



**Postgraduate Diploma
in
Strategic Business Information Technology**

**Module 4
Computer Networking and Management**

**Chapter 8
Network Management**

© NCC Education Limited, 2003

Modification History

Revision	Date	Revision Description
V1.0	January 2003	For issue

© NCC Education Limited, 2003

All Rights Reserved

The copyright in this document is vested in NCC Education Limited. The document must not be reproduced by any means, in whole or in part, or used for manufacturing purposes, except with the prior written permission of NCC Education Limited and then only on condition that this notice is included in any such reproduction.

Information contained in this document is believed to be accurate at the time of publication, but no liability whatsoever can be accepted by NCC Education Limited arising out of any use made of this information.

Trademarks

NCC Education acknowledge that the trademarks and registered trademarks of products mentioned in this material are held by the companies producing them. Use of a term in this material should not be regarded as affecting the validity of any trademark or service mark.

Copyright of any screen captures in this material are the property of the software's manufacturer.

This material may contain some clipart, which is copyright to the Corel Corporation.

Contents

Introduction – Network Management	5
What Is Network Management?	5
Network Management Standards.....	8
The Internet Network-Management Framework.....	8
Structure of Management Information (SMI)	10
SMI Base Data Types	11
SNMP Security and Administration	12
Encryption	13
Authentication	13
Protection against Playback.....	14
Access Control.....	14
Principles in Practice	14
Firewalls	15
Packet Filtering.....	16
Application Gateways.....	17
Limitations of Firewalls and Gateways	19
Case History	19
The Limitations of Firewalls	19
Summary.....	20

Introduction – Network Management

Visual 1 (NCC Course Title Visual)

Visual 2

[Network management](#)

Chapter goals:

- Introduction to network management
 - motivation
 - major components
- Internet network management framework
 - MIB: management information base
 - SMI: data definition language
 - SNMP: protocol for network management
 - security and administration
- Presentation services: ASN.1
- Firewalls

In this final chapter we provide a brief introduction into network management and firewalls. These tools are for monitoring, testing, polling, configuring, analyzing, evaluating, and controlling the operation of a network. The five key components of a network management architecture are:

- (1) a network manager;
- (2) a set of managed remote devices;
- (3) management information bases (MIBs);
- (4) remote agents that report MIB information and take action under the control of the network manager;
- (5) a protocol for communicating between the network manager and the remote devices.


What Is Network Management?

Having made our way through the first seven chapters of this text, we are now well aware that a network consists of many complex, interacting pieces of hardware and software – from the links, bridges, routers, hosts, and other devices that comprise the physical components of the network to the many protocols (in both hardware and software) that control and coordinate these devices.

Visual 3

What is network management?

- ❑ **Autonomous systems (also known as 'network')**: 100s or 1000s of interacting hw/sw components
- ❑ Other complex systems requiring monitoring, control:
 - jet airplane
 - nuclear power plant
 - others?



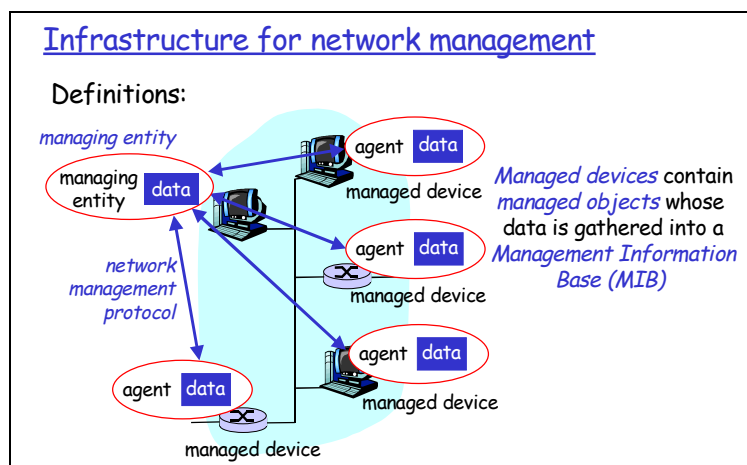
"Network management includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost."

When hundreds or thousands of such components are cobbled together by an organization to form a network, it is not surprising that:

- components will occasionally malfunction;
- network elements will be misconfigured;
- network resources will be over utilized; or
- network components will simply 'break' (for example, a cable will be cut, a can of soda will be spilled on top of a router).

The network administrator, whose job it is to keep the network 'up and running,' must be able to respond to (and better yet, avoid) such mishaps. With potentially thousands of network components spread out over a wide area, the network administrator in a Network Operations Centre (NOC) clearly needs tools to help monitor, manage, and control the network.

Visual 4



We have seen in the previous section that network management requires the ability to 'monitor, test, poll, configure, and control' the hardware, software and components in a network. Because the network devices are distributed, this will minimally require that the

network administrator be able to gather data (for example, for monitoring purposes) from a remote entity and be able to affect changes (for example, control) at that remote entity. A human analogy will prove useful here for understanding the infrastructure needed for network management.

Imagine that you are the head of a large organization that has branch offices around the world. It is your job to make sure that the pieces of your organization are operating smoothly. How would you do so? At a minimum, you:

- will periodically gather data from your branch offices in the form of reports and various quantitative measures of activity, productivity, and budget;
- will occasionally (but not always) be explicitly notified when there is a problem in one of the branch offices; the branch manager who wants to climb the corporate ladder (perhaps to get your job) may send you unsolicited reports indicating how smoothly things are running at his/her branch;
- will sift through the reports you receive, hoping to find smooth operations everywhere, but no doubt finding problems in need of your attention;
- might initiate a one-on-one dialogue with one of your problem branch offices, gather more data in order to understand the problem, and then pass down an executive order (“Make this change!”) to the branch office manager.

Implicit in this very common human scenario is an infrastructure for controlling the organization:

- the boss (you);
- the remote sites being controlled (the branch offices);
- your remote agents (the branch office managers);
- communication protocols (for transmitting standard reports and data, and for one-on-one dialogues);
- data (the report contents and the quantitative measures of activity, productivity, and budget).

Each of these components in human organizational management has a counterpart in network management.

The architecture of a network management system is conceptually identical to this simple human organizational analogy. The network management field has its own specific terminology for the various components of a network management architecture, and so we adopt that terminology here. There are three principle components of network management architecture:

- a managing entity (the boss in our above analogy – you);
- the managed devices (the branch office);
- a network management protocol.

Network Management Standards

Visual 5

Network management standards

<p>OSI CMIP</p> <ul style="list-style-type: none"> ❑ Common Management Information Protocol ❑ Designed 1980s: <i>the</i> unifying net management standard ❑ Too slowly standardized 	<p>SNMP: Simple Network Management Protocol</p> <ul style="list-style-type: none"> ❑ Internet roots (<i>SGMP</i>) ❑ Started simple ❑ Deployed, adopted rapidly ❑ growth: size, complexity ❑ Currently: SNMP V3 ❑ <i>De facto</i> network management standard
---	---

Network management standards began maturing in the late 1980s, with OSI CMISE/CMIP (the Common Management Service Element/Common Management Information Protocol) and the Internet SNMP (Simple Network-Management Protocol) (RFC 2570) emerging as the two most important standards.

Both are designed to be independent of vendor-specific products or networks. Because SNMP was quickly designed and deployed at a time when the need for network management was becoming painfully clear, SNMP found widespread use and acceptance. Today, SNMP has emerged as the most widely used and deployed network management framework. We cover SNMP in detail in the following section.

Visual 6

SNMP overview: 4 key parts

- ❑ **Management information base (MIB):**
 - distributed information store of network management data
- ❑ **Structure of Management Information (SMI):**
 - data definition language for MIB objects
- ❑ **SNMP protocol**
 - convey manager<->managed object info, commands
- ❑ **Security, administration capabilities**
 - major addition in SNMPv3

The Internet Network-Management Framework

Contrary to what the name SNMP (Simple Network-Management Protocol) might suggest, network management in the Internet is much more than just a protocol for moving management data between a management entity and its agents, and has grown to be much

more complex than the word ‘simple’ might suggest. The current Internet Standard Management Framework traces its roots back to the Simple Gateway Monitoring Protocol, SGMP [RFC 1028]. SGMP was designed by a group of university network researchers, users, and managers, whose experience with SGMP allowed them to design, implement, and deploy SNMP in just a few months. Since then, SNMP has evolved from SNMPv1 through SNMPv2 to the most recent version, SNMPv3 [RFC 2570], released in April 1999.

When describing any framework for network management, certain questions must inevitably be addressed:

- What (from a semantic viewpoint) is being monitored? And what form of control can be exercised by the network administrator?
- What is the specific form of the information that will be reported and/or exchanged?
- What is the communication protocol for exchanging this information?

Recall the human organizational analogy from the previous section:

- The boss and the branch managers will need to agree on the measures of activity, productivity, and budget used to report the branch office’s status. Similarly, they will need to agree on the actions the boss can take (for example, cut the budget, order the branch manager to change some aspect of the office’s operation, or fire the staff and shut down the branch office).
- At a lower level of detail, they will need to agree on the form in which this data is reported. For example, in what currency (dollars, euros?) will the budget be reported? In what units will productivity be measured? While these are trivial details, they must be agreed upon, nonetheless.
- Finally, the manner in which information is conveyed between the main office and the branch offices (that is, their communication protocol) must be specified.

The Internet Network-Management Framework addresses the questions posed above. The framework consists of four parts:

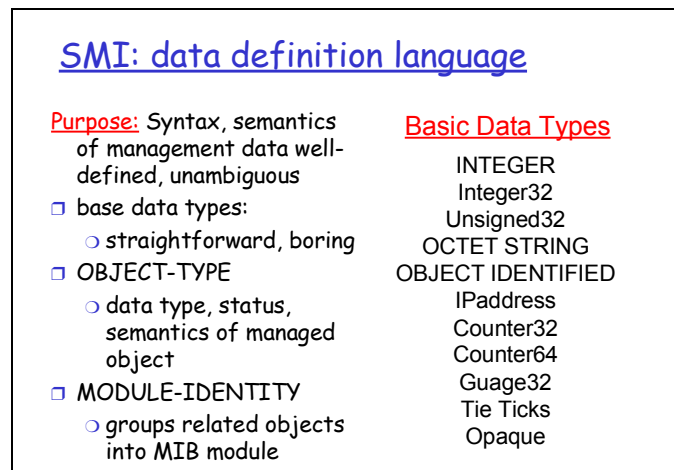
- **Definitions of network-management objects known as MIB objects** – in the Internet network-management framework, management information is represented as a collection of managed objects that together form a virtual information store, known as the Management Information Base (MIB). An MIB object might be a counter, such as the number of IP datagrams discarded at a router due to errors in an IP datagram header; or the number of carrier sense errors in an Ethernet interface card; descriptive information such as the version of the software running on a DNS server; status information such as whether a particular device is functioning correctly or not; or protocol-specific information such as a routing path to a destination. MIB objects thus define the management information maintained by a managed node. Related MIB objects are gathered into so-called MIB modules. In our human organization analogy, the MIB defines the information conveyed between the branch office and the main office.

- **A data definition language** – known as SMI (Structure of Management Information) that defines the data types, an object model, and rules for writing and revising management information. MIB objects are specified in this data definition language. In our human organizational analogy, the SMI is used to define the details of the format of the information to be exchanged.
- **A protocol, SNMP** – for conveying information and commands between a managing entity and an agent executing on behalf of that entity within a managed network device.
- **Security and administration capabilities** – the addition of these capabilities represents the major enhancement in SNMPv3 over SNMPv2.

The Internet network-management architecture is thus modular by design, with a protocol-independent data-definition language and MIB, and an MIB-independent protocol. Interestingly, this modular architecture was first put in place to ease the transition from an SNMP-based network management to a network-management framework being developed by the International Organization for Standardization (ISO), the competing network-management architecture when SNMP was first conceived – a transition that never occurred. Over time, however, SNMP’s design modularity has allowed it to evolve through three major revisions, with each of the four major parts of SNMP discussed above evolving independently. Clearly, the right decision about modularity was made, if even for the wrong reason!

Structure of Management Information (SMI)

Visual 7



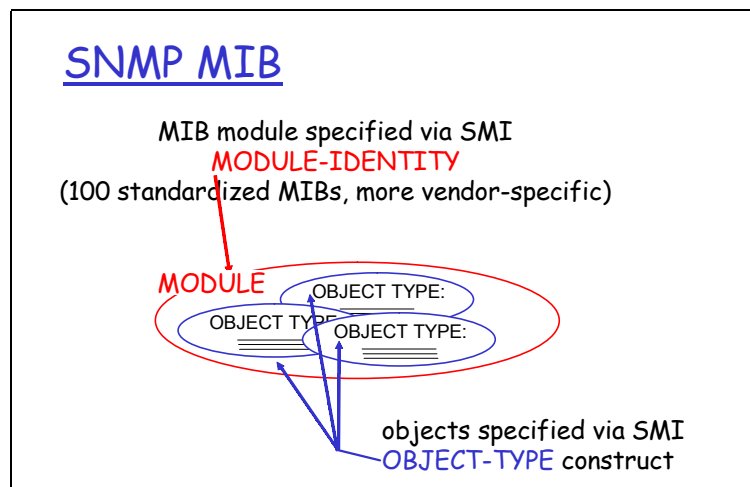
SMI is the language used to define the management information residing in a managed-network entity. Such a definition language is needed to ensure that the syntax and semantics of the network-management data are well-defined and unambiguous. Note that the SMI does not define a specific instance of the data in a managed-network entity, but rather the language in which such information is specified. The documents describing the SMI for SNMPv3 (which rather confusingly, is called SMIV2) are [RFC 2578; RFC 2579; RFC 2580].

SMI Base Data Types

RFC 2578 specifies the basic data types in the SMI MIB module-definition language. Although the SMI is based on the ASN.1 (Abstract Syntax Notation One) [ISO 1987; ISO X.680 1998] object-definition language (see Section 8.4), enough SMI-specific data types have been added that SMI should be considered a data-definition language in its own right. The 11 basic data types defined in RFC 2578.

In addition to these scalar objects, it is also possible to impose a tabular structure on an ordered collection of MIB objects using the SEQUENCE OF construct; see RFC 2578 for details. Most of the data types in Table 8.1 will be familiar (or self-explanatory) to most students. The one data type we will discuss in more detail shortly is the OBJECT IDENTIFIER data type, which is used to name an object.

Visual 8



Basic data types of the SMI

Data type	Description
INTEGER	32 bit integer, as defined in ASN.1, with a value between -2^{31} and $2^{31}-1$ inclusive, or a value from a list of possible named constant values
Integer 32	32 bit integer with a value between -2^{31} and $2^{31}-1$ inclusive
Unsigned 32	Unsigned 32 bit integer in the range 0 to $2^{32}-1$ inclusive
OCTET STRING	ASN.1-format byte string representing arbitrary binary or textual data, up to 65535 bytes long
OBJECT IDENTIFIER	ASN.1-format administratively assigned (structured name); see Section 8.3
IPAddress	32-bit Internet address, in network byte order
Counter32	32-bit counter that increases from 0 to $2^{32}-1$ and then wraps around to 0.
Counter64	64-bit counter
Gauge32	32-bit integer that will not count above $2^{31}-1$ nor decrease beyond 0 when increased or decreased
TimeTicks	Time, measured in 1/100th of seconds since some event
Opaque	Uninterrupted ASN.1 string, needed for backward compatibility

SNMP Security and Administration

Visual 9

SNMP security and administration

- **Encryption:** DES-encrypt SNMP message
- **Authentication:** compute, send MIC(m,k):
compute hash (MIC) over message (m),
secret shared key (k)
- **Protection against playback:** use nonce
- **View-based access control**
 - SNMP entity maintains database of access rights, policies for various users
 - database itself accessible as managed object!

The designers of SNMPv3 have said that “SNMPv3 can be thought of as SNMPv2 with additional security and administration capabilities” [RFC 2570]. Certainly, there are changes in SNMPv3 over SNMPv2, but nowhere are those changes more evident than in the area of administration and security. The central role of security in SNMPv3 was particularly important, since the lack of adequate security resulted in SNMP being used primarily for monitoring rather than control.

As SNMP has matured through three versions, its functionality has grown but so too, alas, has the number of SNMP-related standards documented. This is evidenced by the fact that there is even now an RFC [RFC 2571] that “describes an architecture for describing SNMP Management Frameworks”! While the notion of an ‘architecture’ for ‘describing a framework’ might be a bit much to wrap one’s mind around, the goal of RFC 2571 is an admirable one – to introduce a common language for describing the functionality and actions taken by an SNMPv3 agent or managing entity. The architecture of an SNMPv3 entity is straightforward, and a tour through the architecture will serve to solidify our understanding of SNMP.

So-called SNMP applications consist of a command generator, notification receiver, and proxy forwarder (all of which are typically found in a managing entity); a command responder and notification originator (both of which are typically found in an agent); and the possibility of other applications.

The command generator generates the GetRequest, GetNextRequest, GetBulkRequest, and SetRequest PDUs (Protocol Data Units) and handles the received responses to these PDUs. The command responder executes in an agent and receives, processes, and replies to (using the Response message) received GetRequest, GetNext Request, GetBulkRequest, and SetRequest PDUs. The notification originator application in an agent generates Trap PDUs; these PDUs are eventually received and processed in a notification receiver application at a managing entity. The proxy forwarder application forwards request, notification, and response PDUs.

A PDU sent by an SNMP application next passes through the SNMP ‘engine’ before it is sent via the appropriate transport protocol. A PDU generated by the command generator application first enters the dispatch module, where the SNMP version is determined. The

PDU is then processed in the message-processing system, where the PDU is wrapped in a message header containing the SNMP version number, a message ID, and message size information. If encryption or authentication is needed, the appropriate header fields for this information are included as well; see RFC 2571 for details. Finally, the SNMP message (the application-generated PDU plus the message header information) is passed to the appropriate transport protocol. The preferred transport protocol for carrying SNMP messages is UDP (that is, SNMP messages are carried as the payload in a UDP datagram), and the preferred port number for the SNMP is port 161. Port 162 is used for trap messages.

We have seen above that SNMP messages are used to not just monitor, but also to control (for example, through the SetRequest command) network elements. Clearly, an intruder that could intercept SNMP messages and/or generate its own SNMP packets into the management infrastructure could wreak havoc in the network. Thus, it is crucial that SNMP messages be transmitted securely. Surprisingly, it is only in the most recent version of SNMP that security has received the attention that it deserves. SNMPv3 provides for encryption, authentication, protection against playback attacks and access control.

SNMPv3 security is known as user-based security [RFC 2574] in that there is the traditional concept of a user, identified by a user name, with which security information such as password, key value, or access privileges are associated.

Encryption

SNMP PDUs can be encrypted using the Data Encryption Standard (DES) in cipher block chaining mode. Note that since DES is a shared-key system, the secret key of the user encrypting data must be known by the receiving entity that must decrypt the data.

Authentication

SNMP combines the use of a hash function, such as the MD5 algorithm with a secret key value to provide both authentication and protection against tampering. The approach, known as HMAC (Hashed Message Authentication Codes) [RFC 2104] is conceptually simple.

Suppose the sender has an SNMP PDU, m , that it wants to send to the receiver. This PDU may have already been encrypted. Suppose also that both the sender and receiver know a shared secret key, K , which need not be the same key used for encryption. The sender will send m to the receiver. However, rather than sending along a simple Message Integrity Code (MIC), $MIC(m)$, that has been computed over m to protect against tampering, the sender appends the shared secret key to m and computes a MIC, $MIC(m,K)$ over the combined PDU and key. The value $MIC(m,K)$ (but not the secret key!) is then transmitted along with m .

When the receiver receives m , it appends the secret key K and computes $MIC(m,K)$. If this computed value matches the transmitted value of $MIC(m,K)$, then the receiver knows not only that the message has not been tampered with, but also that the message was sent by someone who knows the value of K , that is, by a trusted, and now authenticated, sender. In operation, HMAC actually performs the append-and-hash operation twice, using a slightly modified key value each time; see RFC 2104 for details.

Protection against Playback

Nonces (a number that a protocol will use only once in a life time) can be used to guard against playback attacks. SNMPv3 adopts a related approach. In order to ensure that a received message is not a replay of some earlier message, the receiver requires that the sender include a value in each message that is based on a counter in the receiver. This counter, which functions as a nonce, reflects the amount of time since the last reboot of the receiver's network management software and the total number of reboots since the receiver's network-management software was last configured. As long as the counter in a received message is within some margin of error from the receiver's actual value, the message is accepted as a non-replay message, at which point it may be authenticated and/or decrypted (RFC 2574 for details).

Access Control

SNMPv3 provides a view-based access control [RFC 2575] that controls which network-management information can be queried and/or set by which users. An SNMP entity retains information about access rights and policies in a Local Configuration Datastore (LCD). Portions of the LCD are themselves accessible as managed objects, defined in the View-based Access Control Model Configuration MIB [RFC 2575], and thus can be managed and manipulated remotely via SNMP.

Principles in Practice

There are hundreds (if not thousands) of network management products available today, all embodying to some extent the network management framework and SNMP foundation that we have studied in this section. A survey of these products is well beyond the scope of this text and (no doubt) the reader's attention span.

Network management tools divide broadly into those from network equipment vendors that specialize in the management of the vendor's equipment and those aimed at managing networks with heterogeneous equipment.

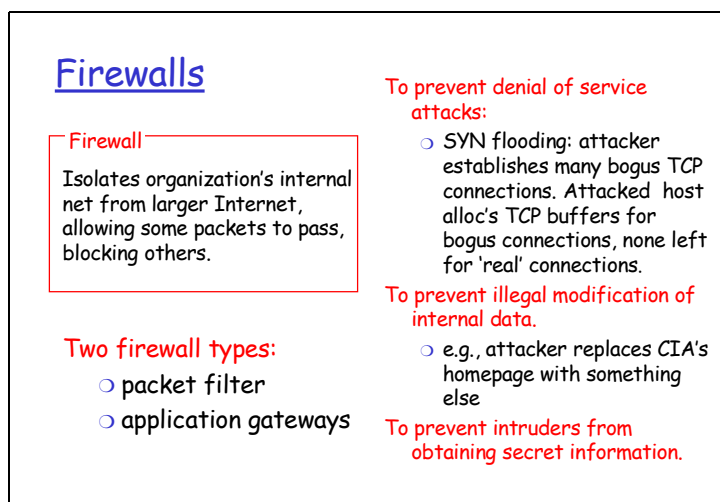
Among the vendor-specific offerings is CiscoWorks2000, for the management of LANs and WANs built on a Cisco device foundation. 3Com's Transcend network management system is SNMP-compliant and provides '3Com SmartAgent intelligent agent' technology to aid in network management. Nortel's Optivity Network Management System provides for network management, service management and policy management (bandwidth management, QoS, application-level security, and IP/address).

Among the popular tools for managing heterogeneous networks are Hewlett-Packard's Openview Aprisa's Spectrum, and Sun's Solstice network management system.

All three of these systems adopt a distributed system architecture in which multiple servers gather network management information from their managed domain. The network management station can then gather results from these servers, display information, and take control actions. All three products support the SNMP and CMIP protocols, and provide automated assistance for event/alarm correlation.

Firewalls

Visual 10



The Internet is not a very 'safe' place – for example, hackers are 'out there' breaking into networks at an alarming rate. (For a summary of reported attacks, see the CERT Coordination Centre (www.cert.org)). As a result, network administrators must be concerned not only with keeping the bits flowing smoothly through their network, but also with securing their network infrastructure from outside threats.

We have seen that SNMPv3 provides authentication, encryption, and access control in order to secure network management functions. While this is important (certainly, the network administrator does not want others to gain access to network-management functionality), it is only a small part of the network administrator's security concerns. In addition to monitoring and controlling the components of one's network, a network administrator also wants to exclude unwanted traffic (that is, intruders) from the managed network. This is where firewalls come in. A firewall is a combination of hardware and software that isolates an organization's internal network from the Internet at large, allowing some packets to pass and blocking others.

Organizations employ firewalls for one or more of the following reasons:

- **To prevent intruders from interfering with the daily operation of the internal network.** An organization's competitor – or just some Internet prankster looking for a good time – can wreak havoc on an unsecured network. In the denial-of-service attack, an intruder monopolises a critical network resource, bringing the internal network (at its network administrator) to its knees. An example of a denial-of-service attack is so-called SYN flooding in which the attacker sends forged TCP connection-establishment segments to a particular host. The host sets aside buffer space for each connection, and within minutes there is no TCP buffer space left for 'honest' TCP connections.
- **To prevent intruders from deleting or modifying information stored within the internal network.** For example, an attacker can attempt to meddle with an organization's public presence on a Web server – a successful attack may be seen by thousands of people in a matter of minutes.

Attackers may also be able to obtain customer purchase-card information from Web servers that provide Internet commerce.

- **To prevent intruders from obtaining secret information.** Most organizations have secret information that is stored on computers. This information includes trade secrets, product-development plans, marketing strategies, personal employee records, and financial analysis.

The simplest firewall consists of a packet filter. More sophisticated firewalls consist of combinations of packet filters and application gateways, topics we cover in the following two subsections.

Packet Filtering

Visual 11

Packet filtering

- Internal network is connected to Internet through a router.
- Router manufacturer provides options for filtering packets, based on:
 - source IP address
 - destination IP address
 - TCP/UDP source and destination port numbers
 - ICMP message type
 - TCP SYN and ACK bits
- Example 1: block incoming and outgoing datagrams with IP protocol field = 17 and with either source or dest port = 23.
 - all incoming and outgoing UDP flows and telnet connections are blocked.
- Example 2: Block inbound TCP segments with ACK=0.
 - prevents external clients from making TCP connections with internal clients, but allows internal clients to connect to outside.

An organisation typically has a router that connects its internal network to its ISP (and hence to the larger public Internet). All traffic leaving and entering the internal network passes through this router. Most router manufacturers provide options for filtering; when these options are turned on, the router becomes a filter in addition to a router. As the name implies, a filter lets some datagrams pass through the router and filters out other datagrams. Filtering decisions are typically based on:

- The IP address the data is (supposedly) coming from.
- IP destination address.
- TCP or UDP source and destination port.
- ICMP message type.
- Connection initialization datagrams using the TCP SYN or ACK bits.

As a simple example, a filter can be set to block all UDP segments and all Telnet connections. Such a configuration prevents outsiders from logging onto internal hosts using Telnet, insiders from logging onto external hosts using Telnet, and ‘weird’ UDP traffic from entering or leaving the internal network. The router filters the UDP traffic by blocking all datagrams whose IP protocol field is set to 17 (corresponding to UDP); it filters all Telnet connections by blocking all TCP segments (each encapsulated in a

datagram) whose source or destination port number is 23 (corresponding to Telnet). Filtering of UDP traffic is a popular policy for corporations – much to the chagrin of leading audio and video streaming vendors, whose products stream over UDP in the default mode. Filtering Telnet connections is also popular, as it prevents outside intruders from logging onto internal machines.

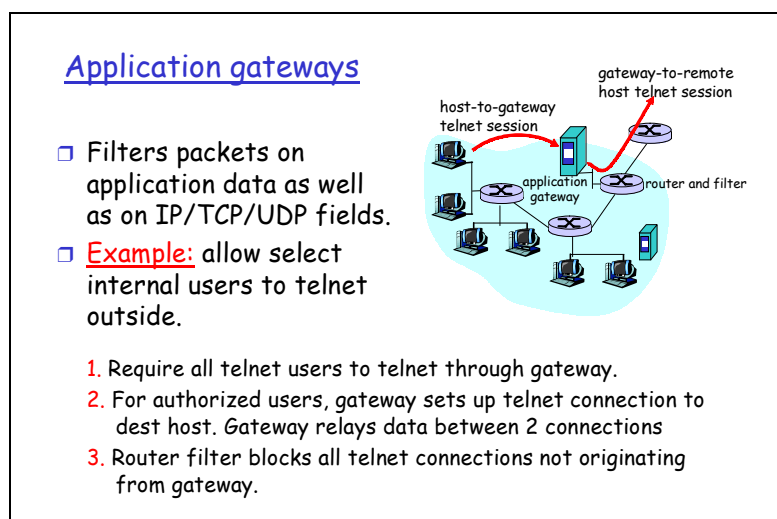
A filtering policy can also be based on the combination of addresses and port numbers. For example, the router can forward all Telnet packets (port 23) except those going to and coming from a list of specific IP addresses. This policy permits Telnet connections to and from hosts on the list. It is highly recommended to reject all datagrams that have internal source IP addresses – that is, packets that claim to be coming from internal hosts but are actually coming in from the outside. These packets are often part of address spoofing attacks, whereby the attacker is pretending to be coming from an internal machine. Unfortunately, basing the policy on external addresses provides no protection from an external host claiming to be a different external host.

Filtering can also be based on whether or not the TCP ACK bit is set. This trick is quite useful if an organization wants to let its internal clients connect to external servers, but wants to prevent external clients from connecting to internal servers. You will recall from Chapter 3 that the first segment in every TCP connection has the ACK bit set to 0 whereas all the other segments in the connection have the ACK bit set to 1. Thus, if an organization wants to prevent external clients from initiating connections to internal servers, it simply filters all incoming segments with the ACK bit set to 0. This policy kills all TCP connections originating from the outside, but permits connections originating internally.

Now suppose an organization does not want to block all connections originating from outside; instead it just wants to block only the Telnet connections originating from outside. This can be done by blocking inbound packets with destination port 23, or outbound packets with source port 23.

Application Gateways

Visual 12



Filters allow an organization to perform coarse-grain filtering on IP and TCP/UDP headers, including IP addresses, port numbers, and acknowledgment bits. We saw that

filtering based on a combination of IP addresses and port numbers can allow internal clients to Telnet outside while preventing external clients from Telneting inside. But what if an organization wants to provide the Telnet service to a restricted set of internal users? Such a task is beyond the capabilities of a filter. Indeed, information about the identity of the internal users is not included in the IP/TCP/UDP headers, but is instead in the application-layer data.

In order to have a finer-level security, firewalls must combine packet filters with application gateways. Application gateways look beyond the IP/TCP/UDP headers and actually make policy decisions based on application data. An application gateway is an application-specific server through which all application data (inbound and outbound) must pass. Multiple application gateways can run on the same host, but each gateway is a separate server with its own processes.

To get some insight into application gateways, let us design a firewall that allows only a restricted set of internal users to Telnet outside and prevents all external clients from Telneting inside. Such a policy can be accomplished by implementing a combination of a packet filter (in a router) and a Telnet application gateway. The router's filter is configured to block all Telnet connections except those that originate from the IP address of the application gateway.

Such a filter configuration forces all outbound Telnet connections to pass through G. When an internal user wants to Telnet to the outside world, it first sets up a Telnet session with the application gateway. An application running in the gateway, which listens for incoming Telnet sessions, prompts the user for its user id and password. When the user supplies this information, the application gateway checks to see if the user has permission to Telnet to the outside world. If not, the Telnet connection from the internal user to gateway is terminated by the gateway.

If the user has permission, then the gateway:

- (1) Prompts the user for the hostname of the external host to which the user wants to connect.
- (2) Sets up a Telnet session between the gateway and the external host.
- (3) Relays to the external host all data arriving from the user, and relays to the user all data arriving from the external host.

Thus the Telnet application gateway not only performs user authorization but also acts as a Telnet server and a Telnet client. Note that the filter will permit Step (2) because the gateway initiates the Telnet connection to the outside world.

Internal networks often have multiple application gateways, for example, gateways for Telnet, HTTP, FTP, and e-mail.

Application gateways do not come without their **disadvantages**:

- First, a different application gateway is needed for each application.
- Second, either the client software must know how to contact the gateway instead of the external server when the user makes a request, and must know how to tell the application gateway what external server to connect to, or the

user must explicitly connect to the external server through the application gateway.

Limitations of Firewalls and Gateways

Visual 13

Limitations of firewalls and gateways

- **IP spoofing:** router can't know if data 'really' comes from claimed source
- If multiple applications need special treatment, each has own application gateway.
- Client software must know how to contact gateway.
 - e.g., must set IP address of proxy in Web
- Filters often use all or nothing policy for UDP.
- Tradeoff: **degree of communication with outside world, level of security**
- Many highly protected sites still suffer from attacks.

Firewalls are by no means a panacea for all security problems. They introduce a tradeoff between the degree of communication with the outside world and level of security. Because filters cannot stop spoofing of IP addresses and port numbers, filters often use an all-or-nothing policy (for example, banning all UDP traffic). Gateways can have software bugs, allowing attackers to penetrate them. Also, firewalls are even less effective if the internal users have wireless communication with the external world.

Case History

The Limitations of Firewalls

In February 2000, a number of major Internet commerce sites were brought to their knees by a Distributed Denial of Service (DDoS) attack. The attackers first hit Yahoo!, and then spread their offensive to other major sites, including Amazon.com, eBay, CNN.com, and Buy.com. In the case of Yahoo!, at the worst moment, less than 10 percent of Yahoo's customers could access a page.

A Denial of Service (DoS) attack is an attack in which the aggressor swamps a host or a set of hosts with incoming packets – a kind of packet blitzkrieg. For a Web site, the aggressor most easily does this by sending massive numbers of HTTP requests, using destination port 80, to the Web site. The Web site then becomes bogged down in serving the bogus requests, causing the TCP connections carrying the bona fide requests to time out. Firewalls can provide limited protection from a denial-of-service attack – by identifying the source IP address of the perpetrator and filtering out all packets with that IP address.

But the perpetrators of the February 2000 attack used some simple (and well-known!) tricks to break the superficial defenses of a firewall. First, they planted programs called 'zombies' in at least 50 innocent hosts, most of which were residing at universities and research institutions. At a given time, they then commanded the zombies to attack the

Yahoo site – the zombies swamped Yahoo!, and then the other sites, with TCP connections. Whoever was behind the attacks did not gain root access on any targeted machine, and no proprietary information was stolen. But the attackers did succeed at bringing many major sites to their knees.

What can be done to prevent such distributed denial-of-service attacks? There doesn't appear to be a clear and easy answer to this question. One approach is to find the perpetrators and prosecute them – thereby discouraging other attackers. But it appears the attackers have left few electronic traces for determining their identities. Investigators are therefore taking a more traditional approach, using informants in the digital underground to try to gain information on who might be behind the attacks.

Summary

Our study of network management, and indeed all of networking, is now complete!

In this final chapter on network management, we began by motivating the need for providing appropriate tools for the network administrator – the person whose job it is to keep the network 'up and running' – for monitoring, testing, polling, configuring, analyzing, evaluating and controlling the operation of the network.

Visual 14

Summary

- ❑ The architecture of network-management systems revolves around five key components
- ❑ How SNMP instantiates the five key components
- ❑ MIB objects, SMI, SNMP protocol
- ❑ Firewalls, packet filtering and application-level gateways

We saw that the architecture of network-management systems revolves around five key components:

- (1) a network manager;
- (2) a set of managed remote (from the network manager) devices;
- (3) the management information bases (MIBs) at these devices, containing data about the device's status and operation;
- (4) remote agents that report MIB information and take action under the control of the network manager; and
- (5) a protocol for communicating between the network manager and the remote devices.

We then delved into the details of the Internet Network Management Framework, and the SNMP protocol in particular. We saw how SNMP instantiates the five key components of a network management architecture, and spent considerable time examining MIB objects, the SMI – the data-definition language for specifying MIBs, and the SNMP protocol itself.

Finally, we concluded this chapter with a discussion of firewalls – a topic that falls within the realms of both security and network management. We saw how packet filtering and application-level gateways can be used to provide the network with some level of protection against unwanted intruders, perhaps allowing the network manager to sleep better at night, knowing the network is relatively safe from these intruders.

It is also worth noting that there are many topics in network management that we chose not to cover – topics such as fault identification and management, pro active anomaly detection, alarm correlation, and the larger issues of service management for example, as opposed to network management.

