Chapter 6

Understanding DSS Architecture, Networking, and Security Issues

INTRODUCTION

Information technology (IT) architectures and computing infrastructures are evolving rapidly in corporations. In some companies, the IT infrastructure is being built in an uncoordinated, opportunistic manner. This approach is understandable given the rapid pace of technological change, but companies need much more than a "Web server here and a router there" approach to IT architecture and networking. Managers need to take steps to design an infrastructure that meets the following evaluation criteria: 1) minimizes support costs and maximizes user productivity; 2) avoids system crashes and other performance problems; and 3) reduces infrastructure impediments that delay the deployment of new IS/IT applications, especially Decision Support Systems (DSS). A network is the critical element of the IT infrastructure that supports enterprise-wide and communications-driven DSS.

According to Evans and Wurster in a 1997 *Harvard Business Review* article, the "rapid emergence of universal technical standards for communication, allowing everybody to communicate with everybody else at essentially zero cost, is a sea change." They note, "It is easy to get lost in the technical jargon, but the important principle here is that the same technical standards underlie all the so-called Net technologies: the Internet, which connects everyone; extranets, which connect companies to one another; and intranets, which connect individuals within companies." Both managers and MIS staff need to understand the magnitude of this sea change in how people can communicate.

One could reasonably ask how the DSS architecture and IS/IT infrastructure is related to networking and security issues. First, part of a DSS architecture is the network design. Second, security issues for a DSS are directly affected by architecture and network choices. These three topics of architecture, networking, and security are closely intertwined and are very important issues for building DSS. Unless one builds a DSS on a stand-alone computer in a secured office environment and keeps the computer under the watchful eye of the manager who is using it, designers and managers need to address DSS architecture, networking, and security issues. If one wants to design, develop, and implement successful DSS, it is important to understand these three fundamental technical topics.

This chapter explores the basics of DSS architecture, enterprise-wide networks and extranets, and security issues. The linkages among these issues are also explored.

DSS ARCHITECTURE AND IS/IT INFRASTRUCTURE

Many academics discuss building DSS in terms of four major components - the user interface, a database, models and analytical tools, and the DSS architecture and network (see Figure 6.1). One can label these components collectively as the overall architecture of a DSS. This traditional view of DSS components remains useful because it identifies commonalities between different types of DSS, but it provides only an initial perspective for understanding the complexity of DSS architectures.

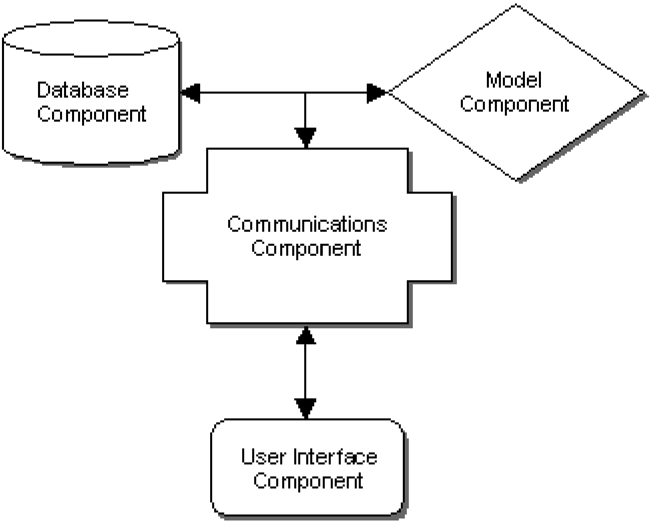

Figure 6.1. DSS Components.

As noted previously, a major component in the design of a DSS is the user interface. The tools for building the user interface are sometimes termed DSS generators, query and reporting tools, and front-end development packages. DSS user interfaces can be distributed to clients in a "thick-client" architecture or delivered over a network using Web pages or Java applets in a "thin-client" architecture. A thin-client architecture, where a user interacts using a Web browser, has many advantages, but until recently, the sophistication of the user interface was limited, compared to a thick-client architecture, where a program resides on a DSS user's computer.

A DSS database is a collection of data organized for easy access and analysis. Large databases in enterprise-wide DSS are often called data warehouses or data marts. Document or unstructured data is stored differently than structured data. Web servers provide a powerful platform for unstructured data and documents. The architecture for a data-driven DSS often involves databases on multiple servers, specialized hardware and in some cases both multidimensional and relational database software. The extraction, transformation, loading, and indexing of structured DSS data is difficult, and there are as many data engineering strategies as there are data warehouses.

Mathematical and analytical models are an important part of many DSS, especially model-driven DSS. Model management software can be centralized on a server with a database, or specific models can be distributed to client computers. Java applets and JavaScript programs provide a powerful new means to deliver models to users in a thin-client architecture.

The DSS architecture and network component refers to how hardware is organized, how software and data are distributed in the system, and how components of the DSS are integrated and physically connected. A major issue today is whether a specific DSS should only be available using thin-client technology on a company intranet or available on the global Internet. This should depend on the needs analysis and feasibility study. Scalability is also an important DSS issue. Scalability refers to the ability to "scale" hardware and software to support larger or smaller volumes of data and more or fewer users. Scalability also refers to the possibility of increasing or decreasing size or capability of a DSS in cost-effective increments.

The DSS framework discussed in Chapter 1 showed the different emphases that are placed on DSS components when specific types of DSS are actually constructed. Architecture, networking and security issues vary for data-driven, document-driven, model-driven and knowledge-driven DSS. Multi-participant systems like group and interorganizational DSS rely heavily on network technologies. The architecture of a data-driven DSS emphasizes database performance and scalability. Most model-driven DSS architectures store the model software on a server and distribute the user interface software to clients. Networking issues create challenges for many types of DSS, but especially for a geographically distributed, multiparticipant DSS.

An architecture for any information system is a formal definition of its elements and subsystems, including decision support systems. A DSS architecture can often be diagrammed in terms of four layers: the business decision process flow chart, the systems architecture, the technical architecture, and a user interface design. The business decision process flow chart shows what tasks are completed. The systems architecture shows the major software components. The technical architecture focuses on hardware, protocols, and networking. The user interface design focuses on outputs and capabilities of the system (see Chapter 5). The architecture also defines the structures and controls that define how the platform can be used, and the categories of applications that can be created on the platform. It includes the hardware and software used to manage information and communication; the tools used to access, package, deliver, and communicate information; the standards, models, and control frameworks; and the overall configuration that integrates the various components (cf., Applegate et al., 1996). Table 6.1 identifies some of the architecture requirements for different categories of DSS.

Type	Network Needed	Components
Communications-driven and GDSS	Always	Message storage, process support for GDSS
Data-driven	Usually	Web-enabled data access
Document-driven	Usually	HTML, TXT and PDF file storage and searching
Knowledge-driven	Sometimes	AI, statistical models, Web delivery
Model-driven	Sometimes	Optimization, Simulation processing
Interorganizational	Always	Depends on purpose

Table 6.1. DSS Framework and Architecture Issues.

Defining the DSS Architecture

A DSS architecture includes the IS/IT architecture components relevant to the DSS. A DSS may be a subsystem of a larger information system and a specific DSS may have multiple types of decision support subsystems. Having a well-defined and well-communicated DSS architecture provides an organization with significant benefits. An architecture document helps developers work together, improves planning, increases the development team’s ability to communicate system concepts to management, increases the team’s ability to communicate needs to potential vendors, and increases the ability of other groups to implement systems that must work with the DSS. Technical benefits of a DSS architecture document include the ability to plan systems in an effective and coordinated fashion and to evaluate technology options within the context of how they will work rather than from a more abstract perspective. A DSS vision and an architecture document help communicate the future, and provide a consistent goal for making individual design decisions. Achieving all these benefits requires that both information system professionals and prospective DSS users cooperate closely in developing the architecture.

An architecture drawing provides the grand scheme of a large-scale DSS project. The overall architecture of a DSS should be diagrammed and understood before specific decisions are made. The nature of the architecture depends on the DSS. Small-scale DSS developed by individuals for their own use do not justify a major architectural planning effort, although the overall information system architecture of the organization may constrain the capabilities of desktop DSS. Enterprise-wide DSS do require careful architecture planning if they are to succeed. Figure 6.2 shows a high-level enterprise-wide data delivery architecture. In general, more detail about the hardware, networks, and software is needed in specifying the architecture than is shown in Figure 6.2.

According to Mallach (1994), a DSS architecture should define and specify the following components:

1. Database or databases, including any existing databases internal or external to the organization, and any databases that are created specifically for DSS use. The architecture schematic should identify who is responsible for different types of databases, including their accuracy, currency, and security.
2. Model or models, including information about their sources of data, processing, the organizational unit responsibility for maintaining them, and limits on access to them.
3. Software tools for users to access the database and the models, and software tools that system administrators can use to manage the database and the models.
4. Hardware and operating system platforms on which the databases and models reside, on which the programs run, and through which users access the DSS. Any constraints, such as a policy to standardize on products of a particular vendor or products that use a particular operating system, should be stated.
5. Networking and communication capabilities needed to connect the hardware platforms. These capabilities must support needs to connect to one or more servers and databases, needs of work group members to communicate within the group, and enterprise needs to link work groups to each other or to shared data. In many DSS situations the corporate network is used. In this case the network must be examined to make sure it meets present and future decision support traffic needs.

Mallach also claims potential users should be specified when a DSS architecture is designed. The specifications should state any assumptions about users' locations, jobs, levels of education, and any other factors that may affect their use of a specific DSS. This information can be part of the business decision process diagram or data flow diagram.

Bob Lambert, in a paper titled "Data Warehousing Fundamentals" (1996), has a similar list of architectural issues that need to be addressed. Lambert argues, "An architecture is a design completed early in a project that encompasses (but not necessarily in detail) all aspects of the finished product."

According to Lambert, a completely specified DSS architecture addresses a number of topics including:

- Major system components and the interfaces, connections, or communication paths among the components;
- Anticipated system enhancements, migration paths, and modifications.

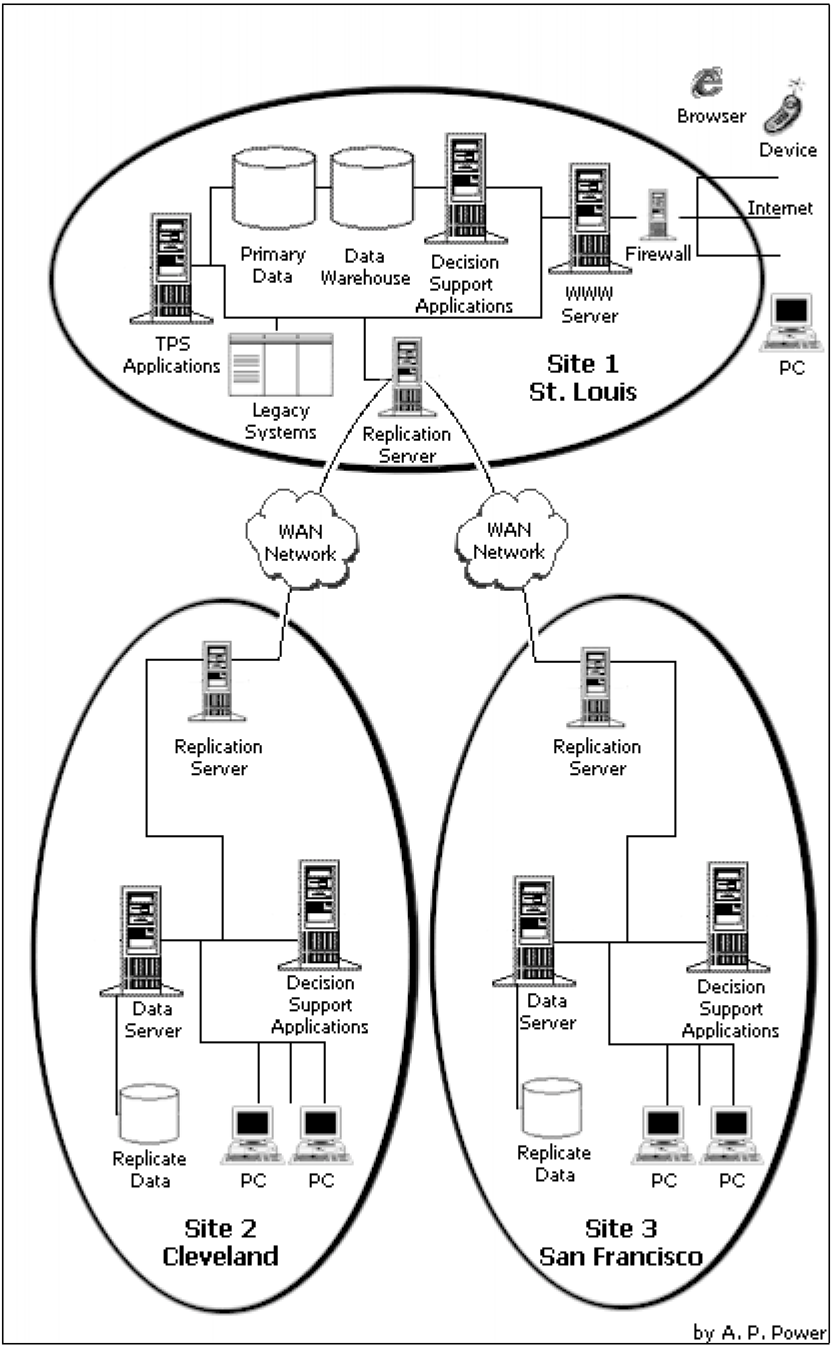

Figure 6.2 High-Level DSS Architecture.

Lambert notes, “All project participants should understand and accept the architecture. The architectural design should set a common level of understanding among technical, non-technical and management participants.”

A Client/Server Architecture

Most DSS are built within the context of a corporate-wide client/server architecture. Based on Taylor (1998), client/server refers to a computational architecture that involves client processes requesting service over a network from server processes. Ravi Kalakota in the Client/Server FAQ (Taylor, 1998) explains that client/server architectures are:

1) A combination of a client or front-end portion that interacts with the user and a server or back-end portion that interacts with the shared resource. The client process provides the interface between the user and the rest of the application system. The server process acts as a software engine that manages shared resources such as databases, analytical processors, or printers.

2) The client and server have fundamentally different requirements for computing resources such as processor speeds, memory, disk speeds and capacities, and input/output devices.

3) Scalable. An important characteristic of client-server systems is scalability. They can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or to multiple servers.

A common error in client/server development is to prototype an application in a small, two-tier architecture environment and then scale up by simply adding more users to the server. This approach usually results in an ineffective system because the server becomes overwhelmed. A three-tier architecture with a second “agent” server between the client and the server can support hundreds or thousands of users.

The Gartner group proposed terminology for describing different client/server styles, or organizing schemes, based on the distribution of the three components of an application: user interface, business analysis or application logic, and data management. The descriptive styles are distributed presentation, distributed function, and distributed data management. Distributed presentation is when only the user interface is processed on the client either using a Web browser or thick client interface. In a distributed function design, one part of the application processing is on the client, additional application processing is on one or more servers. Distributed function applications are the most complex type of design. In distributed data management, the entire application resides on the client, and data management is located on one or more remote servers/hosts. Web-based DSS are implemented using a distributed presentation design, but a DSS may also have distributed functions and distributed data management.

As noted, networks are a major element in the technical specification of a DSS architecture. The next section discusses this key architecture component.

NETWORKING ISSUES

Enterprise-wide DSS have interconnected servers, databases, and workstations. In many DSS development situations, an existing corporate network is used as part of the DSS architecture. In this situation the corporate network must be examined to make sure it meets present and future DSS traffic needs. Also, many DSS proposals are recommending Web-based DSS that are accessed from a client computer connected using the global Internet to a Web server. This architecture uses a public network based on the TCP/IP communications protocol.

This section summarizes a number of major issues in networking and computing communications that managers and DSS analysts should be familiar with so they can participate in networking discussions with network technical specialists. The following discussion is based on Frisch (1995), Nemeth, Snyder, Seebass, and Hein (1995), Kirkner, Ladd, O'Donnell, et al. (1996), and Jones (1997). The three major aims of this section are to:

1. Explain the basic concepts of networking;
2. Provide an explanation of what TCP/IP is and how it works;
3. Define some major networking terms.

Overview

A client/server architecture is based on having a physical network where computers act as either a server managing files and network services or as a client where users run applications and access servers. Clients rely on servers for resources like Web pages, databases, files, printing, and on-line analytical processing.

A network is a collection of computers connected in a way that allows them to communicate with each other and share information. To communicate, the computers need an agreed-upon language for communication. Networked computers are often referred to as hosts. Each host on a network must have some unique identifier that allows other hosts to communicate with it. Typical physical connections for hosts include Ethernet, token ring, serial line, and modems. Communication languages on computer networks are referred to as network protocols. A network protocol is a set of rules and formats that governs how information is sent and in what format it is sent. Some of the different network protocols used today include TCP/IP (Internet and UNIX), IPX (Novell), and Appletalk (cf., Hunt, 1992).

A number of technologies provide sharing of information, capabilities to distribute a DSS, and communications connectivity. These technologies include the Internet, private Integrated Services Digital Networks (ISDN), and remote access dial-up servers. Broadband service is another form of data transmission that uses cable television coaxial and fiber optic cables. Currently, the favored technology for many new DSS is the Internet because it is inexpensive, it is low risk, and it is a mature technology. Managers, customers and suppliers can use a dial-up or high-speed modem to connect to an Internet service provider or to their main office intranet. A major concern with using the Internet for DSS is managing security problems.

Sharing Resources

The fundamental purpose of computer networks is to provide access to shared resources, including storage for decision support data and information. One type of network for providing shared resources is a local area network (LAN). A LAN has several primary components:

- A network interconnection and hubs (for example, copper wire, fiber optic cable, infrared, or radio).
- Network Interface Circuitry (NIC) in the individual personal computers connected to the network.
- The shared resources, like a database server, each with their own NIC connected to the network.
- Software on a personal computer that uses the NIC to access the shared resources. This software is typically arranged to present the appearance to the rest of the operating system that these resources are directly connected.
- Software on the shared resource that coordinates with the software on the individual machines to provide access to the shared resources for users. This type of software is called a multi-user operating system. UNIX is a common operating system for DSS, but Windows NT is used in some architectures and for implementing some DSS packages.

The most common network design is for the server in a LAN to be the same sort of personal computer hardware as the individual personal computers on the network. In this case, the operating system is called a Network Operating System (NOS) to emphasize the difference from the single-user operating system of the personal computer. Novell Netware is an example of this approach. A NOS is an operating system that manages network resources. The NOS is like a traffic cop, controlling the exchange and flow of files, electronic mail, and print jobs. It manages multiple requests concurrently and provides the security needed in a multi-user environment.

A LAN is a communications network that serves users within a specific geographic area. It is made up of servers, workstations, a network operating system and a communications link. A wide area network (WAN) is a much larger network than a LAN, and all machines are not directly connected. A group of LANs are often connected to form a WAN. LANs and WANs can be directly connected to the global Internet.

Connecting the Resources: TCP/IP

The Transmission Control Protocol/Internet Protocol (TCP/IP) is the most widely used set of standard networking protocols. A networking protocol enables computers to communicate with one another.

The general concept of connecting a network of dissimilar computers arose from research conducted by the U.S. Defense Advanced Research Projects

Agency (DARPA). During that research, DARPA developed the TCP/IP suite of protocols to communicate among networks, and implemented a network called ARPAnet, which later evolved into the Internet. The TCP/IP suite of protocols defines formats and rules for the transmission and receipt of information independently of any given network organization or computer hardware. Although the protocols were developed for the Internet, they are also applicable to other cases where networks must be connected, including internal organizational networks called intranets. The Internet is a collection of networks and gateways that use the TCP/IP protocol suite.

Also, the Internet is a packet-switched network. A packet-switched network transmits information in small segments, called packets. If one computer transmits a lengthy file to another computer, the file is divided into many packets at the origin and then reassembled at the destination. Protocols define the format of these packets, including the origin of the packet and its destination, length, and type, as well as the way computers on the networks will receive and retransmit packets. TCP/IP routing capabilities allow forwarding of traffic from one network to another.

TCP/IP Protocol

The objective of TCP/IP is to get data from one host to another host, with the assumption that the connection may be difficult. IP provides three capabilities: 1) a delivery service; 2) a means to fragment and reassemble data packets; and 3) routing functions to move data packets on the network.

Data might start out in Seattle with a final destination in Australia. Along the way, many computers called routers with varying capabilities will be encountered. There might be heavy traffic that causes a particular route to be suboptimal, so the data might have to take another route. In addition, the router may not be able to transfer all the data, so the data has to be fragmented before continuing.

The TCP/IP protocol suite includes a number of protocols or rules. The Internet Protocol is a low level protocol that transports raw data over networks. The Transmission Control Protocol (TCP) sends data between programs using IP. As with all other communications protocol, TCP/IP is composed of layers.

TCP/IP assigns a unique address to every workstation in the world connected using TCP/IP. This “IP number” is a four-byte value that is created by converting each byte into a decimal number from 0 to 255 and separating the bytes with a period. For example, 131.123.2.25 is an IP number. Machines using TCP/IP also have natural language host names. A host name under TCP/IP follows the format hostname.site.domain.country. IP always uses the IP address and not the host name when it is sending information.

Why TCP/IP?

The growing acceptance of TCP/IP is due to several factors. First, TCP/IP has been used since the early 1970s. Second, in the early 1980s it was distributed as a core part of Berkeley’s UNIX Version 4.2 and UNIX workstations became primary servers on the Internet. TCP/IP was initially successful in the mid-1980s because it delivered a few basic services that many users needed (file transfer, electronic mail, remote logon) across a very large number of client and server systems. Several computers in a small department can use TCP/IP (along with other protocols) on a single LAN. The IP component provides routing from the department to the enterprise network, then to regional networks, and finally to the global Internet.

Third, TCP/IP is dependable. On the battlefield a communications network can be damaged, so DARPA researchers designed TCP/IP to be robust and to automatically recover from any node or phone line failure. This modular design allows the construction of very large networks with less central management. Because of its proven capabilities over Internets, its wide availability and support for routing, it has become an accepted standard for interconnecting heterogeneous environments from multiple vendors. Fourth, when organizations use TCP/IP, they can choose to use it exclusively over their own private intranet or as part of the global Internet.

The Internet Protocol was developed to create a network of networks called the Internet. Individual machines are first connected to a LAN. TCP/IP shares the LAN with other uses, for example, a Novell file server or a Windows for Workgroups peer-to-peer system. One hardware device provides the TCP/IP connection between the LAN and the rest of the Internet world. To insure that all types of systems from all vendors can communicate, TCP/IP is standardized on the LAN. TCP/IP and the Internet are not as secure as some alternative systems, but the system is available worldwide, and it is inexpensive. So managers and MIS professionals need to be concerned with maintaining security on networks using TCP/IP.

IMPROVING SECURITY FOR DECISION SUPPORT SYSTEMS

Security is a very important issue associated with building, managing, and using DSS. Reports of computer crime are increasing at a rate of more than 150 percent a year. Viruses and worms attack computers from e-mail message attachments. Hackers disrupt Web sites. Customer and credit card data have been stolen from Web servers. Company and customer data is valuable to competitors and thefts by unhappy employees, and hackers of company data do occur. Security *is* important.

Improving security for DSS involves addressing a number of issues. First, managers and MIS staff must determine security needs. Managers should ask what are the current security problems. This task is often called security evaluation. Based on the diagnosis in the evaluation stage, they need to implement the required security measures and fix any problems. These two tasks occur in what has been called the implementation stage. Once appropriate security is in place, one must monitor the system, and any new security problems need to be fixed. This is the feedback stage. Finally, managers and MIS staff need to stay informed about new security problems and methods for breaking into information systems. Both managers and MIS staff need to assume shared and equal responsibility for the security of DSS.

There are four major stages involved with implementing security for information systems and especially DSS. The four major stages are: evaluating security needs (evaluation), remedying problems and implementing solutions (implementation), observing and monitoring the operation of the system (feedback), and finally, staying informed (research) on security issues (cf., Jones, 1997).

Evaluation: Evaluating Security Needs

Before implementing any form of security, MIS staff need to decide how important security is for the company and identify any current security problems that need attention. This section examines these two steps, looks at some of the possible threats, and introduces some ways to evaluate security problems.

Information systems and especially DSS can be made very secure if enough effort is expended. A very secure system, however, is usually too inconvenient for managers to use. According to Jones (1997), when implementing a security plan, both system administrators and managers must weigh the following costs and factors:

- the importance of the system, its availability, and the data stored on it,
- the amount of effort required to make and keep the system secure, and
- how the security features will affect the users of the system.

A computer containing the plans for Intel's next computer chip or sensitive financial data should be carefully secured. On the other hand, it does not make sense to spend hundreds of thousands of dollars securing a computer used for e-mail by business students. A system can be made as secure as is necessary, but, in doing so, you might lose the ability to make effective use of it. Managers and systems administrators must balance the need for convenience against the need for security.

To implement security on a system, one should first identify the possible threats to the system. There are three major types of threats to a computer system: physical threats, denial of service, and unauthorized access. Physical threats include fire, theft of equipment, and vandalism. Denial of service means that people are unable to use a system because of some type of security breach. One way to deny service for Web servers is repeated and ongoing attempts to access the server that overwhelm its ability to meet legitimate requests for service. Unauthorized access means a "hacker" or a former employee has broken into a company's computers or Web site.

Not all denial of service attacks rely on expert knowledge of computer hardware and software. The quickest way of denying service is to steal or destroy the physical hardware. Mechanisms should be in place to prevent access to the physical hardware of a system. Network cables also create a security risk. The simplest way to disable a computer network is to take a shovel and dig up or cut a few of the cables used for a computer network. This problem may occur by design or accident.

To break into a DSS and gain access a hacker will generally go through a number of stages. The first stage is information gathering. During this phase, a hacker is trying to gather as much information about a site as possible, for example, what are the users' names, their phone numbers, office locations, what machines are there. Second, using the information gathered about a DSS or transaction processing system, a hacker tries to get a login account. It usually doesn't matter whose account. At this stage, the hacker is just interested in getting onto a specific machine.

Third, a hacker tries to get administrator privileges for the system. Hackers exploit bugs in programs and operating systems. Finally, a hacker makes changes to gain access and control of the system. Social engineering is one of the most used methods for gaining access, and it generally requires very little computer knowledge. The most common form of social engineering is for a hacker to impersonate an employee, usually a computer support employee, and obtain passwords or other security related information over the phone. Hackers also sift through the trash of an organization looking for passwords or other information. Some hackers actually get a job at a targeted site. Most hackers consider people to be the weak link in security.

Security threats are also caused by problems with computer software. These problems are caused either by misuse, by hardware incompatibilities, by people, by mistakes in programs, or by program interactions with other programs. MIS professionals need to evaluate the possibilities of technical problems.

Passwords are the first line of defense in the security of a computer system. They are also usually the biggest security problem. The main reason is that users perform actions with passwords that compromise their security including:

- writing their password on a "post it" note and then leaving it lying around,
- typing their passwords very slowly while someone is watching over their shoulders,
- choosing "dumb" passwords like their first name, and
- logging into their secure accounts across insecure connections.

These unfortunate actions by users make it easy for hackers to obtain passwords and bypass this important first line of defense. If a person has managed to crack someone's password and break into his or her account, the next step is to obtain an account with more access. The systems administrator is responsible for initially setting up file permissions correctly and then maintaining them.

The development of large-scale networks, especially global networks such as the Internet, has drastically increased the likelihood that a network-accessible DSS will be attacked. No longer is the worry only about people on site. All of the people on the Internet are now potential attackers. The security threat has increased.

Implementation: Remedying Problems and Executing Solutions

Having decided on an appropriate level of security and having identified security problems, MIS staff need to fix the problems and implement a security policy. A security policy can ensure the safe and organized use of resources. A computer security policy is a document that sets out rules and principles that affect the way an organization approaches security problems. Managers and MIS staff should specify the security rules for major Decision Support Systems. This section examines tools and methods that can be used to improve security with passwords, user education, file permissions, firewalls and secure servers.

Improving Password Security

There are a number of approaches managers and systems administrators can use to help make passwords more secure including: password user education, password generators, password aging, regular password cracking, and one-time passwords. Password generators create cryptic passwords for users, password aging forces periodic change of passwords, password cracking attempts to identify users with "bad" passwords, and one-time passwords are used only once. In some cases, a company wants to use one-time passwords. If a manager is traveling and needs access to sensitive data, we may want to change the password prior to each login.

User Education

Users do not want other people breaking into their accounts. If the users of a system are informed of the dangers of using "bad" passwords, most will choose "better" passwords. How an MIS staff performs user education depends on the users. Different users respond to different methods. System administrators must always remember that it is important not to alienate users, especially senior managers who need to use a DSS. User education is extremely important. DSS users need to learn about the importance of security and how to secure their passwords, equipment and data. One major problem is stolen portable computers. Managers can lose customer and decision support data and other important information.

File Permissions

Security is also related to the management of networked servers. The file system structure and file permissions are the fences of multiuser operating systems like UNIX and Windows NT. If used properly, permissions or rights to access files and directories and data can keep users in their own restricted areas on a server. The system administrator needs to monitor and maintain the rights and permissions granted DSS users. Inappropriate file permissions can lead to destruction of data and unauthorized use of data.

Firewalls

The Internet creates access for hackers who want to break into a DSS. By connecting to the Internet, a company opens a door for hackers. A firewall is designed to shut the doors. Basically, a firewall is a collection of hardware and software that forces all incoming and outgoing Internet data to go through one gate or door. It checks and logs requests made from outside the network to computing systems and requests made from internal users and systems to outside computing resources. The firewall software examines IP addresses and destinations of packets. Rules block access from certain IP addresses or block certain requests. For example, a rule may block file transfer protocol (FTP) requests from external IP addresses.

A firewall creates the following four advantages: protection of vulnerable or strategic services, concentration of security on the most important systems, enhanced privacy, and provision of logging and statistics on network use and users.

Secure Servers

Another security measure is to have a secure server and use encryption. A Web address for a secure server is displayed in a Web browser’s location field beginning with “https” rather than “http” when one enters a secure area. Most browsers also show either a closed lock or a solid key symbol in the status bar at the bottom of the screen. Companies should have a secure server for DSS applications that can be accessed over the Internet.

Feedback: Observing Operations and Maintaining Security Solutions

Once a system has been secured, the job is *not* over. Managers and system administrators must observe what people are doing with a DSS and determine if someone may have compromised the security of a specific DSS. Also, ongoing maintenance of security solutions is important. Operating systems can have “security holes” that are discovered; problems then need to be “plugged” with patches, and eventually the operating system needs to be upgraded. If this is not done, all systems on the server can be compromised.

Stay Informed: Use Web Resources

Managers must also stay informed about security needs and issues and system administrators must research routinely security issues and threats. The Web is the best source of current, timely Internet security and computer security information. Some useful Web hyperlinks include:

CERIAS, the Center for Education and Research in Information Assurance and Security at Purdue University, has a security hotlist at URL http://www.cerias.purdue.edu/hotlist/.

CERT Coordination Center at http://www.cert.org. CERT is a security watchdog and reporting group. Staff members provide technical assistance and coordinate responses to security compromises, identify trends in intruder activity, work with other security experts, and disseminate information to the broad community. CERT also analyzes product vulnerabilities, publishes technical documents, and presents training courses.

Sun Security site at http://java.sun.com/security focuses on UNIX and Sun Solaris security issues.

U.S. Department of Justice Computer Crime and Intellectual Property Section (CCIPS) Web site is at http://www.usdoj.gov/criminal/cybercrime.

World Wide Web Security FAQ by Lincoln D. Stein is at http://www.w3.org/Security/Faq/. It attempts to answer questions relating to the security implications of running a Web server and using Web browsers.

Email lists also provide alerts for System Administrators. MIS professionals with security responsibilities need to try to keep middle-level and senior managers informed about possible security problems.

CONCLUSIONS AND COMMENTARY

It is absolutely essential that a DSS have an appropriate architecture, network design, and level of security. Managers need to realize that the more widely accessible a DSS is, the more security problems that can occur. Managers also need to recognize that the greater the importance of DSS data, the greater the level of security that is needed. By connecting to the Internet, it is no longer a case of "if" a system will be broken into but rather "when." Despite the risks, it is my opinion that we have no choice but to use the Internet for accessing interorganizational DSS and Web-based DSS.

A well-defined DSS architecture has many benefits. Developing a DSS should therefore include adequate attention to the many important architecture issues. Networks provide the high-speed data transmission that many people have come to depend upon. So, managers need to understand the basics of how networks function. Security is not some specialist's responsibility. Managers and MIS staff need to learn about security issues; security for DSS data and systems is a shared responsibility. Managers need to remember that passwords are the first line of defense against unauthorized use of a DSS. Also, DSS users themselves often weaken the defense provided by passwords. There are a number of strategies that can be used to increase the effectiveness of passwords, but the most important is user education. Educate DSS users and remind them regularly of the importance of passwords.

Companies have become very dependent on the Internet, and managers need to be vigilant in their use of it. Attacks on a company's network can be anticipated and should be prevented when possible. The Internet is more than a physical network connecting millions of computers that can exchange information.

The future of distributed DSS capabilities is only limited by a company's technology infrastructure. Technology for DSS is expanding and improving rapidly. Networking technologies will become better, faster, and cheaper. Future technology will provide much higher speeds for video teleconferencing. Communications links will be both wired and wireless. The Internet has proven it can connect managers globally. The security issues associated with the Internet are being addressed proactively, and the Internet is now an integral part of distributing DSS capabilities to users.

Architecture, network and security issues must be examined together during the planning for a new DSS. Once a DSS is implemented, network and security monitoring must then become an ongoing activity.