How to Cheat at Securing Windows 2000 TCP/IP

TOPIC 1: A TCP/IP Primer

TCP/IP is a network protocol based on a 32-bit addressing scheme that enables networks to be interconnected with routers. The bits in each address are separated into four sets of 8 bits, called octets, which are separated by periods. With the binary number system, 8 bits can be used to signify any number from 0 to 255, so the lowest theoretical IP address is 0.0.0.0, while the highest is 255.255.255.255. Each device, or host, on the network must have a unique IP address to communicate on the network. In order to communicate on the Internet, IP addresses must be registered with the organizations that manage the Internet so that routing can be configured correctly. Two specific network addresses and a range of network addresses are reserved for private use and are not routed on the Internet. These two network addresses— 10.0.0.0, 192.168.0.0, 172.16.0.0–172.32.255.255—are used on networks that are not connected to the Internet or connected by using network address translation (NAT) or proxy hosts. NAT and proxy hosts have two IP addresses, one on the private network and one registered on the Internet, and handle all communications between the private network and the Internet.

IP Address Classes and Subnets

As you can see in the following table, IP addresses are divided into classes, or blocks of addresses, for administrative purposes. Each class is also assigned a default subnet mask. The class structure is simply a way to manage address space. For example, the United States government might have one or two Class A address spaces instead of thousands of Class C addresses.

IP Addresses Are Divided into Three Usable Classes

Class	Range	Default Mask	Addresses per Network
A	0.0.0.0–126.255.255.255	255.0.0.0	16 million +
B	128.0.0.0–191.255.255.255	255.255.0.0	64,000 +
C	192.0.0.0–223.255.255.255	255.255.255.0	254
D	224.0.0.0–239.255.255.255	Reserved for multicast addressing	
E	240.0.0.0–254.255.255.255	Reserved for experimental use	

The subnet mask determines which bits in the IP address are the network address, and which bits are the host address. If we assumed that the default subnet mask was 255.0.0.0 for a Class A address, then from the above table it would imply that the first 8 bits (which equals 255) are the network portion of the address, and the three remaining octets are available for host addresses. It is not realistically possible to have 16 million hosts on a single network, or even 64,000, without segmenting the network with routers. Accordingly, networks with Class A and B addresses do not typically use the default mask; often, their subnet masks end up similar to those of Class C networks. When a mask other than the default is used, subnets are created that enable the address space to be split up into several smaller networks and route traffic between them. A Class B network address could be split into 255 networks by using a 255.255.255.0 subnet mask. The actual number of usable networks, however, is a bit less than 255, due to network and broadcast addresses.

Subnets and Routing

Routers are devices that connect networks together and relay traffic between networks according to routing tables that are configured in their memory. IP networks that are not on the same logical network must have a router to connect them in order for their hosts to communicate. TCP/IP neophytes are often baffled when two machines cannot “see” each other, even though they are on the same physical wire. The point they should remember is that the combination of IP address and subnet mask can segregate the physical network into logically separate networks.

Multiple routes can be configured between networks, providing TCP/IP with a measure of fault tolerance. Computers can act as routers if they are running software to perform that function. Routers are, in fact, computers designed for the specific purpose of routing network traffic. Windows NT and Windows 2000 Server can also perform the functions of routers with the Routing and Remote Access Service.

TOPIC 2: The OSI Model

The OSI model is used as a broad guideline for describing the network communications process. Not all protocol implementations map directly to the OSI model, but it serves as a good starting point for gaining a general understanding of how data is transferred across a network.

Seven Layers of the Networking World

The OSI model consists of seven layers. The number *seven* carries many historical connotations; it is thought by some to signify perfect balance, or even divinity. Whether this was a factor when the designers of the model decided how to break down the functional layers, the Seven Layers of the OSI Model are at least as legendary as the Seven Deadly Sins and the Seven Wonders of the World (within the technical community):

1. Application
2. Presentation
3. Session
4. Transport
5. Network
6. Data Link
7. Physical

Data is passed in a top-down fashion through the layers on the sending computer until the Physical layer finally transmits it onto the network cable. At the receiving end, it travels back up in reverse order. Although the data travels down the layers on one side and up the layers on the other, the logical communication link is between each layer and its matching counterpart. Here's how that works: As the data goes down through the layers, it is encapsulated, or enclosed, within a larger unit as each layer adds its own header information. When it reaches the receiving computer, the process occurs in reverse; the information is passed upward through each layer, and as it does so, the encapsulation information is evaluated and then stripped off one layer at a time. The information added by the Network layer, for example, will be read and processed by the Network layer on the receiving side. After processing, each layer removes the header information that was added by its corresponding layer on the sending side. It is finally presented to the Application layer, and then to the user's application at the receiving computer. At this point, the data is in the same form as when sent by the user application at the originating machine. descriptions start (seemingly logically) at the topmost layer. Which way you look at it depends not on which hemisphere you live in, but on whether you're addressing the communication process from the viewpoint of the sending or the receiving computer.

TOPIC 3: The TCP/IP Protocol Suite

Though the OSI model provides a well-understood and modular architecture to describe network communications, not all communication methods map directly to it—including TCP.

As TCP/IP has matured, it has grown to include a number of protocols and applications. Actually, TCP/IP is somewhat of a misnomer, because TCP isn't used by every application that communicates on an IP network. Applications use either TCP or UDP to communicate with IP on the Internet layer. The other Internet layer protocols, ARP, ICMP, and IGMP, are used by IP to resolve IP addresses to hardware addresses and route packets. In the Transport layer, UDP is a connectionless protocol and does not guarantee delivery of network packets; so most applications do not use it. Therefore, since most applications use TCP, the name TCP/IP has become common usage for referring to IP networks and components. The other common applications and protocols that are usually present in a TCP/IP implementation are included with TCP and IP to comprise what is known as the TCP/IP protocol suite. This figure shows the TCP/IP model and protocols compared to the OSI Model:

OSI Model	**TCP/IP Model**
Application	Application: Telnet, FTP, SMTP, DNS, HTTP, SNMP
Presentation	
Session	
Transport	Transport: TCP, UDP
Network	Internet: IP/ARP, ICMP, IGMP
Data Link	Network Interface: Ethernet, Token Ring, ATM, PPP
Physical	

The TCP/IP protocol suite grows as new application protocols are introduced to provide functionality for IP networks. Hypertext Transfer Protocol (HTTP) was created to transmit Hypertext Markup Language (HTML) documents over TCP/IP, and was introduced in 1991. It was swiftly absorbed into the TCP/IP protocol suite, in no small part due to its rapid proliferation throughout the Internet community. It is amazing to think that prior to 1991, the World Wide Web did not even exist, and in the space of a decade, it has become one of the primary vehicles ushering in the global village. Using some of the other TCP/IP applications has been made easier by HTTP technology. Web browsers have incorporated the ability to use File Transfer Protocol (FTP), which is much simpler than using a command-line FTP interface. They can also be used to access e-mail, providing a much simplified and ubiquitous e-mail client. Although you are probably not intimidated by command-line interfaces and SMTP configurations, many users would not even use these applications if it weren't for the World Wide Web. When viewed from a different perspective, the TCP/IP protocol suite becomes something far greater than just a conglomeration of protocols. This technology combined with others has changed the lives of a multitude of people in a startling number of ways. Discussing the details of the technology may be less appealing than considering the way it changes the world, but this is a technical book, so we must.

TCP/IP Core Protocols

There are a handful of protocols that are the mainstay of the TCP/IP suite, the real bread-and-butter protocols. These protocols handle all of the network connections and routing so that applications are simply concerned with handing data to the protocols at the Transport layer.

TCP

Transmission Control Protocol (TCP) works on the Transport layer of the TCP/IP model, providing connection-based communication with other IP hosts. When an application passes data to the Transport layer, it is often too much data to transmit in one packet, so TCP segments the data on the sending side and reassembles it at the receiving end according to sequence information that is packaged with the packet. TCP sends acknowledgments to confirm successful delivery, and analyzes each packet according to checksum information to ensure data integrity. TCP uses a system of ports to manage communication. Applications bind to a specific TCP port, and any inbound traffic delivered to that port will be picked up by the application. This enables multiple applications on one host to use TCP at the same time, and also standardizes the way a client can connect to a given service on a server. For instance, Telnet's standard TCP port is 23, so Telnet clients try to establish connections on port 23 by default. Port assignments are flexible; that is, you can change the port a client or server uses for a specific application if needed. Although Web servers typically use port 80 for HTTP communication, the Web server application can be bound to a different port, but clients will need to know which port to use in order to establish a connection since it differs from the default (see the following table).

TCP Ports Used by Common Applications

TCP	Port Application
20	FTP (data)
21	FTP (control)
23	Telnet
53	DNS zone transfers
80	HTTP
139	NetBIOS session

UDP

UDP also provides Transport layer services, and is a connectionless protocol that does not guarantee delivery or sequencing of data. UDP can be used when data transfer is not critical, or when the application is designed to ensure correct delivery of data. Since it does not acknowledge successful transfer, it is faster and uses less network bandwidth than TCP (see the following table).

UDP Ports Used by Common Applications

UDP	Port Application
53	DNS name queries
69	Trivial File Transfer Protocol
137	NetBIOS name service
138	NetBIOS datagram service
161	SNMP
520	Routing Information Protocol

IP

When TCP and UDP are ready to send data, they pass it to IP for delivery to the destination. IP is connectionless and unreliable, which is why TCP is designed to establish connections and guarantee delivery. IP does not try to detect or recover from lost, out-of-sequence, delayed, or duplicated packets. IP is the foundation of the TCP/IP protocol suite.

The Three-Way Handshake

Computers using TCP to communicate have both a send window and a receive window. At the beginning of a TCP communication, the protocol uses a three-way handshake to establish the session between the two computers. Because TCP (unlike its Transport layer sibling, UDP) is connection oriented, a session, or direct one-to-one communication link, must be created prior to sending and receiving of data. The client computer initiates the communication with the server (the computer whose resources it wants to access). The handshake includes the following steps:

1. **Sending of a SYN (synchronization request) segment by the client machine.** An initial sequence number, sometimes just referred to as the ISN, is generated by the client and sent to the server, along with the port number the client is requesting to connect to on the server
2. **Sending of an ACK message and a SYN message back to the client from the server.** The ACK segment is the client's original ISN plus 1, and the server's SYN is an unrelated number generated by the server itself. The ACK acknowledges the client's SYN request, and the server's SYN indicates the intent to establish a session with the client. The client and server machines must synchronize one another's sequence numbers.
3. **Sending of an ACK from the client back to the server, acknowledging the server's request for synchronization.** This ACK from the client is, as you might have guessed, the server's ISN plus 1. When both machines have acknowledged each other's requests by returning ACK messages, the handshake has been successfully completed and a connection is established between the two.

NOTE

Packets are often referred to as datagrams at this level. These datagrams contain the source and destination IP addresses, which will be translated to MAC (physical) addresses at a lower layer.

IP receives TCP segments (or UDP for connectionless communications such as broadcasts) and then passes it down to the Network layer. Before handing it down, however, IP performs an important function: It looks at the destination IP address on the packet and then consults its local routing table to determine what to do with the packet. It can pass the data to the network card (or if it is a multihomed system, determine which of the attached network cards to pass it to), or it can discard it. When a Windows 2000 computer starts, the routing table is constructed. Certain entries, such as the addresses for the loopback, the local network, and the default gateway (if configured in TCP/IP properties) are added automatically. Other routes can be added by ICMP messages from the gateway, by dynamic routing protocols (RIP or OSPF), or you can manually add routes using the route command at the command prompt.

ARP

The Address Resolution Protocol resolves IP addresses to Media Access Control (MAC) addresses. MAC addresses are unique IDs that are assigned to network interface devices. ARP uses a broadcast, after checking the ARP cache, to send out a query that contains the IP address of the destination host, which replies with its MAC address. When the request is answered, both the sender and the receiver record the IP and MAC addresses of the other host in their ARP table cache to eliminate the need for an ARP broadcast for every communication.

ICMP

Internet Control Message Protocol is used by network devices to report control, error, and status information. ICMP messages are delivered by IP, which means that they are not guaranteed to reach their destinations. ICMP is used by routers to indicate that they cannot process datagrams at the current rate of transmission, or to redirect the sending host to use a more appropriate route. Most of you are probably familiar with the ping utility, which sends ICMP echo requests and displays the replies it receives.

IGMP

Internet Group Management Protocol is used to exchange and update information regarding multicast group membership. Multicasting is a system of sending data to one address that is received and processed by multiple hosts. Multicast addresses are in the Class D IP address range, and addresses are assigned to specific applications. For instance, the 224.0.0.9 address is used by RIP (Routing Information Protocol) version 2 to send routing information to all RIP routers on a network (see the following table).

TCP/IP Core Protocols and Their Related RFCs

Protocol	RFCs
ARP	826
IP	791
ICMP	792
IGMP	1112, 2236
UDP	768
TCP	793

TCP/IP Applications

TCP/IP would be rather useless without applications to run on top of it. In addition to the applications that are considered part of the TCP/IP protocol suite, there are numerous proprietary applications that work on IP networks as well. For instance, NetBIOS over TCP/IP (NetBT) is Microsoft's implementation of NetBIOS for IP. Since NetBT is typically only found on Windows computers, it is not considered part of the TCP/IP protocol suite.

- **SMTP** Simple Mail Transport Protocol is a protocol designed for applications to deliver mail messages. SMTP defines the specific commands and language that mail servers use to communicate, and the format of the messages to be delivered. For instance, if an SMTP server receives a mail message that is addressed to a user that is not defined, according to SMTP standards it will reply to the sender and include information regarding the failed delivery.
- **HTTP** The child prodigy of Internet protocols, Hypertext Transport Protocol is used by Web browsers and Web servers to conduct their business with each other. HTTP defines how browsers request files and how servers respond. HTTP works in conjunction with Hypertext Markup Language (HTML), graphics, audio, video, and other files to deliver the killer application of the 1990s, the World Wide Web.
- **FTP** File Transfer Protocol is a client/server application designed to enable files to be copied between hosts regardless of the operating systems. FTP can also be used to perform other file operations, such as deletion, and it can be used from a command-line interface or a GUI

application. The latest versions of popular Web browsers include complete FTP functionality, although many shareware FTP clients offer interfaces that are faster and more powerful.

- **Telnet** Telnet is an application that enables a remote command-line session to be run on a server. Telnet is available for most operating systems, including Windows 2000. By using Telnet to log on to a server, you can run programs and perform other operations on the server. It's the next best thing to being there!
- **DNS** Domain Name System is used by most of the other applications in the TCP/IP protocol suite to resolve host names to IP addresses. A Web browser, for example, cannot establish a connection to a Web server unless it knows the IP address of the server. DNS is used to resolve host names, such as www.microsoft.com, to IP addresses. DNS is a distributed database that is essential for TCP/IP to be used on a massive Internetsize scale. It provides a function that hides the complexity of IP addresses from users, and makes things such as e-mail and the World Wide Web much easier to use.
- **SNMP** Simple Network Management Protocol was designed to provide an open systems management infrastructure for hardware and software vendors to implement on their systems. This enables management software to be developed that can query a host for information defined in its management Information Base (MIB). Devices running SNMP software can also send traps, which are simply messages formatted according to SNMP specifications, to a management server when a certain event occurs. Since SNMP is an open platform protocol, SNMP management console software can interoperate with systems of various types as long as they comply with SNMP standards.

TOPIC 4: Windows 2000 TCP/IP Stack Enhancements

The most important enhancements that Microsoft has made to the TCP/IP protocol stack in Windows 2000 are related to performance increases. These include:

- RFC 1323 TCP extensions: scalable TCP window size and timestamping.
- Selective Acknowledgments (also called SACK) in accordance with RFC 2018.
- Support for IP over ATM (Asynchronous Transfer Mode) as detailed in RFC 1577.
- TCP Fast Retransmit.
- Quality of Service (QoS).
- Resource Reservation Protocol (often referred to as RSVP).
- IP Security (IPSec).
- The Network Driver Interface Specification version 5.0.

NetBT and WINS

If you have worked with Windows in a network environment, you know that Windows computers have a computer name that is used to identify each system on the network. This computer name is the NetBIOS (Network Basic Input/Output System) name. NetBIOS, which has a history extending back to 1983, is a networking API that was used by Windows computers to register and locate resources. NetBIOS names have a maximum length of 15 characters and a flat namespace, two factors that are severely limiting on a large network.

NetBT is simply the application of NetBIOS working on a TCP/IP network, and WINS was Introduced to help manage the NetBIOS names on a TCP/IP network. WINS is a service that registers IP addresses with the associated computer names and services in a database, and responds to queries from clients who need to resolve a NetBIOS name to an IP address. Without WINS, Windows clients had to rely on broadcasts or static files located on each PC to resolve names to IP addresses. WINS was introduced to reduce the amount of broadcast traffic on a Windows network and provide the ability to resolve addresses for computers throughout a WAN. Windows 2000 has taken a big step away from NetBIOS, NetBT, and WINS, but they are still there to support existing Windows networks. NetBT uses the following TCP and UDP ports:

- UDP port 137 (name services)
- UDP port 138 (datagram services)
- TCP port 139 (session services)

Windows 2000 requires NetBIOS over TCP/IP to communicate with prior versions of Windows NT and other clients. In accordance with the move away from NetBIOS, Windows 2000 supports direct hosting to communicate with other Windows 2000 machines. Direct hosting uses the DNS (on port 445) for name resolution, instead of the NetBT.

NOTE

Windows 2000 by default enables both NetBIOS and direct hosting. When establishing a new connection, both protocols are used simultaneously, and the one that connects first is the winner. In many configurations, NetBIOS should be disabled for performance and security reasons. To force Windows 2000 to use direct hosting:

1. Click Start | Settings | Network and Dial-up Connection. Rightclick on the Local Area Connection and click Properties.
2. Select Internet Protocol (TCP/IP), and click Properties.
3. Click ADVANCED.

4. Click the WINS tab, and select Disable NetBIOS over TCP/IP.

Windows 2000 introduces several new features for WINS that improve its manageability.

DHCP

Windows has long included support for Dynamic Host Configuration Protocol on both the server and client sides, and Windows 2000 is no exception. DHCP enables clients to request the lease of an IP address from a server. The server will also automatically configure other TCP/IP items such as gateways, DNS servers, and WINS servers. Windows 2000 includes several new DHCP features, including performance monitor counters, integration with DNS, disabling NBT on clients, and detection and shutdown of unauthorized DHCP servers on Windows 2000 servers by integration with Active Directory.

DNS

Windows NT 4.0 ships with a DNS server service, and organizations that have deployed it will benefit when they upgrade to Windows 2000. As mentioned previously, Active Directory relies on DNS in order to function, and some older versions of DNS servers will not be suitable. In order for Active Directory to work, it must register SRV records with the DNS service, which are not supported on some DNS servers.

SNMP

An SNMP service ships with Windows NT and Windows 2000, enabling them to participate as SNMP managed hosts. Third-party software is also available so that a Windows NT or 2000 computer can be an SNMP network management station. DHCP, IIS, and other Windows services install custom MIBs so that they can be managed via SNMP. Microsoft Systems Management Server includes a client service, Event to Trap Translator, which converts Windows NT and 2000 events into SNMP traps. This feature is a very useful tool to integrate Windows NT and Windows 2000 into large organizations that depend on an SNMP management infrastructure.

TOPIC 5: Using TCP/IP Utilities

The Windows 2000 distribution ships with a number of command-line utilities to assist in troubleshooting TCP/IP network problems. If you have been supporting Windows NT TCP/IP (or even UNIX), you are probably familiar with most of these utilities. Some of the utilities have been enhanced, and one new utility, pathping, has been added to the tool set.

ARP

The ARP utility is not one that you will use often, but is very useful in certain situations. ARP can be used to display, delete, and add entries in the computer's ARP table. The ARP table contains IP address to MAC address assignments, and you shouldn't need to modify it except under extreme circumstances. The ARP utility is helpful when troubleshooting problems that are related to duplicate IP addresses or duplicate MAC addresses on a segment. The ARP utility allows you to add and delete entries in the ARP cache.

When you add an entry into the ARP cache, you create a static entry. A static entry will appear as static in the type field in the ARP cache. You might want to create static ARP entries for frequently accessed servers on the segment, or perhaps for the default gateway. When you create static entries, the source machine does not need to issue ARP broadcasts to resolve IP addresses to MAC addresses.

Hostname

The hostname utility simply returns the host name of the computer. There are no command-line switches.

Ipconfig

Ipconfig is a utility that can be used to display IP configuration, manage the DHCP client, and manage and display the DNS cache. New switches for the ipconfig command include /flushdns, /registerdns, and /displaydns. Running ipconfig with no switches displays the IP address, subnet mask, and default gateway for each network adapter on the computer. This is especially useful when troubleshooting to see whether a client has received a DHCP address. Let's discuss of the command-line options, since ipconfig is a utility you will probably use more than most of the other TCP/IP utilities. Important switches for ipconfig include:

- **/?** Displays command-line options, syntax, and examples.
- **/all** Displays a multitude of configuration items for all network adapters, including node type, MAC address, IP address, subnet mask, default gateway, DHCP server, and primary and secondary WINS servers.
- **/renew** You can force the DHCP client to refresh its configuration from the DHCP server by using the /renew switch.
- **/release** This switch will remove the IP configuration from all adapters with DHCP configuration. This operation can also be performed on a specific adapter by appending its name after the release switch.
- **/flushdns** The DNS cache is flushed by using the /flushdns switch with ipconfig.
- **/registerdns** This switch renews DHCP leases on adapters, and performs dynamic registration for DNS names and IP addresses. Useful in environments that use dynamic DNS.
- **/displaydns** The DNS resolver cache can be displayed by using the /displaydns switch. To be useful, you may need to pipe this command to a text file so that you can see all of it (ipconfig /displaydns > c:\temp\displaydns.txt).
- **/showclassid** Returns information on the DHCP Class ID that is configured on the client.

- **/setclassid** Class IDs on network adapters can be set by using the /setclassid switch with the network adapter name trailing it. The function of Class IDs is to control DHCP configuration for specific groups if the same configuration is not appropriate for all users.

TIP

TCP/IP parameters for Windows 2000 are stored as Registry values and can be located at HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ Services\Tcpip\Parameters. Remember to back up any keys before changing them!

Nbtstat

Nbtstat is a utility used to view protocol statistics and current TCP/IP connections using NBT. There are a number of command-line switches available to allow you to view adapter status and name tables of remote computers, local NetBIOS names, the cache of NetBIOS names, names resolved by WINS or broadcast, and session information. The following example illustrates that, if interpreted correctly, nbtstat can provide a wealth of information in a Windows network. Examining the results of issuing the command nbtstat –a 192.1.1.1 allows us to determine that the node 192.1.1.1 is a domain master browser [1B], and that the Administrator is logged on.

```
Node IpAddress: [192.1.1.1] Scope Id: []
             NetBIOS Remote Machine Name Table

Name                Type                Status
----                ----                ------
YODA                <00> UNIQUE         Registered
YODA                <20> UNIQUE         Registered
JEDI                <00> GROUP          Registered
JEDI                <1C> GROUP          Registered
JEDI                <1B> UNIQUE         Registered
YODA                <03> UNIQUE         Registered
JEDI                <1E> GROUP          Registered
JEDI                <1D> UNIQUE         Registered
INet~Services       <1C> GROUP          Registered
..__MSBROWSE__.     <01> GROUP          Registered
IS~YODA........     <00> UNIQUE         Registered
ADMINISTRATOR       <03> UNIQUE         Registered

MAC Address = 02-00-4C-4F-4F-50
```

Netstat

Netstat also displays protocol statistics and current TCP/IP connections. Several command-line switches are available to display information such as all connections and listening ports, Ethernet statistics, addresses and port numbers, connections by protocol type, the routing table, and statistics by protocol.

The netstat –s switch provides detailed statistics regarding protocol performance. You can limit which protocols are reported on by using the –p switch, or if you want performance statistics on all TCP/IP protocols, use only the –s switch.

By using a combination of the –a and –n switches, a list of open ports on the machines and their current status is displayed. The –n switch speeds up the screen print process by preventing netstat from translating port numbers to services. Try it with and without the –n switch and you'll see. *Listening* means that the port is open, but no active connections have been made to it. *Established* indicates that the connection is active. *Time-Wait* and *Close-Wait* represent connections that have been established, but are in the process of timing out and closing. The netstat command can provide you with a wealth of information. Every Systems Administrator should run this command on a periodic basis to assess the state of the ports on his servers for security reasons, and to obtain quick TCP/IP statistics. Using the /? switch will display information you need to use the utility.

TIP

A couple of things to watch out for when netstat –s statistics are displayed are the discards entries. These should be hanging around zero. If you find a large number of discards, you likely have problems with the network card itself, or the segment is very busy, and messages are lost or corrupted in the NIC buffer.

Nslookup

Nslookup is a utility used to troubleshoot DNS issues. This is one command where you cannot use the /? switch to get help on how to use the utility. Nslookup can be used as an interactive utility by running the executable with no command-line options. When nslookup is started, you will be greeted with a greater-than prompt. More information on the options available can be displayed after launching nslookup and typing **?** or **help**. The Windows 2000 Help file also has information regarding nslookup.

Ping

The ping utility (Packet Internet Groper) sends an ICMP ECHO request to the specified host, and displays statistics on the replies that are received. Ping is one of the first IP troubleshooting tools to use when you are trying to resolve a network problem. See the following table for command-line switch options for this “oldie, but goodie.”

Command-Line Switches for the Ping Utility

Switch	Description
-?	Displays syntax and command-line options.
-t	The –t switch is useful when you want to continuously monitor a connection. For example, you want to restart a machine remotely, and then want to know when the machine is up again so that you can reestablish your remote connection. Use the ping –t command and watch when the destination computer begins to respond, and then reestablish the connection.
-n count	If you don’t want to continuously ping a remote host, you can specify the number of ICMP echo request messages sent to the destination by using the –n switch.
-l size	Size of send buffer.
-f Set	Don’t Fragment flag in packet.
-i TTL	The default Time-To-Live (TTL) set on the ICMP echo messages is 252, but you can change that value by setting the –i switch.
-v TOS	Type of Service.
-r count	The –r command shows you the routes taken with each ping attempt. Think of this as a quick-and-dirty way to investigate your routing configuration.
-s count	Timestamp for count hops.
-j host-list	Loose source route along host-list.
-k host-list	Strict source route along host-list.
-w timeout	Use the –w switch to configure a custom timeout period on your requests. The default timeout is 1000 milliseconds. If you don’t want to wait that long for a timeout, change the value using the –w switch.

Route

The route command enables you to view, add, remove, or modify the IP routing table on a computer. The route table maintains four different types of routes:

- ✿ **Host** The route to a specific destination IP address.
- ✿ **Subnet** A route to a subnet.
- ✿ **Network** A route to a network.
- ✿ **Default** Used when no other route applies.

Routes, which are available even after rebooting, are called persistent routes and are contained in the Registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Persistent Routes. Use the –p switch to add a persistent route, and –f to clear the routing table. The -? switch will display usage options, and the Windows 2000 Help file can be consulted for supplementary information.

TIP

If you have partitioned one physical network into logical subnets, you can eliminate the requirement to install a router to reach a different logical subnet. This can be achieved by using the route command and then letting ARP do all the work for you. For example, on host 10.1.1.1, the command would be:

```
route add 0.0.0.0 MASK 0.0.0.0 10.1.1.1
```

Tracert

The tracert utility allows you to trace the path of routers to a destination host. You can use the tracert utility to assess whether a router on the path to the destination host may be congested.

The tracert utility sends a series of ICMP echo requests, with each request having a incrementally higher TTL value. The first echo request has a TTL of 1. When the first router receives the message, it will decrease the TTL by 1. Since the TTL on the request was 1, it now is 0, and the router will return a Time Exceeded message to the requesting computer.

The tracert utility then increases the TTL to 2 on the ICMP echo request message. When the message hits the first router, the TTL is decreased by 1, and when it hits the second router, it is decreased by 1 again. The second router then sends a time-exceeded message to the source host. The process continues until all the routers have been traversed to the destination host.

See the following table for command-line options, or just run the executable without indicating a target system, and the command usage will be displayed.

Tracert Command-Line Options

-d	Don’t resolve addresses to host names.
-h max_hops	Maximum number of hops to target.
-j host-list	Loose source route along host-list.
-w timeout	Milliseconds to wait for replies.

Pathping

Pathping, a utility that is new to the Windows operating system, discovers the route to the destination host, pings each hop for a period of time, and then reports the statistics. The PATHPING utility sends ICMP echo request messages to each router along the path to the destination host, and calculates how long it takes the roundtrip from request to reply. The default number of hops is 30, period 250 milliseconds, and queries to each router 100.

NOTE

The Pathping tool combines the capabilities of both tracert and ping, and gives you additional information that you can't get easily from using either tool individually. Pathping will calculate roundtrip times, percent of requests that were lost at each router, and percent of requests lost between the routers.

Pathping provides some interesting statistics because it gives you information regarding where the packet loss is taking place, and the level of stress a particular router may be experiencing.

Note that PATHPING first does a tracert and identifies all the routers in the path to the destination, and provides a list of those routers in the first section. Then, PATHPING provides statistics about each router and each link between routers. From this information, you can assess whether a router is being overloaded, or whether there is congestion in the link between the routers (see the following table).

The last two columns provide the most useful information when troubleshooting routers and links. Notice in the last column the name of the router, the IP address, and the percentage to the left of the router. If there is a high number of lost pings to a router, that is an indication that the router itself may be overloaded.

Pathping Command-Line Switches

Switches	Description
/?	Displays pathping options.
/n	Do not resolve address to host names.
/h maximum_hops	Maximum number of hops to destination.
/g host-list	Loose source route along host-list.
-p period	Number of milliseconds between pings.
-q num_queries	Number of pings per hop.
-w timeout	Milliseconds to wait for each reply.
-T	Test each hop with Layer-2 priority tags.
-R	Test each hop for RSVP awareness.

Just under the name of the router, you see a | character. This represents the link between the router and the next-hop router. When there is a large percentage of lost pings for the link, it indicates congestion on the network between hops. In this case, you would want to investigate problems with network congestion rather than with the router itself.

NOTE

The pathping algorithm takes advantage of the fact that there are two paths the ping request can take: the fast path and the slow path. The fast path is that taken when a router just passes the packet to the next hop, without actually doing any work on that packet. This is in contrast to the slow path, where the router is the recipient of the ICMP echo request and must use processing resources to respond to the request by issuing an ICMP echo reply.

Netdiag

The netdiag command is new with Windows 2000. It is the Swiss Army Knife of network diagnostics for your Windows 2000 installation. When you run this command, it sets forth to test 24 different aspects of the networking subsystem for the machine.

When netdiag is run without any switches, it prints the results to the screen. But, you will likely want to save the results of the analysis, and netdiag allows you to save everything it has discovered to a log file, which you can read at your leisure (or send to somebody else so he or she can figure out what's wrong!).

Perhaps the greatest value of the netdiag command is you can easily tell a user or a junior Administrator to run this command and not have to worry about walking him or her through 24 different command-line tests and switches, which would in all probability lead to a minor disaster.

A list of the tests run when the netdiag command is issued without switches appears in the following table.

Tests Run by Netdiag

Test	**What the Test Does**
Ndis	Tests the NIC.
IpConfig	Runs ipconfig.
Member	Tests the machine's Domain Membership.
NetBTTransports	Tests NetBIOS over TCP/IP Transports.
Autonet	Autonet address test.
IpLoopBk	Pings the loopback address.
DefGw	Pings the default gateway.
NbtNm	NetBT name test.
WINS	Tests the WINS servers.
Winsock	Tests Winsock integrity.
DNS	Tests that correct names are entered in DNS.
Browser	Tests the Workstation Services and Browser Service.
DsGetDc	Discovers Domain Controller availability.
DcList	DC list test.
Trust	Tests Trust Relationships.
Kerberos	Kerberos test.
Ldap	Tests Lightweight Directory Access Protocol.
Route	Tests the routing table.
Netstat	Runs netstat and records the results.
Bindings	Bindings test.
WAN	Tests the WAN configuration.
Modem	Performs Modem Diagnostics.
Netware	Tests NetWare connectivity.
IPX	Tests IPX components.

The netdiag command includes several switches, which you can find by typing **netdiag /?** at the command prompt. The /q switch will only show you the errors that netdiag finds, so that your screen (hopefully) does not get too busy with the results from all the tests. If you want the real nitty-gritty details, use the /v switch to get the verbose output printed to the screen. If verbosity is your middle name, use the /debug switch to wring out every possible bit of information and print that to the screen. The most useful switch is the /l switch, which allows saving all the output to a log file.

When you have users at a remote site reporting problems with connectivity, have them run netdiag with the /debug and the /l switches. Then have them e-mail the NetDiag.log file to you as an attachment. This is an excellent way to start troubleshooting without having to ask a lot of questions of someone who might have marginal understanding of the networking subsystems of the machine. Make the netdiag utility your first line of offense when troubleshooting connectivity programs. An entire report takes less than a minute to complete, and the information gathered is invaluable.

SNMP

The Simple Network Management Protocol is not a utility in and of itself. Rather, it is a protocol used to communicate status messages from devices distributed throughout the network to machines configured to receive these status messages. Machines that report their status run SNMP Agent software, and machines that receive the status messages run SNMP Management software.

How Does SNMP Work?

SNMP allows you to audit the activities of servers, workstations, routers, bridges, intelligent hubs, and just about any network-connected device that supports the installation of agent software. The agent software available with the Windows 2000 implementation allows to you monitor Windows 2000 Server and Professional operating system parameters, the DHCP service, the WINS service, the Internet Information Services, QoS Admission Control Services, the Routing and Remote Access Service (RRAS), and the Internet Authentication Service (IAS). All these Windows 2000 services can be monitored remotely by SNMP Management software.

In order for agent software to collect information regarding a particular service, a Management Information Base (MIB) must be created.

NOTE

The MIB is a database and a collection of instructions about how and what information should be gathered from a system. The MIBs included with Windows 2000 allow the agent software to communicate a wide range of information.

The agent is responsible for reporting the information gathered by the MIB. However, agents rarely volunteer information spontaneously. Rather, the agent must be queried by an SNMP management system before it gives up its knowledge.

There is an exception to this: a trap message. A trap message is sent spontaneously by an agent to SNMP Management System for which is has been configured to send. For example, we could set a trap message to indicate that the World Wide Web service is hung. We would then configure the agent to send a trap message to the IP address of our computer running the SNMP Management software so that we can quickly handle this catastrophic event. SNMP messages themselves are sent to UDP Port 161 for typical GET and SET type messages, and UDP Port 162 for trap messages.

NOTE

A GET message is a request that is sent from an SNMP Management System requesting information from an agent. A SET message allows the SNMP Management System to write changes to MIB, and therefore extend its information-gathering abilities.

Installing the Agent

In order for a system to report to the SNMP Management System, you have to install the agent software first. To install the agent on Windows 2000 machines, go to the Control Panel, open the Add/Remove Programs applet, select Add/Remove Windows Components, scroll down to find Management and Monitoring Tools and select it, then click Details. Place a check mark in the Simple Network Management Protocol check box, and click Ok.

Once the agent software is installed, its behavior can be configured. The way to configure the SNMP agent behavior in Windows 2000 is by launching the Services applet from Administrator Tools | Services. Then scroll down to the SNMP Service. After you install the service, it should start automatically. Right-click on the SNMP Service entry, click Properties, and click the Agent tab. This tab is for descriptive purposes only. SNMP Management Systems can obtain information about a contact person and location from information provided here. Also, information about what type of system the agent is running on is indicated by the selections made in the Service frame area. Click the Traps tab.

If you want the agent to initiate a trap message, you need to make the agent part of a community that the agent and the SNMP Management software have in common. The community name can be anything you like, and it is not related to domain names, usernames, or any other security principle you might think of in Windows 2000.

WARNING

The community name does represent a somewhat primitive degree of security, because only machines from the same community can communicate with the agent. Microsoft documentation states that you should make your community name hard to guess. However, since the community name is transmitted in clear text, it really doesn't make much of a difference how difficult to guess the name of the community might be!
One way around this problem is to use IPSec encryption between the SNMP Management station and the SNMP agent. In this way, the cleartext messages are encapsulated in encrypted IPSec packets and are not vulnerable to network sniffers.

After configuring at least one community membership, you then need to enter the IP addresses or host names of the machines that will receive the trap message. You do so by clicking Add under the Trap destinations text box. On the Security tab, you can configure some basic security parameters for the SNMP agent. In the "Accepted community names" frame, you can add new communities that the agent can report to, and define the level of permissions for Management Station access to the agent and MIBs.

After clicking Add, the SNMP Service Configuration dialog box is displayed. Several security rights can be configured for the community:

- ✿ **None** means no permissions.
- ✿ **Notify** means only traps will be sent to the Management Station, and that the Management Station cannot make SNMP requests.
- ✿ **Read Only** allows the Management Station to read the values of the information provided by the MIBs.
- ✿ **Read Write** and **Read Create** do the same thing, which is to allow a SET command to be sent to the agent.

One really nice addition to the Windows 2000 SNMP agent is a GUI utility that allows you to configure which events will elicit a trap message. By default, no events will send a trap, which isn't very useful. However, there is a GUI utility that you can access from the Run command. Type **evntwin.exe** at the Run command and click Ok.

This launches the Event to Trap Translator, which allows you to con figure which events will elicit trap messages. Notice the DEFAULT option button is selected, and list of events that are configured to send trap messages by default. That's right, none! In order to configure trap events, click CUSTOM, and then click EDIT. In the lower-left pane titled Event sources, double-click on the Security folder. You should see another security folder under that one. Click on that security folder, scroll down to Event ID 529, and click on that. Note that in the lower-right pane, you are able to select from a number of different security events for which you can elicit trap messages to be sent to a management station. After selecting Event ID 529, click ADD. You can decide if the trap will be sent after a certain number of instances take place over a specified time interval. Click OK, and this event will be listed in the top pane of the Translator window. If you prefer a command-line version of this program, type **evntcmd.exe** at the command prompt and you will receive some help on how to use the command-line version of the program.

TOPIC 6: Using Windows 2000 Monitoring Tools

At times it is necessary to collect information about the state of the network (and TCP/IP) by drilling down deeper into its technical core. This can take the form of network analysis where TCP/IP traffic is captured and analyzed, or system monitoring where an individual host is monitored for particular system activity. The tools described in this section are extremely useful for analyzing not only TCP/IP activity, but also a plethora of other protocols, system objects, and activities. Microsoft has included two powerful network-monitoring tools with Windows 2000: the Performance Console and the Network Monitor. With these tools, you can monitor the health of your network from a single location, and you can listen in on network activity in real time. Both of these utilities allow you as the Administrator to have more control over the health and efficiency of your network.

Basic Monitoring Guidelines

When monitoring aspects of your network, you need to have a good idea of what it is that you're looking for. Are you looking for clues for logon validation errors? Are you looking for reasons for complaints of network sluggishness from users? Are you looking for possible security leaks? Are you just obtaining baseline measurements so that you have something to compare to when the network is acting abnormally? When monitoring, a few basic steps should be followed:

1. **Baseline** This is the process of collecting information on a network when everything is working the way you want it to work. It would make no sense to collect baseline information when the network is acting up, or is the subject of complaint and ridicule.
2. **Document** A system must be in place that allows you to quickly and efficiently return to previous measurements, and to measure trends that may exist in the measurements you have taken.
3. **Back up** It is important that you back up this information to multiple locations for fault-tolerance reasons.
4. **Analyze** After you have decided on a location to keep your precious data, you need a system to collate it and bring it together so that you can spot trends.

Performance Logs and Alerts

The application formerly known as Performance Monitor has undergone a name change and a minor overhaul in its appearance in Windows 2000. In fact, it appears to have a couple of different names, depending on the Microsoft documentation you read. It is called either Performance or System Monitor. You can use the Performance Console to obtain real-time data on network performance parameters such as TCP, Web, FTP, and Proxy server statistics. This information can be saved in a log file for later analysis, and it can even be replayed. To open the Performance Console, go to the Administrative Tools and click Performance. Note that there are two panes in the Performance Console. On the left, you see entries for the System Monitor, and then several options for Performance Logs and Alerts. The System Monitor is the counterpart of the Windows NT 4.0 Performance Monitor. There are three views available in the System Monitor:

- ✿ Chart view
- ✿ Histogram view
- ✿ Report view

When working with the Chart view, note that it will display up to 100 units of time. You select the unit of time for which measurements are taken by right-clicking anywhere on the chart area itself, and selecting Properties. Notice the area next to the "Update automatically" field to enter the update period. You can enter the number of seconds you want the chart updated, and the entire chart will contain data for up to 100 update intervals.

TIP

If you would like to see an entire day's worth of activity on one chart screen, you could divide the number of seconds in one day by 100, or 86400/100 = 864 seconds. By setting the chart interval to 864 seconds, you'll be able to see an entire day's worth of data on a single chart screen.

Counters

There are a great variety of network-related counters that can be added to the System Monitor. A noncomprehensive list of these counters includes IP, IIS Global, ICMP Browser, FTP Server, UDP, TCP Redirector, SMTP Server, and Network Interface.

One of the nice things about the System Monitor application in Windows 2000 is that you can populate the Chart view with a number of counters without having to repopulate the Report view. To select all counters from a performance object, select the "All counters" option button and click ADD. After the counters are added to the Chart view, statistics gathered from those counters are displayed in both the Report and the Histogram views. If you would like to create a log file to view the information at a later date, click on the Counter Logs object, then right-click in the right pane and select New Log Settings. Input the name of the log into the New Log Settings dialog box. Make it something meaningful and descriptive so you can find the information later. The first tab displayed is the General tab, and this is where you begin to add new counters to the log file. Click ADD and add counters as you did in the Chart view. After adding the counters, they will populate the area labeled Counters.

Log File Format

In the Log file type drop-down list box, you can choose what format you want the log file to be saved in. The main choices are binary format and delimited text formats. If you save the logs in delimited text formats, you can import the data into an Excel or Access database. Regardless of the format you choose, you can still bring the information back to the System Monitor Console for later analysis in the same way you were able to open log files for later viewing using the Windows NT 4.0 Performance Monitor.

Alerts

To create an alert, click the Alerts object in the left pane, and then rightclick in the right pane and select New Alert Settings from the context menu. Enter the name of the alert and click OK. Counters are added for alerting by clicking ADD. The Actions tab allows the setting of what actions should be taken if the alert is triggered. This action can take the form an entry in the application event log, a network message, starting up of a performance log, or the running of a program. Remember that if alerts are to be sent to a NetBIOS name, then it must be enabled on both the machine generating the alert and the machine receiving an alert. With the Schedule tab, the system can be instructed to look for alert conditions at certain specified times.

Network Monitor

The Microsoft Network Monitor is a software protocol analyzer that captures and analyzes traffic on the network. The version of Network Monitor that ships with the Windows 2000 server family has unfortunately been limited in scope by not allowing the network adapter to be placed in promiscuous mode .

When an adapter is placed in promiscuous mode, it is able to listen to all the traffic on the segment (also referred to as a collision domain), even if that traffic is not destined for the machine running the Network Monitor software. However, one of the advantages of this state of affairs is that because promiscuous mode capturing can potentially overtax your computer's processor, it won't happen.

Even with these limitations, Network Monitor is an extremely useful tool for assessing network activity. It can be used to collect network data and analyze it on the spot, or to save recorded activities for a later time. Network Monitor allows network activity to be monitored and triggers to be set when certain events or data cross the wire. This could be useful, for instance, when looking for certain key words in e-mail communications moving through the network.

Filtering

The Network Monitor program captures only those frames that you are interested in, based on protocol or source or destination computer. More detailed and exacting filters can be applied to data that has already been collecting, which allows you to pinpoint the precise elements you might be looking for in the captured data. We'll discuss how to filter what data you want to capture, and how to fine-tune the captured data after you've collected it.

Security Issues

The Network Monitor program is a network sniffer. Any person with Administrative privileges can install it on a Windows 2000 server family computer and start listening to activity on the wire. If you feel this is a cause for concern, you are correct. This easy availability of such a powerful tool should lead to even further consideration during the assignment of administrative privileges. Fortunately, the Network Monitor is able to detect when someone else on the segment is using Network Monitor, and provide you with his or her location. However, the usefulness of this feature is in doubt due to a lack of consistent results during testing.

Using Network Monitor

Network Monitor is not part of the default installation and can be installed via the Add/Remove Programs applet in Control Panel. After you have installed the program, go to the Administrative Tools menu and click Network Monitor. If multiple adapters are installed on the machine, you may be asked to pick a default adapter. The Network Monitor capture window will then be displayed consisting of four panes.

Capture Window Panes

The top-left pane is depicted with a gas-gauge type format, providing realtime information on percent network utilization, broadcasts per second, and other parameters. Just below that is a pane that provides information about individual sessions as they are established, showing who established a session with whom, and how much data was transferred between the two. The right pane is the local machine's session statistics pane, and provides detailed summary information about the current capturing session. The bottom pane provides information about each detected host on the segment, and statistics gathered on the host's behavior.

TIP

To determine other instances of Network Monitor currently on the network, select the Tools menu, and then click Identify Network Monitor Users. Nbtstat can also be used to track down Network Monitor users, since Network Monitor registers NetBIOS names with a service identifier of [BFh] or [BEh].

Buffer

By clicking the Capture menu item and selecting Buffer settings, you can configure Network Monitor's buffer size and frame size. The buffer size, in megabytes, determines the amount of data that can be captured in a single recording session. Since the buffer is eventually written to disk, remember to ensure that there is more available hard disk space than the amount specified in the buffer size. The second setting in the Capture Buffer Settings window is frame size, which determines how many bytes of the frame should be captured.

Collecting Data

Now that we're finished with the preliminaries, let's get to the job of collecting some data. The first thing to try out is a capture without filters, just to get a feel for how the capture process works. There are a couple of ways to get the capture started: by either selecting the Capture menu and then clicking Start, or clicking the little right-pointing arrow in the toolbar. Either one will begin the capture. When it is running, you'll see the gas gauges moving, and the statistics being collected on the recording session. After letting the capture run for a little bit, or after the % Buffer Used value is 100, click the button that has the eyeglasses next to a square (the stop and view button). This stops the capturing process and provides a view of the frames that have been captured. This window provides a list of all the frames that were captured during the session. If you scroll to the bottom of the list, you'll note that there is a summary frame that contains statistics about the current capture. Take note of the column headers, which are pretty self-explanatory. After double-clicking one of the frames, the display transforms into a tri-pane view. The middle pane contains translated information from the captured frame detailing frame headers and protocol information. The bottom pane presents the raw Hex and translations of the collected frame data. At the very bottom of the window, in the status bar area, there is a description of the frame selected in the top pane (which in this case is Ethernet/802.3 MAC Layer), the frame number out of the total number of frames, and an offset value for the selected character in the bottom pane.

In the preceding example, frame number 244 was selected, which is an ARP broadcast frame. Notice the detail in the middle pane. It indicates the hardware type and speed, and the source and destination IP and hardware address. The destination hardware address is the Ethernet broadcast address [FFFFFFFFFFFF], because the whole purpose of the ARP broadcast is to resolve the IP address to a hardware address.

The capture was taken from EXETER. The ARP broadcast was issued by CONSTELLATION for DAEDALUS, which is the machine with the IP address of 192.168.1.3. Would the ARP reply be found later in the capture? The answer is no, because the reply will not be sent to the hardware broadcast address, but to CONSTELLATION's hardware address; therefore, the Network Monitor on EXETER would be able to capture that conversation. The only reason the ARP request was captured initially was because it was directed to the hardware broadcast address, which means that every machine on the segment had to evaluate the request to see if it was for them.

The bottom pane in this instance isn't very exciting. It shows the Hex data on the left and an ASCII translation on the right.

Filtered Captures

The advantage of doing an unfiltered capture is that data can be gathered on every communication in to and out of the computer doing the capture. However, this method may result in an inordinate amount of information, some of which is unnecessary and could serve to obscure the data that is actually being looked for. If, for example, it is only necessary to capture conversations to one specific host, the captured frames could be limited by using a capture filter.

The purpose of the capture filter is to limit the frames that are actually saved in the capture buffer. This also makes better use of buffer space, since the buffer can be devoted to the precise targets of interest. It also reduces the amount of extraneous information (sometimes called noise) that could obscure important information. In order to create a capture filter, select the Capture menu, and click Filter. Click OK to pass through the warning dialog. A Capture Filter dialog box will then be displayed. There are two ways to filter capture information:

- By machine address pairs
- By a specified pattern in the frames that are examined during the capture sequence

Filtering by Address Pairs

Up to four address pairs can be defined for filtering. For example, suppose there are 30 computers on a segment that is running Network Monitor, and only capture information from four specific computers is required. To start adding address pairs, double-click on the [AND] (Address Pairs) statement. A close look at the elements of the dialog box reveals two option buttons, Include and Exclude. Any address pair selected for Include will be included in the capture. Any address pair selected for Exclude will be excluded from the capture. For example, if *Any was selected (which indicates all frames coming to and leaving this computer), then a pair of computers could be excluded so that messages being sent to and arriving from that machine are ignored.

Under the Include and Exclude options are three panes: Station 1, Direction, and Station 2. Station 1 and Station 2 will define the computers named in the address pairs that will be included or excluded from the filter, with Station 1 always being the machine running the Network Monitor application. The Direction arrows allow you to filter based on the direction of the traffic. The ← → symbol represents traffic leaving Station 1 to Station 2 and arriving from Station 2 to Station 1, the → represents traffic leaving Station 1 to Station 2, and the ← represents traffic arriving from Station 2 to Station 1.

The chances that the machine that you wish to designate as Station 2 is not included in the list are relatively high. To add the machine of interest to the list, click EDIT ADDRESSES. This shows the Addresses Database in its current state on the machine running Network Monitor. The first column gives the machine's NetBIOS name, the second column the machine's addresses, the third column denotes the type of address included in the second column, and the fourth column includes a comment about the entry in the database.

To add a new entry, click ADD. In the Add Address Information dialog box, enter the name of the machine, whether this is a permanent name for the machine, the address, the type of address, and an optional comment. Click OK, and the address is then entered into the database. These addresses will only stay in the database for the time that Network Monitor is open. If several addresses have been added, it is a good idea to save these addresses. To do so, click SAVE, and choose a location and a name for the file. The addresses can then be loaded during subsequent monitoring sessions. After clicking CLOSE, the Address Expression dialog box is displayed again.

TIP

The filtering process can be processor intensive, especially in the case of complex filters. Keep this in mind before running an extended capture session on a machine that is already heavily taxed.

Now the capture session can commence. Click OK in the Capture Filter dialog box to remove it from sight. To start the capture, click the rightpointing arrow in the toolbar. After letting the capture run for a very short period of time, click the stop and view button on the toolbar.

Display Filters

Now that some data has been captured, the second filter type can be applied, known as a display filter. The display filter allows the captured data to be mined for very specific elements, allowing for a much more refined filtering than can be accomplished with the capture filter.

NOTE

A display filter can be used as a database search tool, where the capture frames are the data in our database.

Assume that the purpose of capturing the data is to determine what types of messages are being passed around the network regarding Windows 2000. The first decision is to determine what kind of messages need to be searched for. In this case, assume the requirement is to determine if users have been using the net send command to exchange ideas or opinions regarding Windows 2000.

To get started, select the Display menu (from the Capture Summary screen), and click Filter. Everything other than the protocol of interest needs to be filtered out, and then a key phrase contained within the protocol of interest needs to be identified. It is common knowledge that Net Send uses the SMB protocol, so the search will begin there. Double-click on the line that says Protocol==Any to display the Expression dialog box .

Notice that the Protocol tab is the default. By default, all protocols are enabled, which means that the filter is letting frames from all protocols appear. The objective is to allow only frames from the SMB protocol to appear. The first step is to click DISABLE ALL. This causes all the protocols to be moved to the right pane, into the Disabled Protocols section. The SMB protocol can then be found by scrolling through the disabled protocols. Click on the SMB protocol, and then click ENABLE. When the display filter is enabled, only the SMB frames will be visible. However, only the SMB frames that contain the term *Windows 2000* need to be displayed. In order to drill down to just those frames, click the property tab. After clicking the Property tab, scroll down the list of protocols until the SMB protocol is found. Double-click on the protocol to see all the SMB frame properties. Then scroll down the list of SMB frame properties until the Data property is found.

If you select the contains option in the Relation text box, you will filter out any SMB frames that do not contain the text string Windows 2000. Note toward the bottom of this dialog box there are two option buttons, Hex and ASCII. After selecting ASCII and clicking OK, and then OK again, a single frame containing a reference to Windows 2000 is displayed.

TOPIC 7: Secure Sockets Layer

The Secure Sockets Layer (SSL) describes an encryption technology widely used on the Internet to secure Web pages and Web sites. In this section, we take a mile-high view of SSL and discuss the methods used by SSL to encrypt information to keep it secure. SSL is classified as a Transport layer security protocol, since it secures not only the information generated at the Application layer, but at the Transport layer as well. It is considered a secure protocol by providing the mechanisms for supporting the basic elements of secure communications, namely:

- Confidentiality
- Integrity
- Authentication

Authentication ensures that the information received is indeed from the individual believed to be the sender. Integrity guarantees that the message received is the same message that was sent, while confidentiality protects data from inspection by unintended recipients.

SSL lies between the Application and the Transport layers. It protects information passed by application protocols such as FTP, HTTP, and NNTP. An application must be explicitly designed to support SSL's security features. Unlike Layer 3 protocols, it is not transparent to Application layer processes.

SSL uses several protocols to provide security and reliable communications between client and server SSL-enabled applications. Specifically, the handshake protocol negotiates levels and types of encryption, and sets up the secure session. These include SSL protocol version (2.0 or 3.0), authentication algorithms, encryption algorithms, and the method used to generate a shared secret or session key.

SSL uses a record protocol to exchange the actual data. A shared session key encrypts data passing between SSL applications. The data is decrypted on the receiving end by the same shared session key. Data integrity and authentication mechanisms are employed to ensure that accurate data is sent to, and received by, legitimate parties to the conversation. SSL uses an alert protocol to convey information about error conditions during the conversation. It is also used by SSL hosts to terminate a session.

How a Secure SSL Channel Is Established

To understand how a secure channel is formed, let's examine how an SSL client establishes a session with an SSL Web server:

1. A URL is entered into a Web browser using https rather than http as the protocol. SSL uses TCP Port 443 rather than Port 80. The https entry requests the client to access the correct port on the target SSL Web server.
2. The SSL client sends a client Hello message. This message contains information about the encryption protocols it supports, what version of SSL it is using, what key lengths it supports, what hashing algorithms to use, and what key exchange mechanisms it supports. The SSL client also sends to the SSL server a challenge message. The challenge message will later confirm the identity of the SSLenabled server.
3. The server then sends the client a Hello message. After examining methods supported by the client, the server returns to the client a list of mutually supported encryption methods, hash algorithms, key lengths, and key exchange mechanisms. The client will use the values returned by the server. The server also sends its public key, which has been signed by a mutually trusted authority (a digital certificate of authenticity).
4. The client then verifies the certificate sent by the server. After verifying the server certificate, the client sends a master key message. The message includes a list of security methodologies

employed by the client and the session key. The session key is encrypted with the server's public key (which the server sent earlier in the server Hello message).

5. The client sends a client finished message indicating that all communications from this point forward are secure.

Almost all messages to this point have been sent in clear text, implying that anyone listening in on the conversation would be able to read all parts of the exchange. This is not a problem, since no Information other than the session key is secret. Moreover, the session key is safe because it is encrypted with the server's public key. Only the server is able to decrypt the session key by using its private key. The next series of events takes place in a secure context.

1. The server sends a server verify message to the SSL client. This message verifies that the server is indeed the server with which the client wishes to communicate. The server verify message contains the challenge message the client sent earlier in the conversation. The server encrypts the challenge message with the session key. Only the legitimate server has access to the session key. When the client decrypts the challenge message encrypted with the session key, and it matches that sent in the challenge, then the server has verified itself as the legitimate partner in the communication.
2. The last message used to set up the secure SSL channel is the server finish message. The SSL server sends this message to the SSL client informing of its readiness to participate in data transmission using the shared session key. The SSL session setup is complete, and data passes through a secure SSL channel.

The setup procedure is dependent on several security technologies, including public key encryption, symmetric encryption, asymmetric encryption, message hashing, and certificates. In the following sections, we define these terms and see how SSL uses them to create a secure channel.

Symmetric and Asymmetric Encryption

The two major types of encryption algorithms in use today make use of either symmetric or asymmetric encryption keys. Symmetric techniques use the same key to encrypt and decrypt information, and asymmetric methods use different keys to encrypt and decrypt data. Both types of encryption are examined in the coming sections.

Symmetric Encryption

Symmetric encryption uses the same key to lock and unlock data. There are two elements involved in the data encryption process: an encryption algorithm and a key. The most commonly used symmetric encryption algorithm is the Data Encryption Standard (DES). There are actually several flavors of DES, each using a different encryption methodology and key length. Single DES uses a 56-bit encryption key, while a stronger form of DES, known as Triple DES or 3DES, uses a 168-bit encryption key. The advantage of triple DES with its longer key length is that it provides a higher degree of security. However, this advantage is not achieved without cost: 3DES is slower than DES. In general, symmetric encryption algorithms are faster than asymmetric ones.

An obvious question when considering symmetric encryption is, how is the value of the encryption key known? It could be sent with the message, but if someone intercepted the message, he or she would have access to the key. This is analogous to writing your PIN on the back of your automated teller machine card. The key could be sent via courier; however, that would take time, prove to be expensive, and make it difficult to change keys frequently. A method is required to allow keys to be changed frequently to guard against an intruder discovering the identity of the key.

Asymmetric Encryption

We know that data can be swiftly and securely encrypted using symmetric encryption, but a method is still required to exchange the shared session keys used to encrypt data passing between secure partners. To exchange the shared session key, a secure mechanism that is fast and inexpensive is required. To provide secure passage for shared session key exchange, asymmetric or public key encryption is used.

A Public Key Infrastructure (PKI) uses key pairs: a public key and a private key. The public key is available to anyone and everyone, and is not considered confidential. The private key, on the other hand, is secret, and is available only to the rightful owner of the private key. If the private key is stolen, it is no longer valid, and any messages from the owner of that private key are suspect.

Messages can be encrypted using either the public key or the private key. When a message is encrypted using a public key, a secret message is being sent that cannot be read (decrypted) by anyone other than the holder of the corresponding private key. By encrypting a message with someone's public key, you are assured that no one but the owner of the corresponding private key can read (decrypt) it. Encrypting a message using the recipient's public key provides a digital envelope for the message.

If the sender of a message wants the recipients to be sure of the message's origin, it is encrypted with the sender's private key. Consequently, anyone with the sender's public key can open the message. When you encrypt a message with your private key, it is termed *signing the message*. No one else can sign a message with your private key, since you are the only one who has access to it. Encrypting a message with a private key provides a type of digital signature.

NOTE

The basic concepts of public and private keys can be boiled down to: Messages encrypted with a public key are secret, and can only be read by the holder of the corresponding private key. Messages encrypted with a private key can be read by anybody, since it can be decrypted using the freely available public key. Private key encryption provides a way of signing a message.

Consider the following example: A lawyer needs to send a confidential message to a client. To ensure that only the client can read the message, the lawyer encrypts it with the client's public key. Remember that the client's public key is freely available. When the client receives the message, he decrypts it with his private key, since only the client's private key can decrypt a message encrypted with the same client's public key. Additionally, since no one else has access to the client's private key, the message has consequently remained private between the lawyer and the client.

Though the lawyer is sure that message has remained confidential, how does the client know that the message actually came from the claimed source, his lawyer? Perhaps a third party impersonated the lawyer and set up the secure communication channel. To assure the client that the message was from the lawyer, the lawyer encrypts the message with his private key. The only way the client can then read the message is by decrypting it with the lawyer's public key. Only messages encrypted with the lawyer's private key can be decrypted with the lawyer's public key. If the message cannot be opened with the lawyer's public key, then the client knows the message did not come from him. When a message is encrypted using a private key, the source of the message can then be authenticated.

Hash Algorithms

Using public and private key pairs, we can confirm the authenticity of a message and maintain its confidentiality. But how do we validate the integrity of a message? In other words, how do we know that the message sent by the lawyer to the client was not changed in transit?

Hash algorithms are used to accomplish this task. The two most commonly used hash algorithms are Message Digest 5 (MD5) and Secure Hash Algorithm 1 (SHA-1). These hash algorithms take the content of a message and convert it to a constant-length string. These hashes are safe to transmit because the hashed output cannot be reverse engineered to reproduce the original message; in other words, they are a one-way mathematical function. The hashed output can be used to create a digital signature for the document. To create a digital signature, the hashed output (also known as the message digest) is encrypted with the lawyer's private key. When the document is received, the message is run through the same hash algorithm. After running the hash algorithm on the message, a message digest based on the document received is created. Then the digital signature is decrypted using the lawyer's public key. Finally, the digest attached to the message and the one generated by the client are compared. If they are the same, the document received is indeed the one that was sent. If the digests differ, then the message has been altered in transit. As you can see in this example, the digital signature provides two functions: authentication and message integrity. The sender is authenticated because the recipient was able to decrypt the message digest using the sender's public key. Message integrity was also ensured, since the digest calculated proved the same as the one sent with the message. Unfortunately, there is still one more conundrum to resolve. Recall how the client receives the lawyer's public key—it was sent to the client directly. How does the client know it was really the lawyer who sent him the public key?

This problem can be solved by using digital certificates of authority.

Digital Certificates

A digital certificate is a public key signed by a mutually trusted third party. The trusted third party signs your public key by first hashing your public key, and then encrypting the message digest with its private key. If I can open the message digest using the mutually trusted third-party's public key, and successfully decrypt messages with your public key, then I know for sure that you are the one who sent the message. I am able to authenticate you by virtue of your digital certificate.

Continuing with the lawyer/client analogy, suppose the client wants to verify the lawyer's identity. The client asks the lawyer for his public key. The lawyer responds by providing a public key that has been signed by a party trusted by both the lawyer and the client. The trusted third party has confirmed the lawyer's identity. The client already possesses the public key of the trusted third party, and uses it to decrypt the message digest of the lawyer's public key. If they match, then the lawyer's identity has been confirmed. The lawyer has then been authenticated.

Certificate Authorities

A certificate authority (CA) is responsible for verifying the identities of those who hold certificates signed by them. A certificate authority is a trusted third party. You can create your own key pair, and submit it to the CA for signing, or you can request the CA to create a signed key pair for you. The CA will verify your identify via telephone, personal interview, e-mail, or a combination thereof.

The public key of the CA must be signed too. How do you know that the public key from the certificate authority is valid? Because its certificate is signed too! Certificate authorities can consist of a chain of certificate authorities. On top of this chain or hierarchy is the root certificate authority. Subauthorities are child authorities. Each child authority has its digital certificate signed by a certificate authority above it in the hierarchy. These higher-level certificate authorities are parent authorities. The single point of failure for security in this scheme is the certificate root authority. If the private key of the root authority is compromised, all signed certificates from the root, and all its child authorities, are suspect and should be considered invalid. Similarly, whenever a private key from any child authority is breached, all signed certificates from that child authority and all of its children, are also compromised, and must be considered invalid.

One method to protect against fraud when private keys of certificate authorities are compromised is to publish a Certificate Revocation List (CRL). The certificate authority makes public the serial numbers of invalid certificates. The CRL contains a list of serial numbers from certificates that are no longer valid for reasons other than that they have expired. Grasping the mechanics of PKI and certificates is not necessarily an easy process, and you may want to read through this section a few times to cement your understanding.

SSL Implementation

Windows 2000 Server family includes a Certificate Server that can be used to grant certificates to Web site operators. After the Web site operator has a digital certificate, he can implement SSL and protect the contents of communications between the Web server and Web client.

The Windows 2000 root certificate authority must be installed on a domain controller (DC) running Active Directory. Child certificate authorities can be created on member servers. In this exercise, we will install the certificate server on a member server.

1. Log on as Administrator at a member server in your domain.
2. Open the Control Panel, and then open the Add/Remove Programs applet.
3. In the Add/Remove Programs applet, click on ADD/REMOVE WINDOWS COMPONENTS on the left side of the window.
4. In the Windows Components Wizard window, place a checkmark in the Certificate Services check box. A warning dialog detailing that domain membership cannot be changed after installing certificate server will appear. Click YES.
5. Choose a Certificate Authority type. Since the certificate server is being installed on member server, it cannot be the Enterprise Root CA. Select Enterprise subordinate CA. Click NEXT.
6. Enter identifying information (such as CA name, organization, organizational unit (OU), and e-mail address) in all the fields. Click NEXT.
7. Specify the local paths for the Certificate Database and the Certificate database log. Then click NEXT. The following screens determine how the certificate request is processed. Configuration options include sending the request directly to a parent certificate authority, or saving the request to a file that can be sent later to a parent certificate authority. In this example, select the "Send the request directly to a CA already on the network" option button. Click BROWSE to select a CA to send the request to.
8. After choosing the CA, the name of the computer and the name of the parent CA appear in the request text boxes. Click NEXT. A dialog box appears, warning that Internet Information Services will be shut down if it is running on this computer. Click OK. Insert the Windows 2000 CD-ROM, or point to the location of the Windows 2000 installation files and following the onscreen instructions.
9. The wizard completes the installation of the Certificate Server and presents a dialog box informing you of this. Click FINISH to complete the installation.
10. To confirm successful installation of the certificate server, open the Certificate Server management console, which is located in Administrative Tools, and there should be a green checkmark on the certificate server's name indicating that it is functioning correctly.

The installed certificate server can now issue certificates that will enable Web sites to use SSL for secure communications.

TOPIC 8: Secure Communications over Virtual Private Networks

Remote connectivity is becoming a popular solution to a variety of problems: the need for sales personnel to access company databases while on the road, the need for traveling executives to stay in touch with the office, and the need for telecommuting employees to view and manipulate files on corporate servers. The ability to extend the reach of the corporate network to remote locations is no longer a luxury, but a necessity.

There are several ways to establish a remote connection to a private network. One option is to dial in directly over the public telephone lines, using a modem on the remote computer to connect to a modem on the company server. With security concerns on the increase, this type of basic remote access infrastructure is not always cost effective and does not stand up to lose cost scrutiny when taking into consideration the three pillars of secure communication: confidentiality, integrity, and authentication. Another possibility is to have dedicated leased lines installed from one point to another. A third, increasingly attractive solution, is to take advantage of the widespread availability of Internet connectivity to establish a Virtual Private Network (VPN), which circumvents long-distance charges, doesn't require expensive capital outlays, and can be done from virtually anywhere. In the past, a VPN was considered to be a somewhat exotic, high-tech option that required a great deal of technical expertise. With Windows 2000, setting up a VPN connection is a simple process—there is even a wizard to guide you.

Tunneling Basics

A VPN can use the public network (Internet) infrastructure, yet maintain privacy and security by encrypting and encapsulating the data being transmitted. This is often referred to as tunneling through the public network.

VPN Definitions and Terminology

To understand how a VPN works, it's important to first define the terms used in conjunction with this technology.

- **Tunneling protocols** are used to create a private pathway or tunnel through an internetwork (typically the Internet) in which data packets are encapsulated and encrypted prior to transmission to ensure privacy of the communication. Windows 2000 supports two tunneling protocols: PPTP and L2TP.
- **Data encryption** provides a method of transmitting private data over public networks in a secure form. Modern VPN technologies use both encryption and encapsulation to provide an easier-to-implement and more flexible way to transmit private data over the public network. In a Windows 2000 VPN using the Point to Point Tunneling Protocol (PPTP), encryption keys are generated by the MS-CHAP or EAP-TLS authentication process, and Microsoft Point to Point Encryption (MPPE) is used to encrypt a PPP frame.
- **Encapsulation** inserts one data structure into another. VPN technology encapsulates private data with a header that provides routing information that allows the data to travel over the Internet to the private network.

How Tunneling Works

Tunneling emulates a point-to-point connection by wrapping the datagram with a header that contains addressing information to get it across the public network to the destination private network. The data is also encrypted to further protect the privacy of the communication. The tunnel is the part of the connection in which the data is encapsulated and encrypted; this becomes the virtual private network.

Data encryption is performed between the VPN client and the VPN server; thus, the connection from the client to the Internet Service Provider (ISP) does not need to be encrypted.

IP Addressing

The VPN connection will use a valid public IP address, usually supplied by the ISP's DHCP server, to route the data. This data packet, containing internal IP addresses of the sending and destination computers, is inside the envelope of the VPN, so even if you are using private (nonregistered) IP addresses on the private network, they will never be seen on the Internet. Encryption and encapsulation protect the addresses of the computers on the private network.

Security Issues Pertaining to VPNs

The concept of using an open, public network like the vast global Internet to transfer sensitive data presents obvious security concerns. For virtual networking to be feasible for security-conscious organizations, the privacy component must be ensured. Security over a VPN connection involves encapsulation, authentication of the user, and security of the data.

Encapsulation

The encapsulation of the original data packet inside a tunneling protocol hides its headers as it travels over the internetwork, and is the first line of defense in securing the communication.

User Authentication

Windows 2000 VPN solutions use the same authentication protocols used when connecting to the network locally; authentication is performed at the destination, so the security accounts database information is not transmitted onto the public network. Windows 2000 can use the following authentication methods for VPN connections:

- **CHAP** Challenge Handshake Authentication Protocol, which uses challenge-response with one-way hashing on the response, allows the user to prove to the server that he knows the password without actually sending the password itself over the network.
- **MS-CHAP** Microsoft CHAP, which also uses a challenge-response authentication method with one-way encryption on the response.
- **MS-CHAP v2** An enhanced version of Microsoft-CHAP, which is a mutual authentication protocol requiring both the client and the server to prove their identities.
- **EAP/TLS** Extensible Authentication Protocol/Transport Level Security, which provides support for adding authentication schemes such as token cards, one-time passwords, the Kerberos V5 protocol, public key authentication using smart cards, certificates, and others.

Data Security

Data security is provided through encapsulation and encryption, but the greater the security, the more overhead and the lower the performance. IPSec was designed to work with different encryption levels and provide different levels of data security based on the organization's needs.

NOTE
PPTP uses Microsoft Point to Point Encryption (MPPE) to encrypt data. When using L2TP for VPN connections, data is encrypted using IPSec.

L2TP over IPSec uses certificate-based authentication, which is the strongest authentication type used in Windows 2000. A machine-level certificate is issued by a certificate authority, and installed on the VPN client and the VPN server. This can be done through the Windows 2000 Certificate Manager or by configuring the CA to automatically issue certificates to the computers in the Windows 2000 domain.

Windows 2000 Security Options

Windows 2000 provides the Network Administrator with a great deal of flexibility in setting authentication and data encryption requirements for VPN communications. This next table shows possible security settings combinations for both PPTP and L2TP.

Authentication and Encryption Requirement Settings

Validate My Identity Using	Require Data Encryption	Authentication Methods Negotiated	Encryption Enforcement
PPTP			
Require secure password	No	CHAP, MS-CHAP, MS-CHAP v2	Optional encryption (connect even if no password encryption)
Require secure password	Yes	MS-CHAP, MS-CHAP v2	Require encryption (disconnect if server password declines)
Smart card	No	EAP/TLS	Optional encryption (connect even if no encryption)
Smart card	Yes	EAP/TLS	Require encryption (disconnect if server declines)
L2TP			
Require secured password	No	CHAP, MS-CHAP, MS-CHAP v2	Optional (connect even if no encryption)
Require secured password	Yes	CHAP, MS-CHAP, MS-CHAP v2	Require encryption (disconnect if server declines)
Smart card	No	EAP/TLS	Optional encryption (connect even if no encryption)
Smart card	Yes	EAP/TLS	Require encryption (disconnect if server declines)

These settings are configured on the Security tab of the Properties sheet for the VPN connection. To access this dialog box, from the Start menu, select Settings | Network and Dialup Connections | [name of your VPN connection]. Then click PROPERTIES and select the Security tab. Selecting the Advanced radio button and clicking SETTINGS displays the Advanced Security Settings dialog box, where the authentication and encryption setting combinations can be adjusted.

This dialog box allows you to select whether encryption is optional, required, or not allowed; whether to use EAP or allow other designated protocols; and whether to automatically enter the logged-on account's Windows username and password for MS-CHAP authentication. If you choose to use EAP (for instance, to enable authentication via smart card), you will need to configure the properties for the smart card or other certificate authentication. You can choose from a list of recognized root certificate authorities (CAs).

NOTE
A CA is an entity entrusted to issue certificates to individuals, computers, or organizations that affirm the identity and other attributes of the certificate. VeriSign is an example of a remote third-party CA recognized as trustworthy throughout the industry.

Common VPN Implementations

VPNs are commonly used by companies to provide a more cost-effective way for employees, customers, and other authorized users to connect to their private networks. The VPN is a viable alternative to direct dial-in, which incurs long-distance charges, or the hefty initial and monthly expense of a dedicated leased line.

VPNs are typically used to allow a remote user to connect a stand-alone computer, such as a home desktop system or a laptop/notebook computer when on the road, to the corporate network. However, VPNs can also be used to connect two distant LANs to one another using their local Internet connections, or to securely connect two computers over an intranet within the company.

Remote User Access Over the Internet

A typical scenario is the traveling employee who needs to connect to the company's network from a remote location. The traditional way to do so was to dial in to the company RAS server's modem. While a workable solution, it can prove costly if the remote user is not in the company's local calling area. If the remote user has an ISP local to his location, however, he can avoid long-distance charges by dialing the ISP instead of the company's modem, and setting up a VPN through the Internet.

NOTE
An active Winsock Proxy client will interfere with the creation of a VPN by redirecting data to the proxy server before the data can be processed by the VPN. You must first disable the Winsock Proxy client before attempting to create a VPN connection.

Connecting Networks Over the Internet

Another use of the VPN is to connect two networks through the Internet. If you have offices in two cities with a LAN at each office location, it may be advantageous to connect the two LANs so that users at both locations can share one another's resources. One way to do so is to purchase a leased line such as a T1 line to connect the two networks, but this could prove to be expensive. An alternate option is to create a VPN between the two sites.

Sharing a Remote Access VPN Connection

If both offices already have Internet connections, perhaps through dedicated ISDN lines or DSL service, the existing connection to the Internet can be used to set up a VPN between the two offices.

In this case, setup will be slightly more complicated than connecting a single remote computer to a company network. In order to give all the computers on both LANs access to the resources they need, a VPN server on each side of the connection would have to be configured, as well as VPN client connections. The VPN client connection could then be shared with the rest of the LAN via Internet Connection Sharing. Another level of security can be employed by restricting the VPN client to access resources only on the VPN server and not on the rest of the network.

Using a Router-to-Router Connection

Another way to connect two networks via a VPN is to use a router-to-router VPN connection with a demand-dial interface. The VPN server then provides a routed connection to the network of which it is a part. Routing and Remote Access Service (RRAS) is used to create router-to-router VPN connections, so the VPN servers acting as routers must be Windows 2000 servers or NT 4.0 servers with RRAS.

Mutual authentication is supported, so that the calling router (VPN client) and answering router (VPN server) authenticate themselves to one another.

In a router-to-router connection, the VPN works as a Data Link layer connection between the two networks. The endpoints of a router-to-router connection are the routers, and the tunnel extends from one router to the other. This is the part of the connection in which the data is encapsulated.

Tunneling Protocols and the Basic Tunneling Requirements

Establishing a secure tunnel through a public or other internetwork requires that computers on both ends of the connection are configured to use Virtual Private Networking, and they must both be running a common tunneling protocol. Windows 2000 Server can be a VPN client, or it can be a VPN server accepting PPTP connections from both Microsoft and non-Microsoft PPTP clients.

Windows 2000 Tunneling Protocols

As mentioned earlier, Windows 2000 supports two tunneling protocols for establishing VPNs: PPTP and L2TP. A primary difference between the two is the encryption method: PPTP uses MPPE to encrypt data, while L2TP uses certificates with IPSec.

Point to Point Tunneling Protocol (PPTP)

The Point to Point Tunneling Protocol (PPTP) was developed as an extension to the popular Point to Point Protocol (PPP) used by most ISPs to establish a remote access connection to the Internet through the provider's network. PPTP allows IP, IPX, and NetBIOS/NetBEUI datagrams or frames to be transferred through the tunnel. From the user's perspective, the tunneling is transparent.

PPTP allows for Windows NT 4 authentication, using the insecure Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), and Microsoft's version of CHAP, MSCHAP. PPTP is an open standard.

Layer 2 Tunneling Protocol (L2TP)

The Layer 2 Tunneling Protocol (L2TP) provides the same functionality as PPTP, but overcomes some of the limitations of PPTP. Unlike PPTP, it does not require IP connectivity between the client workstation and the server. L2TP can be used as long as the tunnel medium provides packet-oriented point-to-point connectivity, which means it works with such media as ATM, Frame Relay, and X.25.

L2TP is an Internet Engineering Task Force (IETF) standard, which was developed in a cooperative effort by Microsoft, Cisco Systems, Ascend, 3Com, and other networking industry leaders. It combines features of Cisco's Layer 2 Forwarding (L2F) protocol with Microsoft's PPTP implementation. L2TP can use IPSec to provide end-to-end security.

Using PPTP with Windows 2000

PPTP is installed with RRAS. It is configured by default for five PPTP ports. PPTP ports can be enabled with the Routing and Remote Access wizard. The PPTP ports are displayed as WAN miniports in the RRAS console. The status of each VPN port can be displayed, refreshed, or reset by double-clicking on the port name to display the status sheet and clicking the appropriate button.

How to Configure a PPTP Device

To configure a port device, right-click on Ports in the left panel of the console and select Properties. Highlight the RRAS device you wish to configure, and then click CONFIGURE. In the device configuration dialog box, you can set up the port to be used for inbound RAS connections and/or inbound and outbound demanddial routing connections.

NOTE

A device can be physical, representing hardware (such as a modem), or virtual, representing software (such as the PPTP protocol). A device can create physical or logical point-to-point connections, and the device provides a port, or communication channel, that supports a point-to-point connection.

A standard modem is a single-port device. PPTP and L2TP are virtual multiport devices. You can set up to 1000 ports for PPTP and L2TP devices. Five is the default number of ports.

Using L2TP with Windows 2000

Layer 2 Tunneling Protocol (L2TP) over IPSec provides Administrators the facility to provide end-to-end security for a VPN connection. L2TP does not rely on vendor-specific encryption methods to create a completely secured virtual networking connection.

How to Configure L2TP

To enable the server to be a VPN server for L2TP clients, RRAS must be installed if it has not already. Open the RRAS console: Start | Programs | Administrative Tools | Routing and Remote Access. In the left pane of the console tree, right-click the server to be enabled, and click Configure and Enable Routing and Remote Access. This starts the wizard, which guides you through the process. After the service is installed and started, configure the properties of the server by right-clicking on the server name and selecting Properties.

On the General tab, be sure that the “Remote access server” check box is selected. On the Security tab, under Authentication Provider, you can confirm the credentials of RRAS clients by using either Windows 2000 security (Windows Authentication) or a RADIUS server. If RADIUS is selected, RADIUS server settings need to be configured for the RADIUS server or RADIUS proxy.

In the Accounting Provider drop-down box, choose Windows or RADIUS accounting. Accordingly, remote access client activity can be logged for analysis or accounting purposes. Next, click AUTHENTICATION METHODS, and choose the authentication methods that are supported by the RRAS server to authenticate the credentials of remote access clients.

TIP

Microsoft remote access clients generally will use MS-CHAP authentication. To enable smart card support, use EAP authentication.

On the IP tab, verify that the “Enable IP routing” and “Allow IP-based remote access and demand-dial connections” check boxes are both checked. Next, configure the L2TP ports for remote access. In the RRAS console, right-click on Ports and select Properties. Select the L2TP ports.

How L2TP Security Differs from PPTP

L2TP is similar to PPTP in many ways. They both support multiprotocol VPN links and can be used to create secure tunnels through the Internet or another public network to connect to a private network that also has a connection to the internetwork. L2TP can be used over IPSec to provide for greater security, including end-to-end encryption, whereas Microsoft's PPTP connections are dependent upon MPPE for encryption. L2TP is derived from L2F, a Cisco Systems tunneling protocol.

With L2TP over IPSec, encapsulation involves two layers: L2TP encapsulation and IPSec encapsulation. First, L2TP wraps its header and a UDP header around a PPP frame. Then IPSec wraps an ESP (Encapsulating Security Payload) header and trailer around the package, and adds an IPSec authentication trailer. Finally, an IP header is added, which contains the addresses of the source (VPN client) and destination (VPN server) computers. The data inside the IPSec ESP header and authentication trailer, including the PPP, UDP, and L2TP headers, is all encrypted by IPSec.

Data authentication is available for L2TP over IPSec connections, unlike for PPTP connections. This is accomplished by the use of a cryptographic checksum based on an encryption key known only to the sender and the receiver.

Interoperability with Non-Microsoft VPN Clients

A Windows 2000 VPN server can accept client connections from non-Microsoft clients, if the clients meet the following requirements:

- The clients must use PPTP or L2TP tunneling protocol.
- For PPTP connections, the client must support MPPE.
- For L2TP connections, the client must support IPSec.

If these requirements are met, the non-Microsoft clients should be able to make a secure VPN connection. No special configuration changes on the VPN server are required to allow non-Microsoft clients to connect.

TOPIC 9: IPSec for Windows 2000

IPSec defines a network security architecture that allows secure networking for the enterprise while introducing a minimum of overhead. By performing its services at the Network layer, IPSec secures information in a manner that is transparent to the user and to the protocols that lie above the Transport layer. IPSec provides Layer 3 protection.

The IPSec security architecture exercises an end-to-end security model. Only the endpoints of a communication need to be IPSec aware. Computers and devices that serve as intermediaries of message transfer do not need to be IPSec enabled. This allows the Administrator of a Windows 2000 network to implement IPSec for end-to-end security over diverse network infrastructures, including the Internet. Transit network devices such as bridges, switches, and routers can be oblivious to IPSec without compromising its efficacy.

NOTE

IPSec provides protection of the data transmission from end to end. This is different from the PPTP model that only protects the link.

This end-to-end capability can be extended to different communication scenarios, including:

- Client to client
- Gateway to gateway

When IPSec is used to protect communications between two clients—for example, on the same LAN—the machines can use IPSec in what is known as transport mode. In transport mode, both clients must use TCP/IP as their network protocol. In this example, the endpoints of the secure communication are the source machine and the destination host.

In contrast, with a gateway-to-gateway solution, information traversing a transit network (such as the Internet) is protected by IPSec. Packets are protected as they leave the exit gateway and then decrypted or authenticated at the destination network's gateway. In this scenario, the host and destination computers do not employ IPSec, and can use any LAN protocol supported by IPSec (IPX/SPX, AppleTalk, NetBEUI, TCP/IP).

When gateways represent the endpoints of secure communication, IPSec works in tunnel mode. A tunnel is created between the gateways, and client-to-client communications are encapsulated in the tunnel protocol headers. Tunnels can be created using IPSec as the tunneling protocol, or you can combine IPSec with L2TP, which stands for Layer 2 Tunneling Protocol and allows for data encryption via IPSec. In this case, L2TP rather than IPSec creates the tunnel.

Overview of IPSec Cryptographic Services

IPSec ensures secure communications by providing robust solutions that support confidentiality, integrity, and authenticity (CIA). It is a worthwhile exercise to revisit the aspects of CIA and understand how they apply to IPSec.

Message Integrity

Message integrity implies that the contents of a message have not changed during transit. Creating a digital signature can protect message integrity, acting almost as a digital fingerprint. This fingerprint represents the contents of the message. If someone were to capture and change the contents of the message, the fingerprint would change. The destination host could detect the fraudulent fingerprint and would be aware that "other hands" had touched the document. The assumption is that if other hands have touched the document, then the message is invalid. It has lost its integrity. Hash algorithms create these fingerprints.

Hashing Messages

The result of a hash is a fixed-length string known as a message digest. The message digest represents the hashed output of a message. Microsoft's implementation of IPSec uses one of two algorithms for hashing:

- ✿ **Message Digest 5 (MD5)** processes each message in blocks of 512 bits. The message digest ends up being 128 bits.
- ✿ **Secure Hash Algorithm (SHA-1)** processes messages in blocks of 512 bits. However, the resulting message digest is 160 bits long. This makes the message more secure. It is more processor intensive,

and therefore slower than MD5.

Each partner in the communication must use the same key in order to come up with the same hashed result. Though we have already discussed the use of public key infrastructure and key exchange, we will touch on these topics again.

Message Authentication

When a host is authenticated, its identity is confirmed. While integrity is concerned with the validity of the contents of a message, authentication is aimed at confirming the validity of the sender. IPSec can use any of the following methods to authenticate the sender:

- Preshared key authentication
- Kerberos authentication
- Public key certificate-based digital signatures

Preshared Key Authentication

Preshared key authentication schemes depend on both members of the communication having preselected a secret key that will be used to identify them to each other. Data leaving the sending computer is encrypted with this agreed-to key, and is decrypted on the other end with the same key.

You can use the preshared key to authenticate a computer using the following procedure:

1. The sending computer hashes a piece of data (a challenge) using the shared key and forwards this to the destination computer.
2. The destination computer receives the challenge, performs a hash using the same secret key, and sends it back.
3. If the hashed results are identical, both computers share the same secret and are thus authenticated.

Preshared keys are effective and simple to implement. They circumvent potential complications introduced when other authentication schemes are used. However, the shared-key approach is not very scaleable or mutable. The shared key must be manually entered into every extant IPSec policy.

An organization with a large number of organizational units, all using different IPSec policies, would find it difficult to track all the keys. In addition, the keys should change frequently. Manually changing the keys can be an arduous process within large organizations.

Kerberos Authentication

The Kerberos authentication method is also based on the shared secret principle. In this case, the shared secret is a hash of the user's password.

Public Key Certificate-Based Digital Signatures

When a private key encrypts a hash, the message digest forms a digital signature. A message is authenticated after it is decrypted with the source's public key and then run through the hash algorithm. In a public key infrastructure, each computer has a public and a private key. The public key is open and available to the public at large; it is not secret. The private key is a secret key that is only available to the owner of the private key. The private key must remain private. If the private key is ever compromised, all messages from the owner of that private key should be considered suspect.

A viable public key infrastructure includes elements:

- Secret private keys
- Freely available public keys
- A trusted third party to confirm the authenticity of the public key

The trusted third party will digitally sign each party's public key. This prevents people from providing a public key that they claim is theirs, but is in fact not the public key of the person they are impersonating. Public key authentication is used when non-Kerberos-enabled clients need to be authenticated and no preshared key has been established. You must also use public key authentication when using L2TP tunneling and IPSec.

Confidentiality

Neither integrity nor authentication is concerned with protecting the privacy of information. In order to ensure confidentiality, data is encrypted using algorithms such as the Data Encryption Standard (DES) or the Cipher Block Chaining (CBC).

DES is a symmetric encryption algorithm. DES works on 64-bit blocks of data. The DES algorithm converts 64 input bits from the original data into 64 encrypted output bits. While DES starts with 64-bit keys, only 56 bits are used in the encryption process. The remaining 8 bits are for parity.

CBC prevents each DES block from being identical. This DES-CBC algorithm makes each ciphertext message appear different.

IPSec Security Services

IPSec engages two protocols to implement security on an IP network:

- Authentication header (AH)
- Encapsulating security protocol (ESP)

Authentication Header (AH)

The authentication header ensures data integrity and authentication. The AH does not encrypt data, and therefore provides no confidentiality. When the AH protocol is applied in transport mode, the authentication header is inserted between the original IP header and the TCP header. The entire datagram is authenticated using AH.

Encapsulating Security Payload (ESP)

The encapsulating security payload protocol can provide authentication, integrity, and confidentiality to an IP datagram. Authentication services are available with ESP, but the original IP header prior to application of the ESP header is not authenticated. The ESP header, in transport mode, is placed between the original header and the TCP header. Only the TCP header, data, and ESP trailer are encrypted. If authentication of the original IP header is required, you can combine and use AH and ESP together. AH and ESP can be applied at a gateway machine connecting the LAN to a remote network. In this case, tunnel mode would be used. In tunnel mode, an additional IP header is added that denotes the destination tunnel endpoint. This tunnel header encapsulates the original IP header, which contains the IP address of the destination computer.

TOPIC 10: Security Associations and IPSec Key Management Procedures

When two computers establish a connection using IPSec, they must come to an agreement regarding which algorithms and protocols they will use. A single security association (SA) is established for each link a computer maintains with another computer via IPSec. If a file server has several simultaneous sessions with multiple clients, a number of different SAs will be defined, one for each connection via IPSec. Each security association has associated with it these parameters:

- An encryption algorithm (DES or 3DES)
- A session key (via Internet Key Exchange, or IKE)
- An authentication algorithm (SHA1 or MD5)

A security parameters index (SPI) tracks each SA. The SPI uniquely identifies each SA as separate and distinct from any other IPSec connections current on a particular machine. The index itself is derived from the destination host's IP address and a randomly assigned number. When a computer communicates with another computer via IPSec, it checks its database for an applicable SA. It then applies the appropriate algorithms, protocols, and keys, and inserts the SPI into the IPSec header.

An SA is established for outgoing and incoming messages, necessitating at least two security associations for each IPSec connection. In addition, a single SA can be applied to either AH or ESP, but not both. If both are used, then two more security associations are created. One SA for inbound and one SA for outbound communications will be created.

IPSec Key Management

Keys must be exchanged between computers in order to ensure authenticity, integrity, and confidentiality. Key management defines the procedure of how the keys are formed, the strength of the keys, how often they are changed, and when they expire. The establishment of a shared secret key is critical to secure communications. The shared secret can be manually established using the prearranged key method, but this technique does not scale very well because of its inherent lack of flexibility.

Automated key management is the preferred method of key exchange. Automated key management uses a combination of the Internet Security Association Key Management Protocol and the Oakley Protocol (ISAKMP/Oakley). This combination of protocols is often referred to collectively as the Internet Key Exchange (IKE). The IKE is responsible for exchange of key material (groups of numbers that will form the basis of new key) session keys, SA negotiation, and authentication of peers participating in an IPSec interaction.

The IKE takes place across two phases: Phase 1, in which the two computers agree upon mechanisms to establish a secure, authenticated channel, and Phase 2, where Security Associations are negotiated for security protocols; either AH, ESP, or both.

The first phase establishes what is called the ISAKMP security association (ISAKMP SA), and the second phase establishes the IPSec SA.

Phase 1: Establishing the ISAKMP SA

The following points detail the sequence of events during the ISAKMP SA:

1. The computers establish a common encryption algorithm, either DES or 3DES.
2. A common hash algorithm is agreed upon, either MD5 or SHA1.

3. An authentication method is established. Depending on policy, this can be Kerberos, public key encryption, or prearranged shared secret.
4. A Diffie-Hellman group is agreed upon in order to allow the Oakley protocol to manage the key exchange process. Diffie-Hellman provides a mechanism for two parties to agree on a shared master key, which is used immediately or can provide keying material for subsequent session key generation. Oakley will determine key refresh and regeneration parameters.

Phase 2: Establishing the IPSec SA

After a secure channel has been established by the creation of the ISAKMP SA, the IPSec SAs will be established. The process is similar, except that a separate IPSec SA is created for each protocol (AH or ESP) and for each direction (inbound and outbound). Each IPSec SA must establish its own encryption algorithm, hash algorithm, and authentication method.

One important difference is that each IPSec SA uses a different shared key than that negotiated during the ISAKMP SA. Depending on how policy is configured, the IPSec SA repeats the Diffie-Hellman exchange, or reuses key material derived from the original ISAKMP SA. All data transferred between the two computers will take place in the context of the IPSec SA.

TOPIC 11: Deploying IPSec

In the implementation of IPSec in an organization, planning takes on special importance in the design of a security infrastructure. The planning phase is followed by the implementation phase. Windows 2000's graphical interface makes it easy to develop an IPSec policy for any organization. IPSec policy, filters, filter actions, and interoperability with downlevel clients and other operating systems are a vital part of implementation.

Building Security Policies with Customized IPSec Consoles

IPSec configuration and deployment is intimately intertwined with Active Directory and group policy. You must create a policy in order to deploy IPSec in the organization. A policy can be applied to a site, a domain, an organizational unit, or a single computer. It is within the group policy that we can choose from built-in policies or create custom policies to meet our specialized needs. These policies can be configured by creating an MMC and then using the appropriate MMC plug-in. It is possible to configure a custom IPSec console that is used to configure IPSec policy and monitor significant IPSec-related events.

Building an IPSec MMC Console

1. Create a new console by starting the Run command and typing **mmc**. Click OK to open an empty console.
2. Click the Console menu, and then click Add/Remove Snap-in. Click ADD, select Computer Management, and click ADD. A dialog box will appear that will want to know which computer the snap-in will manage. Select Local Computer (the computer this console is running on). Then click FINISH.
3. Scroll through the list of available snap-ins, select Group Policy, and click ADD. At this point, a wizard will appear that will query you on what Group Policy Object you want to manage. In this case, confirm that it says Local Computer in the text box, and click FINISH. If you want to define a policy for another group policy object, click BROWSE and select from the list.
4. Scroll through the list of Group Policy Objects again, this time looking for Certificates. Select Certificates, and click ADD. A dialog box will appear asking you what you want the snap-in to always manage certificates for. Select Computer Account, click NEXT, and then select Local Computer for the computer that you want the snap-in to manage. Then click FINISH.
5. Click CLOSE on the Add Standalone Snap-in dialog box, and then click OK in the Add/Remove Snap-in dialog box. Expand the first level of each of the snap-ins.

IPSec policies can be configured and managed from this custom console. In this example, IPSec policy is managed for a single machine. This might be appropriate when configuring IPSec policy for a file or application server. If you wanted to manage policy for an entire domain or organizational unit, you would select the appropriate policy when selecting the Group Policy snap-in configuration.

Flexible Security Policies

Now that we have our console, we can get to the business of building IPSec security policy. Three built-in IPSec policies can be used, and custom policies can be created.

To begin, you need to find where the IP security policies are located. Expand the Local Computer policy, expand the Computer Configuration object, expand the Windows Settings object, then click IP Security Policies on Local Machine. In the right pane, you will see listed the three built-in IPSec policies: Client (Respond Only), Secure Server (Require Security), and

Server (Request Security). The Client (Respond Only) policy is used when secure IPSec connections are required once another computer requests them. For example, a workstation requires connectivity to a file server that requires IPSec security. The workstation with the built-in Client policy enabled negotiates an IPSec security association. However, this client never requires IPSec security; it will only use IPSec to secure communications when requested to do so by another computer.

The Server (Request Security) policy is used when IPSec security is requested for all connections. This could be used for a file server that must serve both IPSec-aware (Windows 2000) clients and non-IPSec-aware clients (such as Windows 9*x* and Windows NT). If a connection is established with an IPSec-aware computer, the session will be secure. Unsecured sessions will be established with non-IPSec-aware computers. This allows greater flexibility during the transition from mixed Windows networks to native Windows 2000 networks. The Secure Server (Require Security) policy is used when all communications with a particular server need to be secured. Examples include file servers storing sensitive data and security gateways at either end of an L2TP/IPSec tunnel. The server with the Secure Server policy will always request a secure channel. Connections will be denied to computers not able to respond to the request.

Security policies are bidirectional. If a Secure Server attempts to connect to non-IPSec-aware network servers such as DNS, WINS, or DHCP servers, the connection will fail. It is imperative that all scenarios are tested in a lab that simulates a live network before implementing IPSec policies. During the testing phase, it is important to assiduously check the event logs to ascertain what services fail because of IPSec policies.

Rules

An IPSec policy has three main components: IP security rules, IP filter lists, and IP filter actions. Double-click the Server Policy to see the Server (Request Security) Properties sheet. Rules are applied to computers that match criteria specified in a filter list. An IP filter list contains source and destination IP addresses. These can be individual host IP addresses or network IDs. When a communication is identified as a participant included in an IP filter list, a particular filter action will be applied that is specific for that connection. The All IP Traffic filter list includes all computers that communicate with the server via TCP/IP. Any instructions in the filter action associated with All IP Traffic will be applied to all computers.

First, double-click All IP Traffic filter list. This opens the Edit Rule Properties dialog box for the All IP Traffic filter. You should see a tabbed dialog box consisting of five tabs. The option button for the IP filter list is selected, and a description is included which explains the purpose of the list. Double-click All IP Traffic filter list to see the details of the All IP traffic filter. The Name, Description, and the details of the filter are displayed in the details.

If you want to see more details regarding the Addressing, Protocol, and Description of the filter, you can click EDIT. Click CANCEL twice to return to the Edit Rules Properties dialog box.

Filter Actions

Filter Actions define the type of security and the methods by which security is established. The primary methods are Permit, Block, and Negotiate security. The Permit option blocks negotiation for IP security. This is appropriate if you never want to secure traffic to which this rule applies. The Block action blocks all traffic from computers specified in the IP filter list. The Negotiate security action allows the computer to use a list of security methods to determine security levels for the communication. The list is in descending order of preference. If the Negotiate security action is selected, both computers must be able to come to an agreement regarding the security parameters included in the list. The entries are processed sequentially in order of preference. The first common security method is enacted.

Click the Filter Action tab, and click Request Security (Optional) to view these options. Of the check boxes at the bottom of the dialog box, “Accept unsecured communication, but always respond using IPSec," allows unsecured communication initiated by another computer, but requires the computers to which this policy applies to always use secure communication when replying or initiating. This is essentially the definition of the Secure policy. The "Allow unsecured communication with non-IPSec-aware computer" option allows unsecured communications to or from another computer. This is appropriate if the computers listed in the IP filter lists are not IPSec-enabled. However, if negotiations for security fail, this will disable IPSec for all communications to which this rule applies. Perhaps the most important of these options is the session key Perfect Forward Secrecy. When you select this option, you ensure that session keys (or keying material) are not reused, and new Diffie-Hellman exchanges will take place after the session key lifetimes have expired. Click CANCEL to return to the Edit Rule Properties dialog box. Click the Authentication Methods tab. Here you can select your preferred authentication method. Kerberos is the default authentication method. You can include other methods in the list, and each will be processed in descending order. You can click ADD to include additional authentication methods.

Click the Tunnel Setting tab if the endpoint for the filter is a tunnel endpoint. Click the Connection Type tab to apply the rule to All network connections, Local area network (LAN), or Remote access. You cannot delete the built-in policies, but you can edit them. However, it is recommended that you leave the built-in policies as they are, and create new policies for custom requirements.

Flexible Negotiation Policies

Security method negotiation is required to establish an IPSec connection. The default policies can be used, or custom policies can be created. To add a new filter action, which will be used to create a new security policy, click ADD after selecting the Filter Action tab. When the wizard has completed, you can edit the security negotiation method.

When you double-click on the Request Security (Optional) filter action, you will see the Request Security (Optional) Properties dialog box. If you select the Negotiate security option, and then click ADD, you can add a new security method.

You may fine-tune your security negotiation method by selecting the Custom option, and then clicking SETTINGS. After doing so, you will see the Custom Security Method Settings dialog box. Here you can configure whether you want to use AH, ESP, or both. For each option, you can select the integrity algorithm, encryption algorithm, or both. All algorithms supported in Windows 2000 are included. Session key lifetimes can be customized by entering new key generation intervals by amount of data transferred or time span.

Filters

Rules are applied to source and destination computers or networks, based on their IP addresses. To create a new filter, you can avail yourself of the New Filter Wizard. To do this, return to the Edit Rule Properties dialog box, click on the IP Filter List tab, and then click ADD. This brings up the IP Filter List dialog box, where you enter in the name of the new filter and a description of the filter. Click ADD to start the wizard. When the wizard starts, you see the Welcome dialog box. Click NEXT.

Choose the source address of the wizard. Your options appear after you click the down arrow on the list box. Note that you can identify the source by individual IP address, all IP addresses, DNS name, or subnet. Click NEXT to continue.

The next dialog box asks for the destination IP address. You are afforded the same options as when you designated the source. Click NEXT to continue through the wizard. At this point, you can select which protocols will be included in the filter. All protocols are included by default, but you can select from a list of protocols or define your own by selecting Other and entering a protocol number.

Click NEXT, and then click FINISH. Your new filter will appear in the IP filter list included in the IP Filter List tab of the Edit Rule Properties dialog box.

Creating a Security Policy

Consider the following scenario: You are the Administrator of the network for a large hospital. The network is subdivided into multiple subnets. The Medical Records department contains a large amount of data that must be kept secure. The hospital would suffer a large amount of liability if security were breached. Computers within the Medical Records department are closely monitored, and therefore the overhead of confidentiality is not required, but authentication and integrity should be applied to intradepartmental communications.

The Medical Records department must regularly send information to the hospital floor. The network infrastructure is more open to attack between the well-guarded Medical Records department and the less secure, open hospital environment. All computers within the Medical Records department are located in network ID 192.168.1.0, and all floor computers that access medical records database information are located on network ID 192.168.2.0. The default Class C subnet mask is used. In order to implement your new security policy, you need to:

1. Create a security policy for the hospital's domain. In this way, all computers in the domain will inherit the IPSec policy.
2. Computers in the Medical Records department need to communicate with two sets of computers: machines within their own department, and the machines on the hospital floor. Characterizing these machines by subnet, you could say that machines on subnet 192.168.2.0 need to communicate with machines on 192.168.1.0, and machines on 192.168.1.0 need to communicate with machines on 192.168.2.0. When selecting the protocols, you would select All so that all IP traffic is filtered. Therefore, you need to create two filters, so that you can assign different filter actions to each filter.
3. Now you need to create two filter actions (Negotiation policy); the first filter action will be applied to intradepartmental communications, in which just authentication and integrity are important, and the second filter action will be applied to extradepartmental communication, where authenticity, integrity, and confidentiality are required. The first filter action might use AH, which provides for authenticity and integrity. The second filter action might use a combination of AH and ESP, to provide the highest level of authentication and integrity, while also providing confidentiality.

By implementing these combinations of filters and filter rules, you can effectively secure traffic in a customized fashion. You can easily implement this solution by invoking the Security Rule Wizard after you create the new security policy.

Making the Rule

The rule will create a filter for all communications emanating from 192.168.1.0 that are directed to 192.168.2.0. After the filter is created, you will create a filter action. In this case, you need to ensure secure communications, because you are communicating with the unsecured hospital floor. You need to insure integrity, authentication, and confidentiality.

1. Click Start | Programs | Administrative Tools | Active Directory Users and Computers. When the console opens, right-click on a domain name, then click Properties. In the Domain properties dialog box, click on the Group Policy tab.
2. Select Default Domain Policy, and click EDIT.

3. This opens the Group Policy Editor. Expand Computer Configuration, expand Windows Settings, expand Security Settings, and then right-click on IP Security Policies on Active Directory. Click Create IP Security Policy.
4. A wizard starts up, welcoming you. Click NEXT.
5. You now need to enter the name of the Policy. Name it **MedRecToFloor**, and then click NEXT. Remove the checkmark in the "Activate the default response rule" check box. Click NEXT.
6. Now you are at the end of the Wizard. Leave the check in the Edit Properties box, and click FINISH.
7. At this point, you have no IP filter lists. Use the Add wizard to create a new filter list and filter action. Together they create a filter rule. Make sure there is a check in the "Use Add Wizard" check box, and click ADD.
8. This takes you to the Security Rule Wizard. The first dialog box is a Welcome box. Click NEXT.
9. The next dialog box asks whether the rule applies to a tunnel endpoint. In this case, it does not, so select "This rule does not specify a tunnel." Click NEXT.
10. The wizard now asks what network connections this rule should apply to. Select "All network connections," then click NEXT.
11. Now decide what the default authentication protocol should be used. Select Windows 2000 default (Kerberos V5 protocol). Then click NEXT.
12. Create the IP filter list by adding a filter for all traffic sent from 192.168.1.0 with the destination of 192.168.2.0. Click ADD.
13. You now see the IP Filter List dialog box. Type **Secure from MedRec to Floor**, and make sure the Use Add Wizard check box is filled. Now click ADD.
14. The IP Filter Wizard (yes, another wizard!) appears. Click NEXT to move past the Welcome dialog box. Now you are at the IP Traffic Source dialog box. Click the down arrow under Source address and select A specific IP Subnet. Type **192.168.1.0** and a subnet mask of **255.255.255.0**. Then click NEXT.
15. Now enter the IP traffic destination. Under the Destination address, click the down arrow and select A specific IP Subnet. Then type the destination subnet **192.168.2.0** with a subnet mask of **255.255.255.0**. Click NEXT.
16. You want all the protocols to be included in the filter, so select Any for the protocol type and click NEXT, and then click FINISH to complete the wizard.
17. This takes you back to the IP Filter List dialog box. Click EDIT. Mirrored should checked. Match packets with the exact opposite source and destination addresses, to ensure that machines from the destination subnet are also included in the incoming filter. Click OK to close the dialog box, and then click CLOSE. You are now back to the IP Filter List dialog box in the Security Rule Wizard. Select the Secure from MedRec to Floor filter list, and then click NEXT.
18. At this point, configure a filter action. Select the Require Security option. Make sure there is a check mark in the Use Add Wizard check box, and then click ADD.
19. The IP Security Filter Action Wizard starts. Click NEXT to move past the Welcome dialog box. Here you are asked for a name; enter **SecureMedRec**, and click NEXT.
20. The Filter Action General Options dialog box asks for a filter action behavior. Select Negotiate security, and click NEXT.
21. This dialog box asks whether you want to support communications with computers that do not support IPSec. Select the "Do not communicate with computers that do not support IPSec" option. Click NEXT.

22. Now select the security method for IP traffic. To ensure confidentiality, authentication, and integrity, select Custom, and then click SETTINGS. Select the "Data and address integrity with encryption" check box, and then click the down arrow and select SHA1. Make sure there is a checkmark in the "Data integrity and encryption (ESP)" check box, and select MD5 and 3DES. Do not set the session key settings; you will select Perfect Forward Secrecy later. Click OK, then click NEXT. The final dialog box appears. Ensure that a check is in the Edit box, and then click FINISH.
23. You are brought to the New Filter Action Properties dialog box. Check "Session key Perfect Forward Secrecy." Click OK to return to the Security Rule Wizard, then click NEXT.
24. This is the last dialog box for the Security Rule Wizard. Click FINISH. Click OK to close the New Rule Properties dialog box. You are returned to the MedRecToFloor Properties box. Click the General tab. You can configure how often the Policy Agent checks for policy changes here. Click ADVANCED to control the Internet Key Exchange Process.
25. Here you control the security of the Internet Key Exchange process. Click METHODS to configure the security methods that are used to protect identities during the Key Exchange process.
26. Click OK, click OK, and then click CLOSE. Your new security policy appears in the console.

As you can see, what looks easy on paper can be somewhat daunting when you actually apply the principles! With the rule you created, all traffic leaving 192.168.1.0 to 192.168.2.0 will be secured according to the filter rule you set up. Because it is mirrored, the same rule applies in the other direction.