:

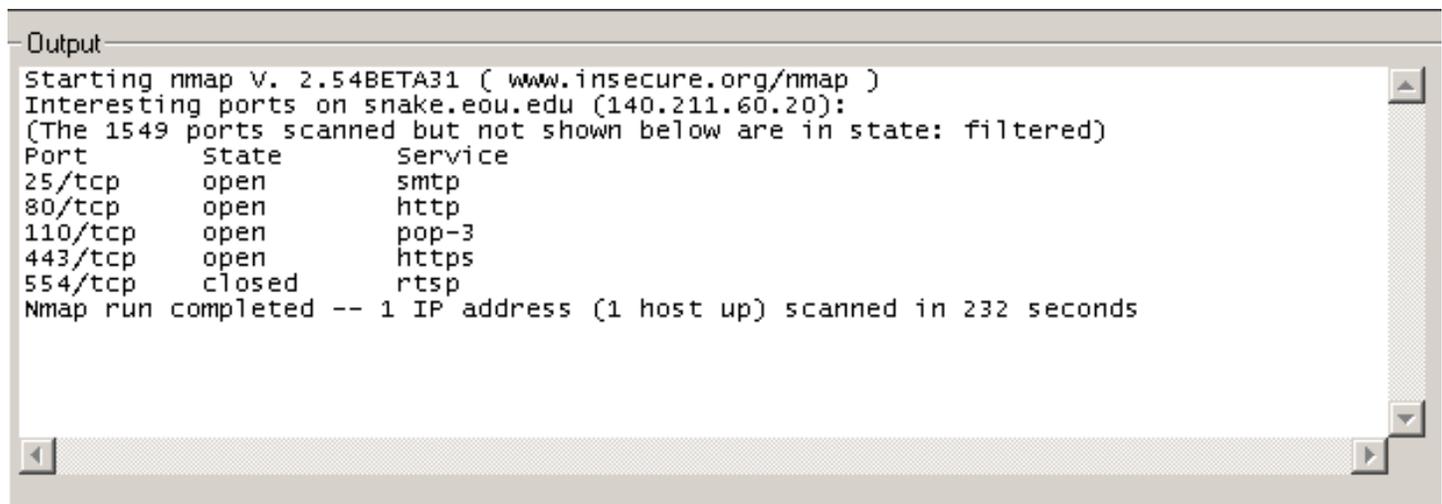


```
Output
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap )
Interesting ports on MACG3 (192.168.123.118):
(The 1024 ports scanned but not shown below are in state: closed)
Port      State      Service
53/udp    open       domain
68/udp    open       bootpc
123/udp   open       ntp
137/udp   open       netbios-ns
138/udp   open       netbios-dgm
427/udp   open       unknown
514/udp   open       syslog
1023/udp  open       unknown
Unable to find nmap-services! Resorting to /etc/services
Nmap run completed -- 1 IP address (1 host up) scanned in 17 seconds
```

internal UDP scan using NMap

This scan shows a number of ports open support UDP and the service (protocol) support by each. An internal network attack could use this information to gain access to this machine and confidential data on its drives.

A general scan can be done to look for open ports on remote networks to determine vulnerabilities in the network. The output below is a remote scan of the Snake server on the EOU network:



```
Output
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap )
Interesting ports on snake.eou.edu (140.211.60.20):
(The 1549 ports scanned but not shown below are in state: filtered)
Port      State      Service
25/tcp    open       smtp
80/tcp    open       http
110/tcp   open       pop-3
443/tcp   open       https
554/tcp   closed    rtsp
Nmap run completed -- 1 IP address (1 host up) scanned in 232 seconds
```

remote Nmap scan of snake.eou.edu

This output shows port numbers of the network that are in an open state, and the protocol running on those ports. An attacker might now be able to determine what tools to use for an attack, based on

the services running on the server, and try to exploit using additional tools.

Comments and Conclusions

Port scanning as a passive activity is fairly harmless, but is one of the first activities in a larger processor a intruder may use to gain unauthorized access to a network. It is virtually impossible to monitor this kind of remote activity; substantial monitoring resources are needed to be able to detect and follow port scanning activity.

Network Sniffing

Passive monitoring or 'sniffing' of traffic on a network is possible with various Sniffer programs. The purpose is to find vulnerabilities in network configurations or to gain access to data traveling across a network for some malicious purpose. A person with a sniffing program is working passively on a network, and thus is virtually impossible to detect.

Overview

A local area network test was done with the Ethereal software package running on a WindowsXP machine with Ethernet connections to a LAN router and DSL modem. Although well known, precautions were taken to not run this or other programs on the EOU network to avoid introducing any harmful virus or other malicious software into the network.

Ethereal allows for data capture for a virtually indefinite period and must be installed on a client machine inside the network to function. Below is a screenshot of Ethereal at work:

The screenshot shows the Ethereal network sniffer interface. The main window displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, and Info. The first packet is highlighted, showing it is an Echo (ping) request from 192.168.123.151 to 206.204.18.22. Below the list, the details of the selected packet are shown, including the Ethernet II header, Internet Protocol header, and Internet Control Message Protocol header. The packet data is displayed in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
2	0.012937	192.168.123.118	192.168.123.255	CUPS	ipp://MacG3.local./printers/_192_168_123_254 (idle)
3	0.066462	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply
4	1.023656	192.168.123.118	192.168.123.255	CUPS	ipp://MacG3.local./printers/Ip_on_192_168_123_254 (idle)
5	13.314732	192.168.123.118	224.0.0.251	MDNS	Standard query PTR _register._mdns._udp.local PTR _default.
6	20.848501	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
7	20.915114	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply
8	30.066923	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
9	30.123823	00:10:b5:0f:dc:85	ff:ff:ff:ff:ff:ff	ARP	who has 192.168.123.151? Tell 192.168.123.254
10	30.123837	00:10:b5:0f:dc:85	00:00:94:cf:3b:4b	ARP	192.168.123.151 is at 00:10:b5:0f:dc:85
11	30.124815	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply
12	30.533824	192.168.123.118	192.168.123.255	CUPS	ipp://MacG3.local./printers/_192_168_123_254 (idle)
13	31.535090	192.168.123.118	192.168.123.255	CUPS	ipp://MacG3.local./printers/Ip_on_192_168_123_254 (idle)
14	50.915423	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
15	50.967895	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply
16	60.124805	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
17	60.185565	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply
18	61.598897	192.168.123.118	192.168.123.255	CUPS	ipp://MacG3.local./printers/_192_168_123_254 (idle)
19	62.600170	192.168.123.118	192.168.123.255	CUPS	ipp://MacG3.local./printers/Ip_on_192_168_123_254 (idle)
20	80.968293	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
21	81.020821	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply
22	90.185700	192.168.123.151	206.204.18.22	ICMP	Echo (ping) request
23	90.238401	206.204.18.22	192.168.123.151	ICMP	Echo (ping) reply

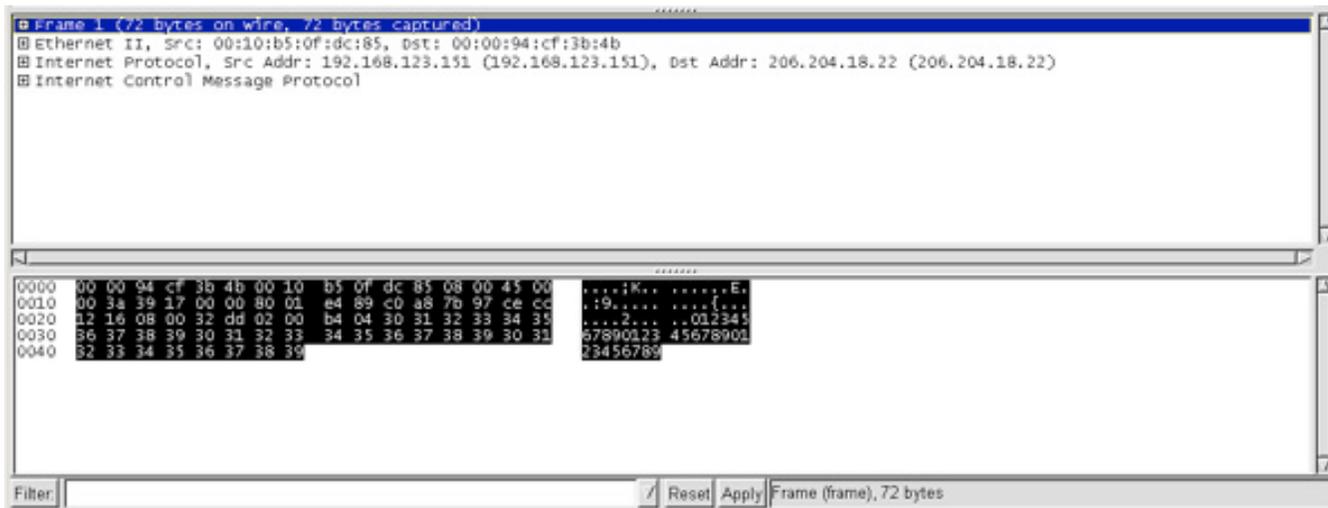
Frame 1 (72 bytes on wire, 72 bytes captured)
Ethernet II, Src: 00:10:b5:0f:dc:85, Dst: 00:00:94:cf:3b:4b
Internet Protocol, Src Addr: 192.168.123.151 (192.168.123.151), Dst Addr: 206.204.18.22 (206.204.18.22)
Internet Control Message Protocol

```
0000  00 00 94 cf 3b 4b 00 10  b5 0f dc 85 08 00 45 00  . . . . .K. . . . .E.
0010  00 3a 39 17 00 00 80 01  e4 89 c0 a8 7b 97 ce cc  .19. . . . .
0020  12 16 08 00 00 32 d8 02 00  b4 04 30 31 32 33 34 35  . . . 2. . . . .012345
0030  36 37 38 39 30 31 32 33  34 35 36 37 38 39 30 31  67890123 45678901
0040  32 33 34 35 36 37 38 39  23456789
```

Ethereal Network sniffer

The top frame contains packet information flowing across the network, the source and destination addresses, the protocol transmitting the data, and general comments of the type of traffic. The Frame

window (below) shows statistics on an individual packet frame, along with the actual packet data. If the packet data is not encrypted it is possible to read the actual contents of the packet that traveled across the network.



Packet content in Ethereal

There are a variety of other tools contained in the Ethereal package, including decoding functions; protocol hierarchy, response time, IO and RTP analysis statistics; and various display settings for analyzing data captured.

Comments and Conclusions

Sniffing programs are fairly harmless in a passive state; use of unencrypted data in a malicious manner, however, is possible for a person using a machine on the network. The information could be passed to users outside the network, and vulnerabilities discovered for accessing sensitive data. For example, it could be possible for a person to sniff a monetary transaction on an unencrypted network connection and obtain credit card and other financial information on the user.

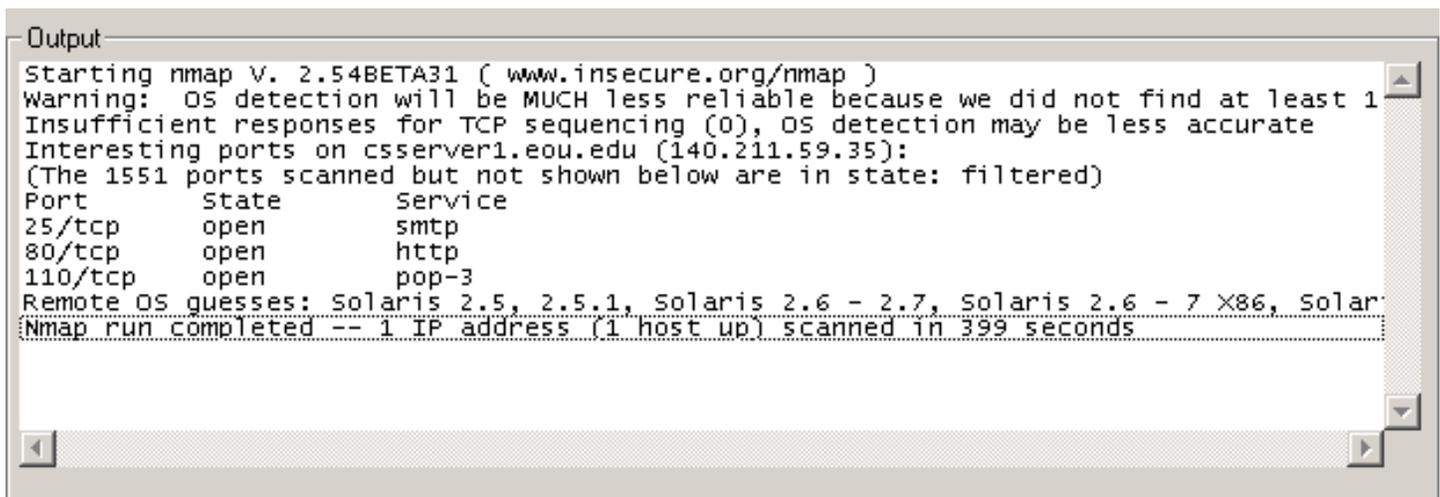
Exploits

An exploit is an attempt to gain access to a network and perform some sort of malicious activity. To do this, the attacker must first determine if a network component or server on a network has a weakness that can be taken advantage of; in the case of a web server two key pieces of information are needed initially:

- the type of Operating System running on the server
- the application software supporting web operations on the server.

Overview

With these two bits of information, an attacker could do some sort of damage based on known vulnerabilities in a system. Below is an OS scan of the CSMM web server running on subnet 59 of the EOU network from a remote location:



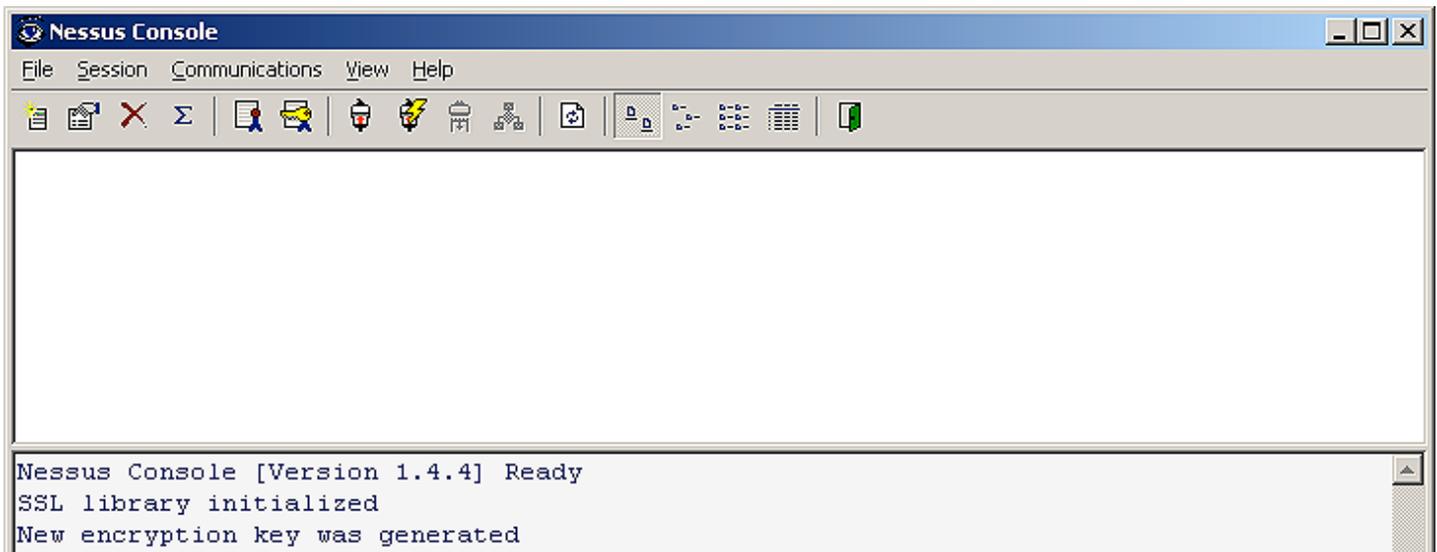
```
Output
Starting nmap V. 2.54BETA31 ( www.insecure.org/nmap )
Warning: OS detection will be MUCH less reliable because we did not find at least 1
Insufficient responses for TCP sequencing (0), OS detection may be less accurate
Interesting ports on csserver1.eou.edu (140.211.59.35):
(The 1551 ports scanned but not shown below are in state: filtered)
Port      State      Service
25/tcp    open      smtp
80/tcp    open      http
110/tcp   open      pop-3
Remote OS guesses: Solaris 2.5, 2.5.1, Solaris 2.6 - 2.7, Solaris 2.6 - 7 X86, Solar
Nmap run completed -- 1 IP address (1 host up) scanned in 399 seconds
```

Nmap port scan

Security is sufficient on this server that Nmap was not able to determine the exact operating system running on the server, or the web server software installed. Since the OS is not known an attacker would have a much harder time attempting to break into the system. The attacker could, instead, attempt to mount an attack based on the web server application running on this machine.

There are a variety of tools available for detecting weaknesses in applications running on network servers. Network administrators routinely use such tools to help determine possible holes in their systems; an attacker can use the same tools to do precisely the same for malicious purposes. One tool is Nessus available for download; this tool is comprised of client monitoring component and a server-side daemon. Both components must be installed for the software to be utilized.

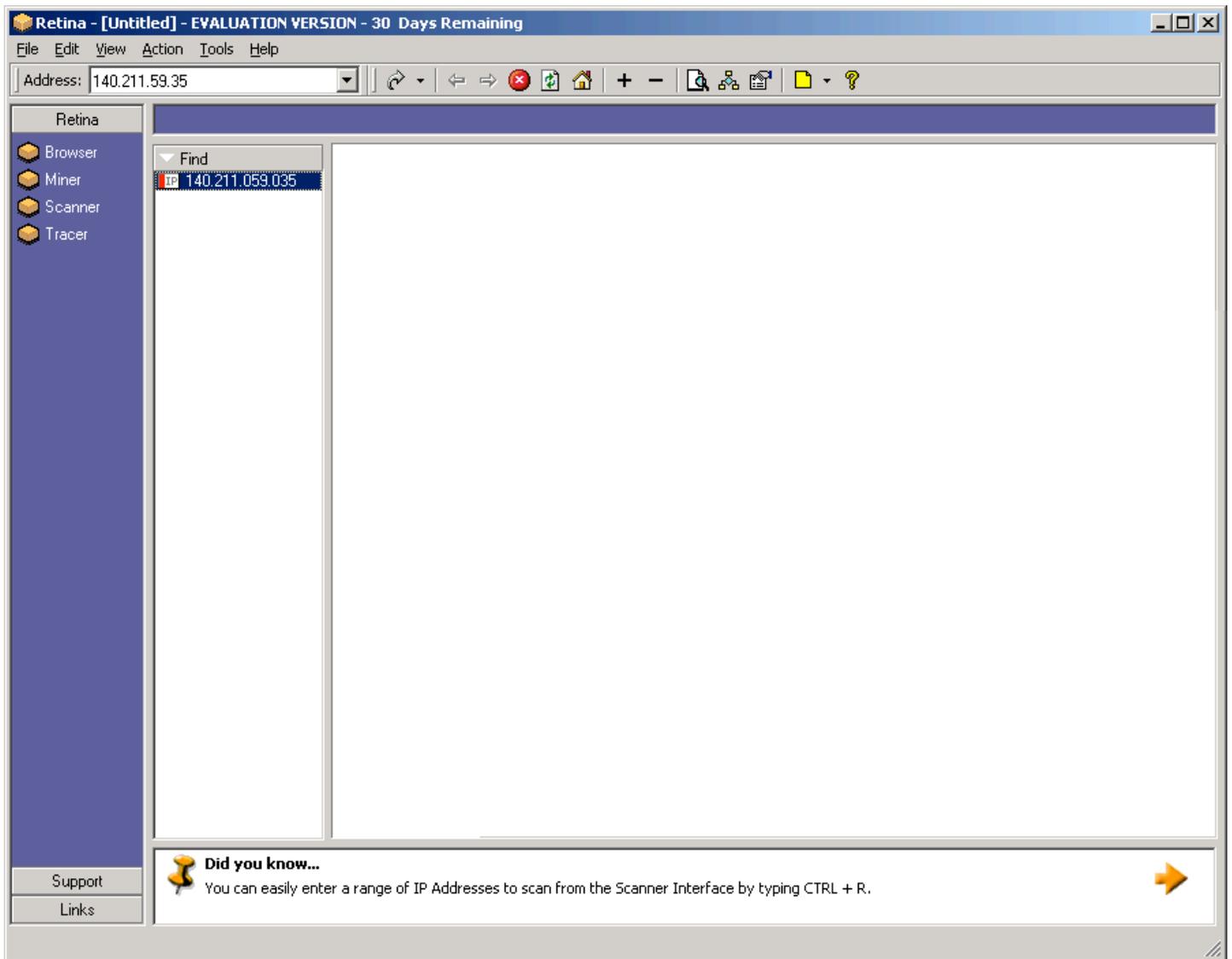
The client-side component interface is comprised of various commands and an output area, shown below:



Nessus client interface (Windows)

This interface allows the user to establish a connection with a server running the server daemon, monitor security weaknesses on the server, generate network certificates, and generate reports with statistics and other data concerning server security. The Nessus organization maintains a database of known server application vulnerabilities on their web site administrators can access for informational purposes; attackers have access to the same information.

If an attacker knew the OS and web server application of a server on a network, a variety of tools exist for gaining additional information about security issues of the server. The Apache web server application is a widely used, open source application with documented vulnerabilities that can be exploited by attackers for malicious purposes. There are various network monitoring tools designed to check for vulnerabilities in the Apache application used by network administrators; one such tool is the Retina Apache Chunked Scanner.



Retina Apache Chunked Scanner

In Retina, the user inputs an IP address in the input field to check for a variety of information on a server. In this example, scanning of the CSMM web server provides a potential attacker with a variety of information about the server, shown below:

▼	General	140.211.059.035
	Address	140.211.59.35
	Report Date	01/11/04 12:03:04 PM
	Domain Name	cssserver1.eou.edu
	Ping Response	Host Did Not Respond
	Traceroute	192.168.123.254,4.5.112.1,4.9.0.130,63.211.200.246,207.98.64.140,207.98.64.46,207.98.64.62,
▼	Audits	140.211.059.035
	🔺 Web Servers	TCP:80 - Apache 2.0.44 LineFeed DoS
	🟡 Web Servers	TCP:80 - Apache 2.0 Cipher Downgrade
	🟡 Web Servers	TCP:80 - Apache 2.0.45 DS2 Filestat DoS
	🟡 Web Servers	TCP:80 - Apache 2.0.46 APR_PSPrintf Memory Corruption
	🟡 Web Servers	TCP:80 - Apache HTTP Server FTP proxy server DoS
	🟡 Web Servers	TCP:80 - Apache HTTP Server prefork MPM denial of service
	🟡 Web Servers	TCP:80 - Apache mod_alias and mod_rewrite Buffer Overflow
	🟢 Web Servers	TCP:80 - HTTP TRACE method supported
▼	Machine	140.211.059.035
	OS Detected	Not Enough Data (No Closed Ports)
	Filtered Ports	1911
	Open Ports	1
▼	Ports	140.211.059.035
	80	WWW-HTTP - World Wide Web HTTP (Hyper Text Transfer Protocol)
▼	Services	140.211.059.035
▼	Shares	140.211.059.035

Retina web server scan

The user now knows that port 80 (supporting HTTP) is open on this machine and is running Apache v2.0.44; known vulnerabilities in this version of Apache can be exploited. However, the OS of this machine is not known and provides for security regarding attacks on that level of the server.

A Retina scan of the main EOU server (snake) provides the following information:

▼	General	140.211.060.020
	Address	140.211.60.20
	Report Date	01/11/04 12:29:18 PM
	Domain Name	snake.eou.edu
	Ping Response	Host Did Not Respond
	Traceroute	192.168.123.254,4.5.112.1,4.9.0.161,209.247.9.41,209.247.9.54,63.211.200.246,207.98.64.140,207.98.64.46,207.98.64.62
▼	Audits	140.211.060.020
▲	Mail Servers	TCP:25 - Sendmail 8.12.9 Buffer Overflow
▲	Mail Servers	TCP:25 - Sendmail address field parsing buffer overflow
▲	Mail Servers	TCP:25 - Sendmail DNS Map TXT Overflow
▲	Mail Servers	TCP:25 - Sendmail prescan() address buffer overflow
▲	Web Servers	TCP:443 - Apache chunking integer overflow vulnerability
▲	Web Servers	TCP:80 - Apache chunking integer overflow vulnerability
▲	Web Servers	TCP:80 - ApacheBench multiple buffer overflows
▲	Web Servers	TCP:443 - ApacheBench multiple buffer overflows
■	Web Servers	TCP:80 - Apache 1.3.12 Mass virtual hosting CGI source disclosure
■	Web Servers	TCP:443 - Apache 1.3.12 Mass virtual hosting CGI source disclosure
■	Web Servers	TCP:80 - Apache 1.3.20 Multiview directory listing
■	Web Servers	TCP:443 - Apache 1.3.20 Multiview directory listing
■	Web Servers	TCP:80 - Apache 1.3.27 0x1A Character Logging DoS
■	Web Servers	TCP:443 - Apache 1.3.27 0x1A Character Logging DoS
■	Web Servers	TCP:443 - Apache 1.3.27 HTDigest Command Execution
■	Web Servers	TCP:80 - Apache 1.3.27 HTDigest Command Execution
■	Web Servers	TCP:443 - Apache httpd scoreboard modification vulnerability
■	Web Servers	TCP:80 - Apache httpd scoreboard modification vulnerability
■	Web Servers	TCP:80 - Apache mod_alias and mod_rewrite Buffer Overflow
■	Web Servers	TCP:443 - Apache mod_alias and mod_rewrite Buffer Overflow
◆	CGI Scripts	TCP:80 - Debian Linux httpd Vulnerability
◆	Mail Servers	TCP:25 - EXPN Command Enabled
◆	Mail Servers	TCP:25 - VRFY Command Enabled
◆	Web Servers	TCP:80 - Apache 1.3.20 Root Directory Access Vulnerability
◆	Web Servers	TCP:443 - Apache 1.3.20 Root Directory Access Vulnerability
◆	Web Servers	TCP:443 - OpenSSL CBC encryption timing attack vulnerability
◆	Web Servers	TCP:80 - OpenSSL CBC encryption timing attack vulnerability
☒	Web Servers	TCP:80 - HTTP TRACE method supported
▼	Machine	140.211.060.020
	Last Boot:	14 days, 8 hours, 26 minutes, 0 seconds
	OS Detected	Apple Mac OS 7.5.5 - 9
	Closed Ports	1
	Filtered Ports	1907
	Open Ports	4
▼	Ports	140.211.060.020
	25	SMTP - Simple Mail Transfer Protocol
	80	WWW-HTTP - World Wide Web HTTP (Hyper Text Transfer Protocol)
	110	POP3 - Post Office Protocol - Version 3
	443	HTTPS - HTTPS (Hyper Text Transfer Protocol Secure) - SSL (Secure Socket Layer)
	554	RTSP - Real Time Stream Control Protocol - CLOSED
▼	Services	140.211.060.020
▼	Shares	140.211.060.020
▼	Users	140.211.060.020

Retina web server scan

An attacker has a large amount of data to work with from this scan, with a variety of port numbers and protocols supported by the machine. Additionally, the OS, web and mail server software is known allowing the attacker to formulate an attack based on the known vulnerabilities of these systems (this is not to say that additional security precautions have been established to deal with

possible attacks). Some known vulnerabilities are stated in the output; if used on a regular basis a Network Administrator would be able to monitor and secure these vulnerabilities with a variety of built-in security mechanisms and software updates or patches.

Comments and Conclusions

Retina (and other monitoring software) can be run in the background of a workstation on a network, and provide the administrator with real-time monitoring on a variety of machines. One of the useful features in Retina is the ability to scan a range of IP addresses concurrently, providing the administrator with information on a range of servers in selected portions or subnets of a larger network. If implemented and monitored on a regular basis, such a tool could provide a monitoring scheme of the activity on a network and red-flag suspicious traffic or activity as a early warning system.

Rootkits

When an intruder has gained unauthorized access to a network, to continue working on a repeated basis they must leave no evidence of ever being on the network. Use of a Rootkit helps to accomplish two things:

- cover the tracks of the intruder's access, and
- help gather information about network servers and their users.

Overview

Rootkits can erase the contents of server monitor logs and monitor keystrokes of systems users to sniff for passwords and other confidential information⁵. A rootkits may be comprised of several programs that allow the intruder to gain access and hide their activity.

Tuxkit is an example of a rootkit that contains numerous features; it is (apparently) easy to install for the user, covers intrusion activity, replaces server system files with binaries used to gather information from the server, and leaves a background on the network for future access. The only problem with using Tuxkit appears to be its availability; formats of mentioned downloads are for Unix-based systems and were not readily available at the time of this writing.

Tuxkit installs several different packages on a system; below is a listing of packages installed on a Linux Redhat 7 system utilized on a security honeypot:

```

--] Packages

[root@angel tuxkit-1.0]# ls -l ../tuxkit (tuxkit.tgz)
total 2600
-rw----- 1 sfork sfork 502884 Dec 5 07:55 bin.tgz
-rw----- 1 sfork sfork 406 Dec 5 07:55 cfg.tgz
-rw----- 1 sfork sfork 16213 Dec 5 07:55 lib.tgz
-rw----- 1 sfork sfork 3684 Dec 5 07:55 README
-rw----- 1 sfork sfork 461892 Jan 6 00:06 sshd.tgz
-rw----- 1 sfork sfork 1644819 Dec 5 07:55 tools.tgz
-rwx----- 1 sfork sfork 9489 Jan 6 00:53 tuxkit

[root@angel tuxkit-1.0]# ls -l ../tuxkit-1.0 (tuxkit-1.0.tgz)
total 2600
-rw----- 1 sfork sfork 502884 Dec 5 07:55 bin.tgz
-rw----- 1 sfork sfork 406 Dec 5 07:55 cfg.tgz
-rw----- 1 sfork sfork 16213 Dec 5 07:55 lib.tgz
-rw----- 1 sfork sfork 3684 Dec 5 07:55 README
-rw----- 1 sfork sfork 461892 Jan 6 00:06 sshd.tgz
-rw----- 1 sfork sfork 1644819 Dec 5 07:55 tools.tgz
-rwx----- 1 sfork sfork 9489 Jan 6 00:53 tuxkit

[root@angel tuxkit-1.0]# ls -l ../tuxkit-short (tuxkit-1.0-short.tgz)
total 1556
-rw----- 1 1001 1001 502884 Dec 5 07:55 bin.tgz
-rw----- 1 1001 1001 406 Dec 5 07:55 cfg.tgz
-rw----- 1 1001 1001 16213 Dec 5 07:55 lib.tgz
-rw----- 1 1001 1001 3684 Dec 5 07:55 README
-rw----- 1 1001 1001 461892 Jan 6 00:06 sshd.tgz
-rw----- 1 1001 1001 577089 Jan 6 01:12 tools.tgz
-rwx----- 1 1001 1001 9489 Jan 6 00:53 tuxkit

--] tuxkit-1.0.tgz

```

Tuxkit package contents

Note: A utility called Chkrootkit used for detecting rootkit activity did not detect the presence of these packages on the honeypot servers.

There are six files in the Tuxkit utility that include a ReadMe file and an installation script. The remaining utilities are descriptively named. To install Tuxkit, a single install script is run to create various tools on the server. Below are the contents of the individual files:

- bin.tgz - contains precompiled trojan binaries
- cfg.tgz - contains tuxkit's configuration files
- lib.tgz - contains libproc libraries, for process hiding purposes
- sshd.tgz - contains precompiled sshd, complete with sshd_config
- tools.tgz - contains an arsenal of tools (duh!) for the skrip kiddie who don't know how to get their own tools. The tools are:

```
[root@angel tools]# ls -la
total 44
drwxr-xr-x  11 root  root   4096 Mar  1 13:14 .
drwxr-xr-x   4 root  root   4096 Mar  1 13:14 ..
drwx-----   2 root  root   4096 Nov 12 20:50 bitchx
drwx-----   2 root  root   4096 Dec 12 23:59 dos
drwx-----   2 root  root   4096 Nov 12 20:57 mirkforce
drwx-----   2 root  root   4096 Nov 12 20:57 nmapv
drwx-----   8 root  root   4096 Nov 12 23:05 psybnc
drwx-----   2 root  root   4096 Nov 13 01:00 sniffer
drwx-----   2 root  root   4096 Nov 12 20:58 ssh
drwx-----   2 root  root   4096 Nov 12 23:22 synscan
drwx-----   2 root  root   4096 Nov 12 20:58 utils
```

Tuxkit file contents

All these files are precompiled, making it easy for non-programmers to install and use the tools. The 'utils' file contains one tool called 'wget', that enables script kiddies to easily download other intrusion tools. Wget is readily available for download at various security sites.

Once installed, Tuxkit covers intruder activity and gathers information; it also send an email to the intruder the the IP address and backdoor information on the server being compromised:

```
ssh 192.168.0.40 -l root -p 4000 # test.allan.org password: R0s3ann3
psyBNC: 4001
```

Tuxkit email ACK

The contents of Tuxkit allow it to function in an automated fashion; using a security scanner the intruder looks for vulnerable servers and installed the rootkit on a server when discovered. Information is gained when installed and the intruder can then attempt to take control of other servers on the network.

There are other rootkit utilities available for a variety of platforms. tOrn rootkit is a Linux-based utility design to be quickly installed and implemented by an intruder; all binary code is precompiled and easily executed¹. The utility consists of three primary tools:

- a log cleaner (t0rnsb),
- a network sniffer (t0rns), and
- a log parser (t0rnp).

Below is a listing of the tOrn binary files before installation:

¹ Toby Miller, discussion @ <http://www.sans.org/y2k/t0rn.htm>

File	Size	Timestamp
Du	22460	August 22 2000
Find	57452	August 22 2000
Ifconfig	32728	August 22 2000
In.fingerd	6408	August 22 2000
Login	3964	August 22 2000
Ls	39484	August 22 2000
Netstat	53364	August 22 2000
Pg	4568	September 13 2000
Ps	31336	August 22 2000
Pstree	13184	August 22 2000
Sz	1382	July 25 2000
TOrn	7877	September 13 2000
Top	266140	July 17 2000

When installed, tOrn creates its own directory on the server, and installs the required files into this directory. This makes detecting the rootkit fairly easy; sizes of key files in the /bin directory on the server may change, and file timestamps (modification dates) may also be updated. An example is a test done on a Linux 6.1 system with a /bin/ps file listing using the ls-la command:

```
-r-xr-xr-x  1 root  root  61244 Sept 26 1999
If tOrn is installed the user would see the following:
-r-xr-xr-x  1 root  root  31336  Sept 26 1999
```

The top line shows normal output for a the /bin/ps permissions, file size and timestamp. The bottom indicates a difference in file size and an offset of one byte from the original, although there is no modification of the timestamp.

tOrn can be detected by using lsof on a Linux system. The rootkit uses port 47107 to monitor the network and will be evident when running a lsof | grep LISTEN command. Running nMap and scanning for port 45-48k also will shown evidence of tOrn on a system.

Comments and Conclusions

Exploration on various security discussion boards indicated that Tuxkit sniffs SSH passwords; updates to newest versions of SSH are suggested to battle the use of this intruder utility, openSSH v3.6.x being suggested most often.

Another recommended solution is to not used SSH passwords on a system, but rather using public/private server key authentication instead. To do this, run ssh-keygen to generate a key, and publish the key to make it available:

```
"ssh-keygen -t rsa" generates a rsa key pair, writes the public key in
".ssh/id_rsa.pub" and the private one in ".ssh/id_rsa";

adding the content of ".ssh/id_rsa.pub" to ".ssh/authorized_keys" on the
remote host will let you log in without your password travelling the net.
```

² Security discussion @ <http://lists.indymedia.org/pipermail/imc-sysadmin/2003-July/002323.html>

Installation of a rootkit detector is recommended. There are several available as free downloads, including the following:

- Chkrootkit
- Rootkit Hunter

Chkrootkit has been tested on Linux 2.0-2.4 systems, FreeBSD 2.2 – 5.x, OpenBSD 2.x and 3.x, NetBSD, Solaris 2.5, 2.6 and 8.0, HP-UX11, Tru64 and BSDI³. It is a shell script that checks system binary files for the presence of rootkit software on network servers; the following tests are advertised as currently available in the utility:

```
◊ aliens asp bindshell lkm raxedcs sniffer wted w55808 scalper
  slapper z2 amd basename biff chfn chsh cron date du dirname echo
  egrep env find fingerd gpm grep hdparm su ifconfig inetd inetdconf
  init identd killall ldsopreload login ls lsof mail mingetty
  netstat named passwd pidof pop2 pop3 ps pstree rpcinfo rlogind
  rshd slogin sendmail sshd syslogd tar tcpd tcpdump top telnetd
  timed traceroute vdir w write
```

- ♦ **ifpromisc.c**: checks if the interface is in promiscuous mode.
- ♦ **chklastlog.c**: checks for lastlog deletions.
- ♦ **chkwtmp.c**: checks for wtmp deletions.
- ♦ **check_wtmpx.c**: checks for wtmpx deletions. (Solaris only)
- ♦ **chkproc.c**: checks for signs of LKM trojans.
- ♦ **chkdirs.c**: checks for signs of LKM trojans.
- ♦ **strings.c**: quick and dirty strings replacement.

This utility is readily available for download from the manufacturer.

Tripwire is a system tool that checks for variance in file integrity⁴. It can determine if a file has been altered in any way, and detects if files have been added or deleted from a network system. It can be used as a daily monitoring tool, and will notify administrators if system files have altered or tampered with. This monitoring utility is open source code and readily available for Linux, Sun Solaris, HP-UX, IBM AIX and Windows NT systems. The AIDE tool has been designated as a free Tripwire replacement for Unix systems.

Numerous other network scanning and monitoring software packages are available for various platforms. A very comprehensive listing of tools is available at: <http://www.insecure.org/tools.html>

³ info available on software mfg page @ <http://www.chkrootkit.org/>

⁴ author web site @ <http://www.tripwire.org/downloads/index.php>

Spoofting and Denial of Service

Spoofting and Denial of Service attacks are two of the main ways an intruder can affect the functionality of a network. Both these methods have gained popularity in recent years, with famous cases of Denial of Service being conducted on Yahoo.com and other large networks in the last decade. The implications of these attacks economically can be substantial; technically they challenge a network administrator to create monitoring schemes to recognize such attacks before they happen, and implement some solution when they inevitably occur on a network.

Spoofting

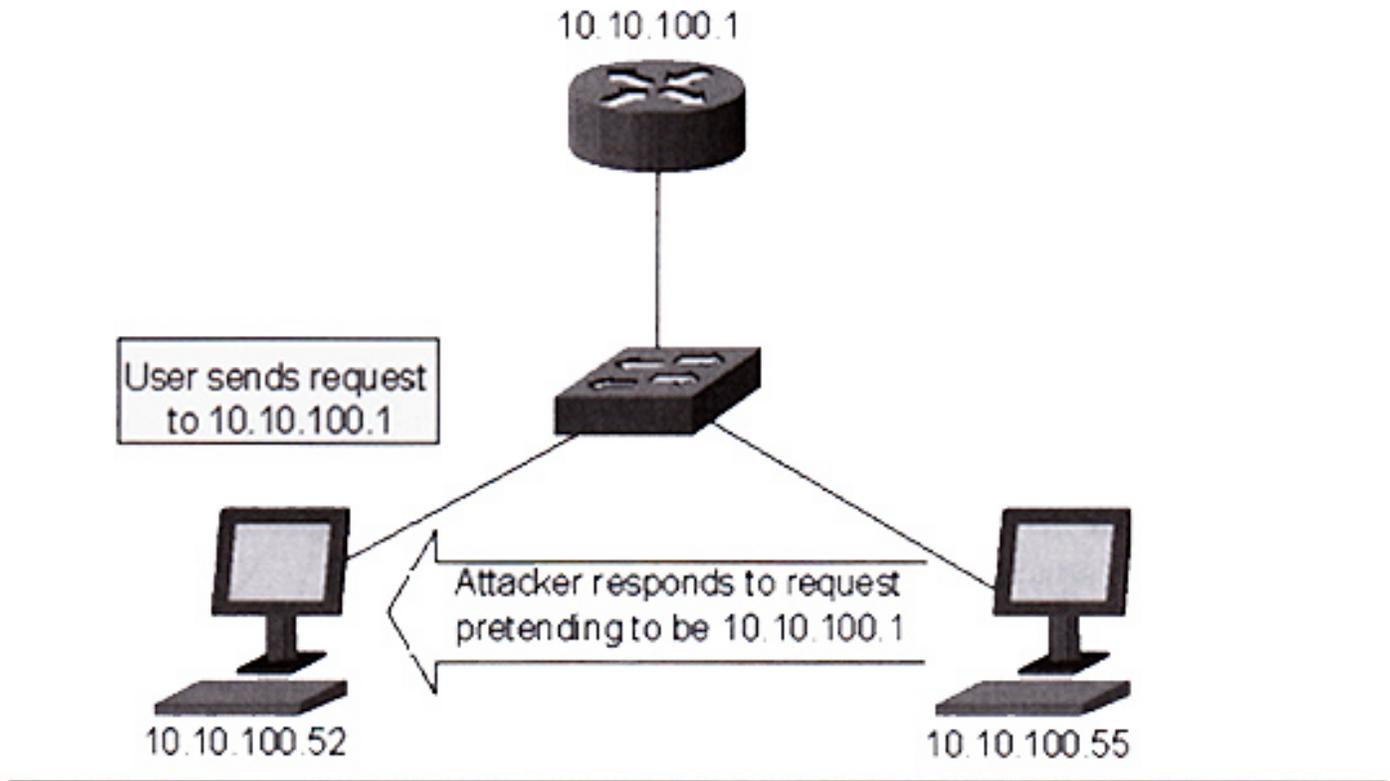
A Spoofting attack is one in which the source IP address of a data packet is forged. There are two general types of spoofting attacks:

- IP spoofting used in DoS attacks, and
- spoofting for Man-in-the-Middle attacks.

IP spoofting is an attempt for an intruder to forge the IP address of packets sent to a network so as to disguise their source; oftentimes this address is contained in the RFC 1918 address space. Normally, the targeted host sends a response (ACK) to the request which never gets returned; if multiple spoofting requests are sent they stack up in the host's memory buffer. If enough unanswerable requests are sent to the host, the memory buffer will eventually fill up, the device will become unstable and cease to function or crash. When this happens, the network is unable to receive or respond to other legitimate requests, essentially rendering it inoperable. This type of attack can be performed on an intermediary network device as a switch or router, on a specific server or series of network devices on a network.

Man-in-the-middle attacks can be even more devastating to a network's functions. This type of attack involves the intruder monitoring and/or altering data on a network; this requires a passive or undetected access to data traveling across network devices. The intruder attempts to intercept data before it reaches its destination device on the network, and sends a response from a compromised host; if the original sender responds to this spoofted host address then the intruder can then collect and monitor data from the original source.

⁵ Liska, The Practice of Network Security; 2003



Network Spoofing scheme

An intelligent man-in-the-middle attack will intercept data, monitor & collect it, even modify it and then send it to its original destination so as to not draw attention to the process. If access to the right host is gained and enough data collected, an intruder could create a list of usernames and passwords on a network essentially undetected. This would then allow uninhibited access to potentially sensitive data on a server.

There are precompiled tools that allow for simple man-in-the-middle attacks. One of the most readily available and popular is Ettercap, available for Windows, Sun Solaris, BSD, Mac OSX and Linux systems. According to the manufacturer,

“Ettercap is a multipurpose sniffer/interceptor/logger for switched LAN. It supports active and passive dissection of many protocols (even ciphered ones) and includes many feature for network and host analysis.”

With Ettercap you can sniff a network in four modes:

- **IP** based: filter packets according to IP destination and source
- **MAC** based: filter packet MAC addresses, useful for sniffing connections on network gateways
- **ARP** based: uses ARP ‘poisoning’ to sniff in a switch LAN between two hosts in full-duplex mode, and
- **publicARP** based: as above from a compromised host to all other hosts on a network in half-

⁶ <http://ettercap.sourceforge.net/>

duplex mode.

When launched Ettercap gives a list of IP addresses of hosts on the local area network the application is being running from; a user can then choose one of the above methods of sniffing:

```
----- ettercap 0.0.7 -----
----- 48 hosts in this LAN (192.168.0.30 : 255.255.255.0) -----
 1> 192.168.0.76      1) 192.168.0.76
 2> 192.168.0.22     2) 192.168.0.22
 3> 192.168.0.205    3) 192.168.0.205
 4> 192.168.0.123    4) 192.168.0.123
 5> 192.168.0.89     5) 192.168.0.89
 6> 192.168.0.235    6) 192.168.0.235
 7> 192.168.0.194    7) 192.168.0.194
 8> 192.168.0.98     8) 192.168.0.98
 9> 192.168.0.199    9) 192.168.0.199
10> 192.168.0.183   10) 192.168.0.183
11> 192.168.0.98    11) 192.168.0.98
12> 192.168.0.191   12) 192.168.0.191
13> 192.168.0.135   13) 192.168.0.135
14> 192.168.0.214   14) 192.168.0.214
15> 192.168.0.191   15) 192.168.0.191
16> 192.168.0.232   16) 192.168.0.232
17> 192.168.0.46    17) 192.168.0.46
18> 192.168.0.18    18) 192.168.0.18
19> 192.168.0.128   19) 192.168.0.128
20> 192.168.0.190   20) 192.168.0.190
21> 192.168.0.103   21) 192.168.0.103
22> 192.168.0.68    22) 192.168.0.68
23> 192.168.0.19    23) 192.168.0.19
24> 192.168.0.222   24) 192.168.0.222
25> 192.168.0.210   25) 192.168.0.210
26> 192.168.0.63    26) 192.168.0.63
27> 192.168.0.21    27) 192.168.0.21
-----
Your IP: 192.168.0.30 with MAC: 00:00:24:4C:00:F9 on Iface: eth0
Host: Unknown host (192.168.0.76) : 19:00:00:00:4C:26
```

Etterccap IP address capture

When choosing a method of sniffing, a single IP host address can be chosen and traffic noted on that host. The interface changes to show source and destination ports of traffic; actual packet data can be observed flowing on these ports:

⁷ <http://www.securemac.com/macosexettercap.php>

```

ettercap 0.0.7
SOURCE: 192.168.0.76 <
DEST : 192.168.0.22 <
      doppleganger - illithid - ettercap

48 hosts in this LAN (192.168.0.30 : 255.255.255.0)

192.168.0.76:65427 active
190..N..a..G..A..200..N..a..
.G..A..210..N..a..G..A..220..
.N..a..G..A..

192.168.0.22:17
182..A..L..o..R..183..A..L..
.o..R..184..A..L..o..R..185..
.A..L..o..R..186..A..L..o..
R..188..A..L..o..R..189..A..
.L..o..R..191..A..L..o..R..
192..A..L..o..R..193..A..L..
.o..R..194..A..L..o..R..195..
.A..L..o..R..196..A..L..o..
R..197..A..L..o..R..198..A..
.L..o..R..199..A..L..o..R..
201..A..L..o..R..202..A..L..
.o..R..203..A..L..o..R..20
5..A..L..o..R..206..A..L..o..
R..207..A..L..o..R..208..
A..L..o..R..209..A..L..o..R
211..A..L..o..R..212..A..
L..o..R..213..A..L..o..R..2
14..A..L..o..R..215..A..L..
o..R..216..A..L..o..R..217..
A..L..o..R..218..A..L..o..
R..219..A..L..o..R..

Your IP: 192.168.0.30 with MAC: 00:00:24:4C:00:F9 on Iface: eth0

```

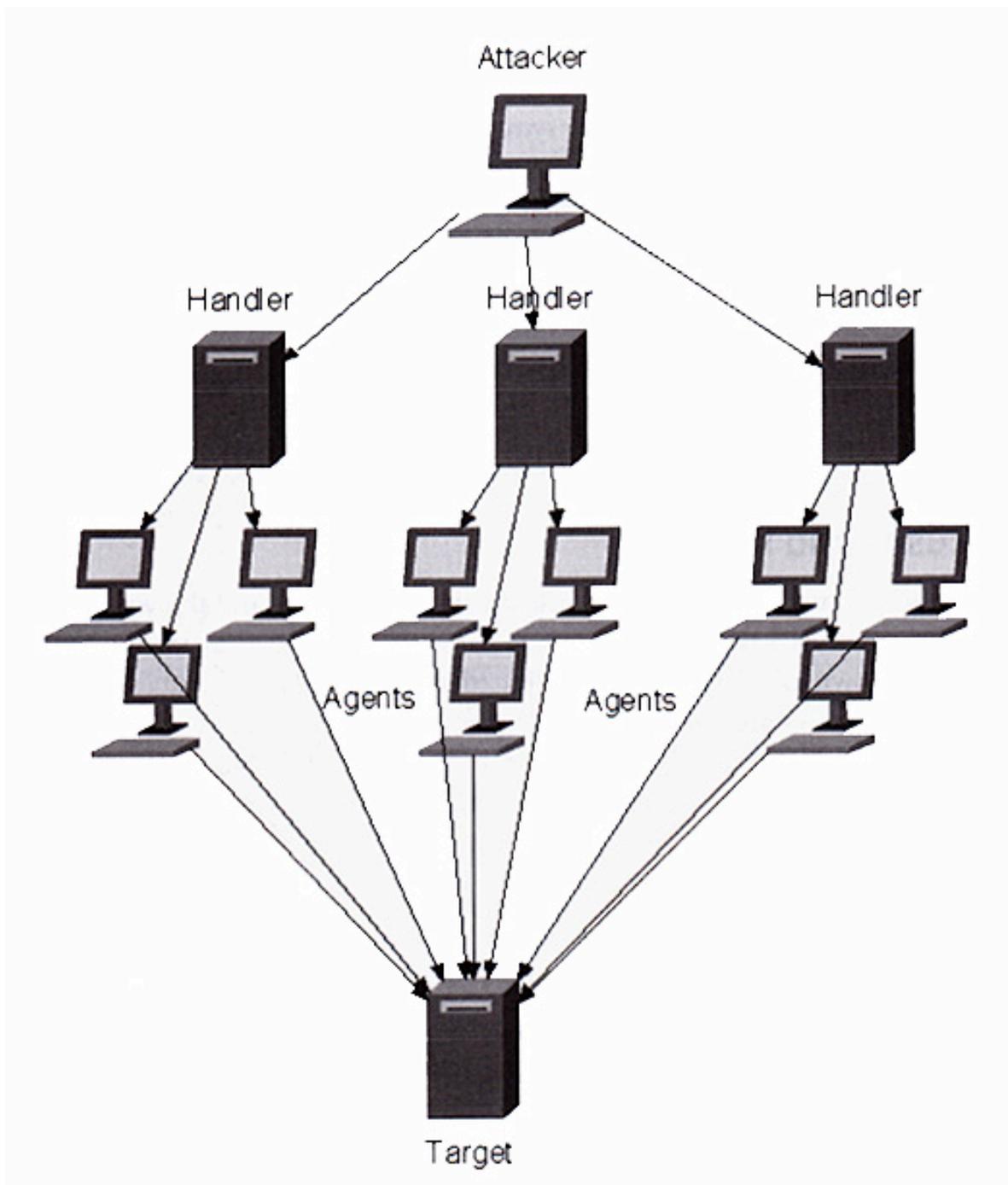
Ettercap Packet capture

Packet data can be viewed, even modified, in these packet streams allowing an intruder to monitor and alter data as it flows on the network. There is a built-in Help Guide explaining functions and commands for the interface. This software utility is freely available for multiple platforms at the author's website.

Denial of Service (DoS)

DoS is a method of combining together intruder technologies in an attempt to disable a network. The process can involve a number of methods, and can be effective in exploiting specific weaknesses in a network security scheme. Large scale Distributed Denial of Service (DDoS) attacks came to light in February 2000 when implemented on eBay and Amazon web sites, costing these companies substantial network downtime and lost revenues.

A DDoS attack consists of a client machine gaining access to one or more intermediary machines on a network to launch an attack on multiple hosts on a network. The attacker starts by scanning for a specific network with security weaknesses that will allow access to a network device, preferably a server. Once access is gained to said host, a sniffer program is installed to allow scanning of larger parts of the network. The intruder attempts to find multiple machines to use as 'handlers' to gain control of the network as a whole; further exploration of various parts of the network through these handlers can result in a chain of computers that are under the control of the intruder through the single compromised host.



DDos Attack scheme

When access is gained to a variety of remote machines, a rootkit program is installed allowing a DDoS to be automated throughout the network using these machines as 'agents'. This allows for the intruder identity and location to be hidden, and for an attack to be generated from the handler machines. There is generally one targeted host machine (as a web server) that is trying to be reached through this subnetwork of remote machines.

Through this system the intruder attempts to flood the target with large amounts of TCP requests, originating at the handlers and generated by the agents. A large number of requests would use up network resources and leave the target in an unstable state, making it inoperable for legitimate traffic.

There are several popular tools for such attacks including Trin00 (also known as Trinoo), Tribla Glod Network, and Stacheldraht.

Trin00, and its Windows cousin Wintrin00, uses UDP packets to flood a network. These packets are sent from a single source but flood various ports on the targeted host. The hosts then respond with unreachable ICMP port messages until all resources on the host are used up. Following is a description of an attack on a University of Michigan network in 1999⁸:

“Trinoo (also known as trin00) was the first well known DDoS attack used against the University of Minnesota in August 1999. This two day attack involved flooding servers with UDP packets originating from thousands of machines. Source addresses were not spoofed, so systems running the offending daemons were contacted. However, the attacker responded simply by introducing new daemon machines into the attack. Trinoo was first found as a binary daemon on a number of compromised Solaris 2.x systems. Malicious code had been introduced through exploitation of buffer over-run bugs in the remote procedure call (RPC) services ‘statd’, ‘cmsd’ and ‘ttdbserverd’. (See CERT IN-99-04 for a description of these exploits).”

Comments and Conclusions

Tools have been developed for detecting and resolving DDoS attacks with the above programs. PestPatrol⁹ for Windows automatically detects the presence of Trin00 on a network, and provides solutions for removing the tool from the system by running processes in the Task Manager. The Nessus tool will also allow for scanning of the Trin00 bug on a system by using the UDP Port scanning option; there is a plugin module available from Nessus.org for this purpose¹⁰. The ISS SafeSuite of intrusion detection utilities contains the RealSecure tool designed to detect DDoS tools on a network for early warning and monitoring. RealSecure allows for firewall and router configurations to block traffic on a particular port or from a particular service¹¹. RealSecure is available for download in various applications including network Management and Protection of network servers and desktop workstations at <http://www.iss.net/download/>.

⁸ <http://www.sans.org/resources/idfaq/trinoo.php>

⁹ <http://www.pestpatrol.com/PestInfo/w/win-trin00.asp>

¹⁰ <http://cgi.nessus.org/plugins/dump.php3?id=10288>

¹¹ <http://xforce.iss.net/xforce/alerts/id/advice43>

Computer Viruses

Computer virus detection has become an important aspect of network security with the proliferation of more and sophisticated types of viruses and worms. In just the past year several new strains of computer viruses have flourished in North America, including releases of W32.Nimda.E@mm, W32.Sobig.B@mm, and most recently the W32.Novarg.A@mm mass-mailing worm (among numerous others). The most popular method of distributing viruses has become through desktop workstation email clients, using a client's address book to proliferate the distribution of the worm when the virus is launched by opening the executable in the form of an attachment. This allows the worm to spread internally through the client's machine and externally to other recipients of the email attachment.

Workstation Security

Desktop virus security consists of installation of virus detection systems on client workstations; these security systems use virus definitions files to detect and eradicate known malicious code on a workstation. One such system is the Norton Anti-Virus software available for various platforms. Upon installation of the software, a user can configure the utility to for specific detection functions:

Norton SystemWorks

Home LiveUpdate Options Help

System Status: Attention ⚠

Security Scanning Features

✓ Auto-Protect	On
✓ Email Scanning	On
✓ Script Blocking	On
⚠ Full System Scan	8/14/2003

Virus Definition Service

✓ Virus Definitions	1/29/2004
✓ Subscription Service	9/22/2004
✓ Automatic LiveUpdate	On

Item Details

The items marked in red need your attention.

Please select an item by clicking on the item at left in order to get more information and take the necessary action.

Norton SystemWorks 2002

Virus definition files are periodically installed via the LiveUpdate function, allowing for up-to-date protection on desktop machines throughout a network. Unfortunately, this kind of protection relies upon two major circumstances to be effective: **1)** the updates must be installed by the client on a regular basis, and **2)** virus definitions only become available when a virus is detected and analyzed by the software manufacturer, resulting in a natural lag in deployment.

One of the deadliest of virus strains comes from new 'Blended Threat' viruses, worms that combine a variety of attack methods together in a single threat. Examples of such threats include the Sadmin, CodeRed, Nimda and Lion worms¹². Characteristics of such blended threats include:

- harmful in nature
- exploits network vulnerabilities
- incorporates multiple attack methods
- requires no user interaction to launch
- uses multiple methods to propagate.

Nimda represents a recent blended threat that was detected in September, 2001. It is a complex virus with a mass mailing worm component which spreads itself in attachments named README.EXE. It affects Windows 95, Windows 98, Windows Me, Windows NT 4 and Windows 2000 users. Nimda is the first worm to modify existing web sites to start offering infected files for download. Also it is the first worm to use normal end user machines to scan for vulnerable web sites. This technique enables Nimda to easily reach intranet web sites located behind firewalls - something worms such as Code Red couldn't directly do¹³. The worldwide economic impact of the Nimda threat was estimated at more than \$590 million. An in-depth description of the Nimda threat is available at <http://www.cert.org/advisories/CA-2001-26.html>.

Software manufacturers as Symantec and McAfee have issued virus definitions for their antiVirus products to deal with the Nimda threat. Additionally, current updates to virus definitions virus removal tools are available from Symantec for end users of their antivirus products.

Network Security

For network administrators, the most effective defense against viruses is to not allow them to infiltrate the network and propagate. Real-time detection is a fundamental requirement of this security scheme, accomplished in part by ingress filtering of traffic as it enters the network. Specifically for Nimda, port and protocol filtering on the border of a network (80/tcp) can help eliminate infiltration and infection of network servers not explicitly authorized for public access. Filtering on 69/udp can help prevent Nimda from infecting a network server via ftp¹⁴. Cisco published a tech tip regarding the Nimda worm at <http://www.cisco.com/warp/public/63/nimda.shtml>.

Microsoft has created several server OS patches to deal with Nimda, offered in Security Bulletins MS00-078 and MS01-044. Further patches were made available for vulnerabilities in the Internet Explorer browser, offered in MS01-020 and MS01-027.

¹² The Journal: Technology Horizons in Education, December 2002

¹³ <http://www.f-secure.com/v-descs/nimda.shtml>

¹⁴ <http://www.cert.org/advisories/CA-2001-26.html>

Egress filter, or managing network traffic as it exits the network, can also help eradicate threats. There is normally little need for egress filtering; in the case of Nimda, employing egress filtering on port 69/udp at your network border will prevent certain aspects of the worms propagation both to and from your network.

Comments and Conclusions

Clearly, virus propagation is a evolving and serious threat to network security; a network-wide scheme for virus protection is needed for detection and eradication of malicious code, while guaranteeing normal network functionals and access.

There is much information available online of network security regarding virus and worm propagation, many providing suggested solutions to specific virus threats. Several notable online resources include:

- CERT Coordination Center at Carnegie Mellon University [<http://www.cert.org/>]
- Symantec Security Center [<http://www.symantec.com/>]
- Network Associates Security Center [<http://www.nai.com/us/index.asp>]
- WindowSecurity [<http://www.windowsecurity.com/>]
- RedHat Linux Security Alerts and News [<http://www.redhat.com/solutions/security/news/>]
- SecureStandards [<http://www.securestandard.com/>]

Further Study

As virus detection and eradication is a continuous and growing problem as new strains and threats are introduced, on-going research is required to keep abreast of threats and emerging solutions. Research in the area of viruses for this study is being continued by a CSMM student in an Independent Study during Winter 2004; results are expected to be available at the end of that term.

Footnotes and Bibliography

- ¹ Network Security Essentials: Applications and Standards, William Stallings; Prentice Hall 2003
Network Security Essentials web site, <http://www.williamstallings.com/NetSec2e.html>; Last updated: Saturday, May 24, 2003
- ² a discussion of the OSI 7-layer model is available online at: <http://www.freesoft.org/CIE/Topics/15.htm>
- ³ an online demo of SET is available at: <http://www.mastercardintl.com/newtechnology/set/>
- ⁴ RFC2570, Introduction to Version 3 of the Internet-standard Network Management Framework, April 1999; J. Case, R. Mundy, D. Partain, B. Stewart
- ⁵ an overview of the IBM Digital Immune System is available online at: <http://www.research.ibm.com/antivirus/>
- ⁶ Computer Security Threat Monitoring and Surveillance, April 1980; Anderson, J. ,
- ⁷ A State Transition Analysis Tool for Intrusion Detection, July 1992; Porras, P.
- ⁸ NSA Commercial Product Evaluation Programs website: <http://www.nsa.gov/isso/bao/cpep.htm>
- ⁹ The Practice of Network Security, Allan Liska; Prentice Hall Publishing, 2003
- ¹⁰ SSH: The Secure Shell, DJ Barrett & RSilverman; O'Reilly Press, 2001
- ¹¹ Wireless Security, MMaxim & DPollino; RSA Press/McGraw-Hill, 2002
- ¹² ColumbiTech web site press release, ComDex conference, January 2002: <http://www.columbitech.com/News/detail.asp?Id=76>
- ¹³ WebDAV working group site: <http://www.webdav.org/>
- ¹⁴ IETF RFC 2518: HTTP Extensions for Distributed Authoring – WEBDAV; February 1999. Available online at: <http://www.ietf.org/rfc/rfc2518.txt>
- ¹⁵ The Journal: Technology Horizons in Education, December 2002
- ¹⁶ F-Secure Security web site: <http://www.f-secure.com/v-descs/nimda.shtml>
- ¹⁷ Carnegie Mellon Software Engineering Institute web site: <http://www.cert.org/advisories/CA-2001-26.html>

References

RFC Editor home page, <http://www.rfc-editor.org/>; last updated on 12Aug02.

International Telecommunications Union web site, <http://www.itu.int/home/index.html>; Updated : 2003-04-28

Internet Society web site, <http://www.isoc.org/>; Copyright © 2003 Internet Society. Last Modified Tuesday, 22-Jul-2003 EDT

IPSec specs and white papers site: <http://www.ietf.org/ids.by.wg/ipsec.html>

MIT Kerberos web site: <http://web.mit.edu/kerberos/>

Verisign Network Security info site: <http://www.verisign.com/products/networksecurity/>

Pretty Good Privacy web site: <http://www.pgp.com/>

RSA S/MIME Central web site: <http://www.rsasecurity.com/standards/protocols/smime.html>

Secure Socket Layer 3 specification: <http://wp.netscape.com/eng/ssl3/>

Open SSH web site: <http://www.openssl.org/>

IEEE 802.11 Wireless Group site: <http://grouper.ieee.org/groups/802/11/>

WiFi Alliance web site: <http://www.wi-fi.org/>
