Network Security In a Heterogeneous Environment

Ronald L. Sharp

Steven R. Eisen

W. E. Kleppinger

Mark E. Smith

This paper provides a framework for enhancing the security of networks, large and small. It delves into the problems faced when adding new network-security mechanisms. It also defines a process that can be used to determine what security mechanisms are required, and where to place them, within a network. A large portion of the paper describes the types of mechanisms that are available, and provides examples of AT&T network-security products.

Introduction

Very few hosts today are not connected to some type of network. It could be a local-area network (LAN), a wide-area network (WAN), or perhaps just a modem. There are many types of networks, with even more types of computers attached to them. In addition, there are many different networking protocols, including NetWare, TCP/IP, and SNA. (NetWare is a trademark of Novell, Inc.) Unfortunately, for most of these network components, security was not included in their initial design. In the last five years this situation has been changing, with vendors starting to provide security enhancements and new products for networks. The difficult part is determining the correct products to use—and where to integrate them—without adversely affecting network operation.

The Challenge

In order to retrofit security into an operational environment, the primary challenge is overcoming people's natural resistance to change. Implementing changes in security is especially difficult, because there is usually no perceived, added value to users. Security is often viewed as something that "gets in the way of real work." Without proper administration of security controls, this perception can be valid.

A basic goal of good security procedures is to restrict access to data and services only to those individuals having a legitimate need and who are authorized. A "tight" security policy will sometimes result in an authorized user being prevented from accessing a required resource, especially when a new product is being integrated and new procedures established. Such a situation can result in negative first impressions of a product. Any new security "solution" should first be tested, with a subset of users, in order to uncover potential problems.

Another concern is integrating security with existing network applications. A new security mechanism may prevent an application program from running properly. An example of this problem is a data-base application that bypasses the standard file-system interface and goes directly to "raw" disk to boost performance. A secure operating system will usually prevent access to the raw disk due to the security ramifications of such a capability. Again, the recommendation is to test-run a new security procedure, on only a portion of the network, prior to widespread implementation.

The best security is invisible to users and their applications. This is one of the primary goals for all products developed by the AT&T Secure Systems Engineering Department. The AT&T System V/MLS operating system, for example, modifies the UNIX kernel and some of its commands, but the resulting secure operating system is UNIX compatible, with no change to user applications. (UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/OPEN Corporation.)

Complete invisibility, however, is not always possible. For example, one of the

Panel 1. Abbreviations, Acronyms, and Terms

ARPA—Advanced Research Projects Agency
BSO—Basic Security Option
CIPSO—Common IP Security Option
DCE—distributed computing environment
DES—data encryption standard
firewall—special routers or gateways that examine all network packets and connections between two networks to determine if the communication is authorized
ftp—file transfer protocol
I&A—identification and authentication
IP—Internet protocol
LAN—local-area network
MAC—mandatory access control
NAAS—Network Audit Analysis System
telnet—remote login protocol
trust domain—geographical or conceptual groupings of users, hosts, networks, and associated resources
WAN—wide-area network

mechanisms provided by System V/MLS is Trusted Path. This capability guarantees that a user will get the true *login* program when establishing a login session. To initiate this capability, a user must press a secure, attention-key combination. This procedure requires a change in user behavior, which is not easy to accomplish. In System V/MLS, users are always required to use the attention-key combination. This routine trains users on the security mechanism before they log onto the system. The Microsoft Windows NT operating system employs the same attention-key concept. (Microsoft, Windows, and NT are registered trademarks of Microsoft Inc.)

The Process

How does one determine what type of security is needed and where it should be placed? There are no definitive answers to these questions. Security location and placement will depend on many factors, including:

- Existing network architecture (for example, a simple LAN or set of networks);
- External networks or hosts with which communication is required (for example, the public Internet);
- Communication services required between networks (for example, e-mail, telnet, and ftp);
- The real cost of a security violation (for example, the worst that could happen); and
- Funding for enhancements (for example, how much management is willing to pay for security).

In order to help answer the questions about security location and placement, a basic process is discussed in this section. The process is derived from—and is consistent with—the more general and formal approach described in the paper, "An Engineering Approach to System Security Analysis, Design, and Integration," also appearing in this issue of the *AT&T Technical Journal.*

Developing a Security Policy. The development of a network-security policy is the first important step. Such policies range from simple statements about network security philosophy to detailed rules and regulations. A description of the basic goal of network security should be the starting point. Without some overall goal, it is easy to concentrate on security against one threat and to ignore other threats. As a consequence, a security architecture having "steel doors with paper walls" could result.

A basic goal may be to protect only from outside threats and to provide a completely open environment for employees. Or, the sensitivity and criticality of network resources may require having a strict security policy for inside and outside users. After developing a well-defined goal, the process should proceed without delay. Later, the policy can be modified as knowledge of the situation grows.

Identifying Resources. It is impossible to provide protection for resources that are not known, and to do so without ascertaining their true value. A list of the resources requiring protection—and their locations—should first be developed. Such a list does not need to be highly detailed. A research and development LAN in a laboratory, or an employee data base containing proprietary information, could both be on the list.

When listing resources, the type of protection required for each should be included. The four basic types of resource security are protection from unauthorized reading, modification, creation, and use. For example, such resources as data bases may require protection only from modification and not from reading.

Depending on security policy, definition of resources may be required only at the host and network levels. Protection of files and data bases could be relegated

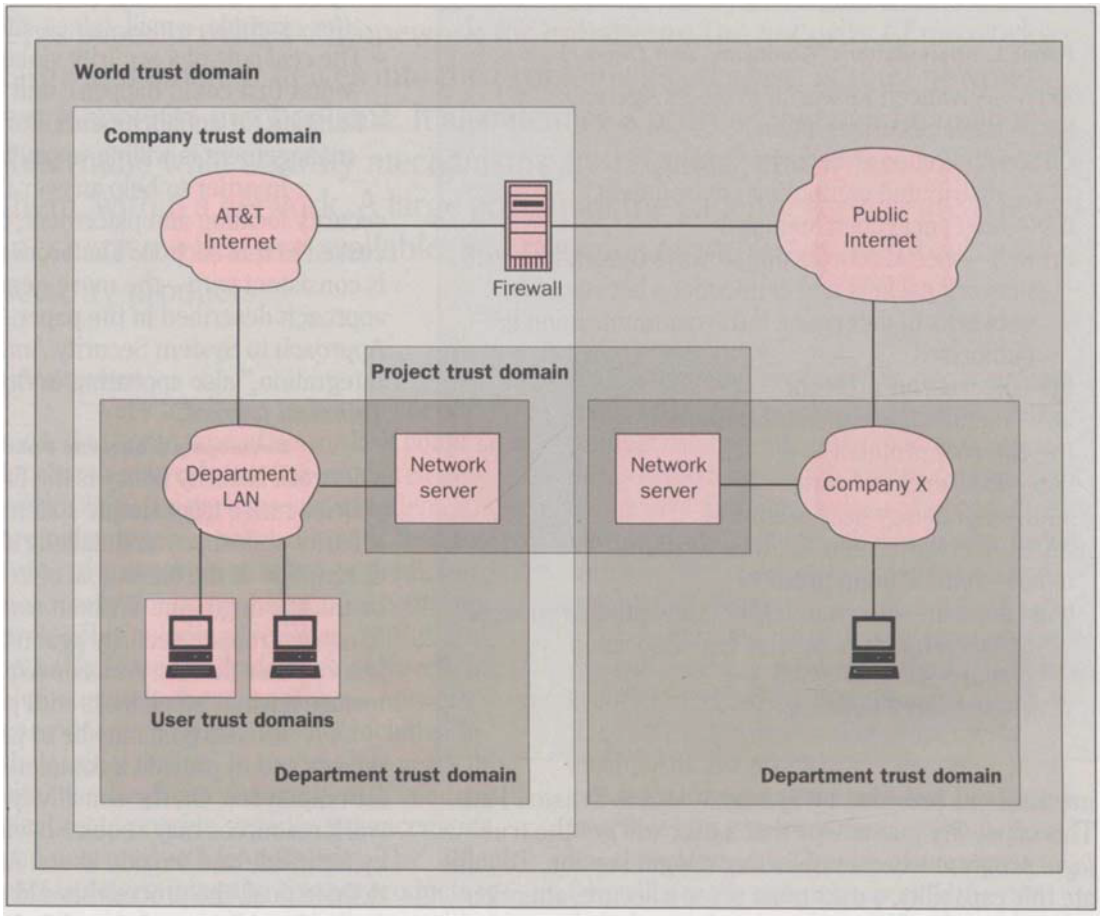

Figure 1. To assist in the determination of where protection needs to be placed, a network can be separated into *trust domains.* These represent geographical or conceptual groupings of users, hosts, networks, and associated resources. All users and hosts in the domain are trusted to access all resources in the same domain.

to the individual host's security policy. If the latter approach is taken, both the host's and network's security policies should be reviewed to ensure there are no conflicts.

Defining Trust Domains. To assist in the determination of where protection needs to be placed, a network can be separated into *trust domains*. These represent geographical or conceptual groupings of users, hosts, networks, and associated resources. All users and hosts in the domain are trusted to access all resources in the same domain. Figure 1 is a graphical representation of various trust domains.

Domains are hierarchical in nature. The users in a domain are trusted by the other users to access objects only within that domain. It is also assumed that the protection of internal subdomains is sufficient to prevent unauthorized access by the users in the next higher domain. This assumption is based on the strength of the protection mechanisms and the inherent trust of the users in the next higher domain.

For example, the Secure Systems Engineering Department of AT&T Bell Laboratories maintains sufficient protection for its local LAN to shield proprietary information from other AT&T employees. Such local security may not be sufficient, however, to safeguard company resources from users on the public Internet. The AT&T internal Internet organization is responsible for providing sufficient additional security at the higher trust-domain boundary, such as at the AT&T public Internet access point.

After the trust domains have been identified, a trust level can be assigned to each of them. Trust, in this context, is defined as a degree of confidence; users and hosts within a domain will not try to access resources in a subdomain for which they are not authorized. For example, users in AT&T departments typically can be trusted not to attack another department's LAN. Such

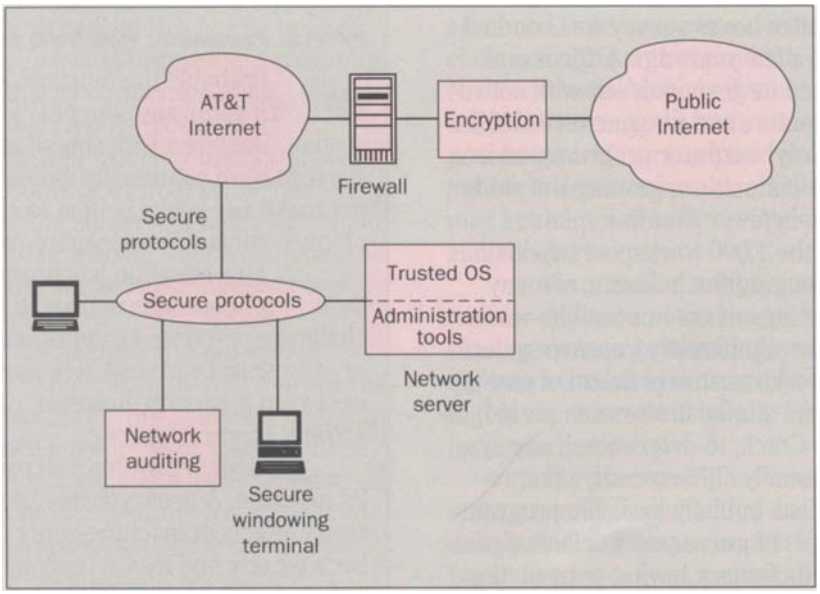

Figure 2. Several types of tools can be used to improve the security of a network. This illustration shows where in a network the tools discussed in this paper can be placed.

trust is crucial in determining the strength of the security mechanism required for this domain. Another important factor is identifying resources, and their associated value, as discussed in the previous subsection.

Selecting Mechanisms. In order to protect the interfaces between domains, mechanisms must be selected. For example, a secure operating system will protect single-user domains from others using the same host. The required "strength" of these mechanisms will depend on the trust placed on the domain and the value of the resources contained in the subdomains.

Selecting mechanisms can be very difficult. There are many to choose from, but they are not always easy to find—and even harder to evaluate—for a particular architecture. To assist in this process, a list of mechanisms and their uses has been included in "The Tools" section.

The architecture must be laid out, placing the security mechanisms in each domain. Existing mechanisms must be included, such as host passwords and physical safeguards. A personal computer, however, may not need boot-up security if it is locked in a room. Scrutinizing the layout ensures that no crucial areas are missed. Attention must also be focused on low-trust domains containing high-value resources, to ensure that adequate protection is provided.

Integrating and Evaluating. Finally, the required mechanisms are configured, purchased, or developed. Not everything must be implemented at once. In fact, it would probably be best to introduce security measures in phases in order to minimize network disruption. As recommended earlier, it is best to test a mechanism first, on a representative subsection of the network, to ensure that the mechanism works and does not inhibit authorized activity. Even users in high-trust domains will try to circumvent security if it gets in their way.

As new mechanisms are implemented, their effectiveness must be evaluated to determine whether they provide the level of protection needed. Some additional security may be required to offset a weakness. Obviously, this evaluation process is never ending and requires constant refining of controls. In addition, a typical network architecture is always changing, requiring constant attention to ensure that security between domains remains intact.

The Tools

This section describes several types of tools that can be used to improve the security of a network. Figure 2 illustrates where in a network the tools discussed in this section can be placed.

Administration Tools. As sophisticated as computer and network security sometimes appears to be, experiences both within AT&T and with customers reinforce the fact that weak system administration, low user awareness, and other unsophisticated exposures often cause the most significant security vulnerabilities.

An authorized, after-hours survey was conducted at several AT&T sites a few years ago. Offices and work spaces were checked for terminals left with active connections to host computers and for passwords left in plain sight, taped under keyboards, or programmed into keyboard scripts. The individuals performing the study succeeded in accessing—in fewer than two minutes each—more than 300 of the 1,000 workspaces visited.

Even though changing the behavior of many computer users is a daunting—if not impossible—task, system administrators can significantly improve system security rather easily. To address the problem of password security, for example, administrators can periodically run such programs as Crack, to determine if any system users have unintentionally chosen easily compromised passwords. Crack is a publicly available program that automatically attempts to guess passwords in a password file using a built-in dictionary having permutation rules. To prevent intruders from using Crack on a password file, resource owners should use a "shadow password" mechanism that hides the encrypted passwords from nonprivileged users. Better still—local culture permitting—machine-generated passwords can be used, which are fairly easy to remember yet difficult to guess.

A great many resources have appeared over the past few years that can be of enormous value to system administrators. For UNIX systems, COPS[1] is one of the most popular security packages. Like Crack, it is available on the Internet. The most widely used tool at AT&T is QSA, which is the successor to Quest [2], an AT&T computer-security software tool.

Organizations that have been formed to deal with computer-security problems can be invaluable information resources for system administrators. The Computer Emergency Response Team/Coordination Center, which is operated by the Software Engineering Institute for the Advanced Research Projects Agency (ARPA), is one such organization. System administrators should also stay in touch with their computer-system vendors, many of whom have dedicated staff to act as a clearing-house of information on security vulnerabilities for their products. System administrators may also discover in-house security functions in their own organizations that can provide support, tools, and information.

A variety of worthwhile Internet mailing lists and USENET news groups provides valuable information about security to system administrators, who can find a more extensive treatment of the subject in several excellent books that have appeared over the past few years. Curry's book on UNIX-system security is one such publication.[3]

Panel 2. Passwords: How Good Are They?

Probably the most obvious point of attack on a network is through the password-identification subsystem. Despite a widespread awareness of this fact, surveys have continually indicated that users often do not make any special effort to choose safe passwords or, once chosen, to safeguard password secrecy.

This situation has prompted some organizations to propose alternatives to passwords, such as challenge-response systems using hand-held devices, or even smart cards. Before completely rejecting the password approach, however, one should consider the *optimal password system.*

In such a system, all passwords are generated by machine. A user typically has a choice whether to reject or accept machine-generated passwords, because any one may or may not be particularly memorable. For the required level of user-friendliness, a password-generation system must present a large selection of passwords to a user at once. AT&T research indicates that if the generation algorithm is well-designed (using pronounceable syllables), a user will find at least one easily remembered password in a set of 50 alternatives.

It is important to realize that writing down a password is not, in itself, a poor security risk, as long as the password notation is safely hidden on the user's person (in a wallet or purse) until memorized. Such flexibility gives a user significant control over the password without compromising its security.

Trusted Operating Systems. If the resources to be protected are on a host (disk files, process spaces, shared memory, and so forth) and users in multiple security domains have access to that host, then a secure operating system is needed in order to preserve the trust domains.

Trust domains are instantiated on a host. As an example of how this is done, trust domains can be established on a UNIX host at the granularity of a user, or at the granularity of a set of users organized into a UNIX group.

Access to active resources (executable programs) or to passive resources (disk files, and so forth) are mediated through the access control provided by the

host. This basic host-access control is referred to as *discretionary access control*, so called because the creator of the resource has the authority to alter the ability of users in other trust domains to access the resource.

If a rule is established whereby users cannot "give away" access to resources without proper authority, it becomes necessary for the secure operating system to impose some sort of nondiscretionary controls. The name given to this type of control is *mandatory access control* (MAC).

The most common model for imposing MAC is the Bell-LaPadula model, which depicts trust domains in a much stricter sense than purely discretionary controls. Under Bell-LaPadula, once a resource or user has been assigned to a trust domain, it cannot be reassigned to another trust domain by anyone without a specific privilege to do so. Furthermore, access to the resource is strictly limited to users from a specific set of trust domains based on mathematical lattice rules. These concepts are thoroughly discussed in E. Amoroso's recent book on security technology.[4]

System V/MLS[5], a UNIX-compatible operating system, uses the Bell-LaPadula model to preserve the secrecy and integrity of host trust domains. It combines enhanced, discretionary controls with flexible, mandatory controls. In addition, System V/MLS has a very efficient auditing system containing an advanced set of audit-analysis tools.

Network Auditing and Event Detection. Access controls act like filters. Each control is very effective for screening out undesirable elements, but there is no control over elements allowed to pass through them. Moreover, all security controls suffer from the same basic problems—too many restrict legitimate activity, and too few could allow unauthorized resource access.

One solution is to require users to identify themselves continually, but this would not be acceptable in most environments. The best solution is a reliable auditing system that records the actions of authorized users and applications. The resulting audit trail can then be reviewed by an administrator for any unusual activity, although this can be a difficult and tedious job. There are tools available from some vendors, however, that make audit-trail analysis much easier. The AT&T Computer-Watch® Audit-Trail Analysis Tool[6] is used to examine a System V/MLS audit trail, providing detailed reports to the administrator. In addition, this program has an advanced, rule-base system to assist in identifying suspicious activity. Another product, called the AT&T Network Audit Analysis System (NAAS), is used to consolidate system audit trails across a network. This process provides a complete view of network activity and assists in detecting problems between systems, which may not be possible by reviewing only individual system audit trails.

An effective audit system can be a strong deterrent for anyone considering an unauthorized action. Placing a message in each day's *logon* greeting, informing users that all system activities are being audited, is highly recommended to help increase security, as well as for legal reasons.

For an even higher level of protection, there are software tools available that analyze an audit trail continuously, seeking out unusual activity. These tools normally use vendor-defined and administrator-defined rules for detection. When a set of conditions is met, a predetermined action is taken. The standard response to a condition set is the generation of an alarm, which is sent to the administrator. For the System V/MLS, this capability is provided by the AT&T SatWatch tool.

Firewalls. In order to separate trust domains at the network level, *firewalls* are used. These are special routers or gateways that examine all network packets and connections between two networks to determine if the communication is authorized. With the recent increase of companies connecting to the public Internet, firewalls have become a popular method of restricting access from the Internet into a company's private network. Typical firewall systems can be divided into *packet filters* and *application gateways*.

A packet-filter firewall performs its function at the Internet-protocol (IP) packet layer of the protocol stack. As each packet is received, a filter "looks" at the source and destination IP addresses, as well as the requested network service. The filter then allows or disallows passage of the packet based on a set of rules configured by the firewall administrator. An example of one such rule is to allow all hosts to send mail to the AT&T Mail host. Several router vendors, such as Cisco, provide this capability.

An application-gateway firewall requires hosts to pass through the gateway host to arrive at their desired destination. Special applications in the gateway host accept connections from other hosts and, if approved,

forward the connection to the requested host. Like packet filters, it is common to control access based on an administrator-defined rule set. Application gateways have the disadvantage that they are visible and, many times, require special software on the source host. They are more flexible than a packet-filter firewall, however, and can provide a finer level of control. For example, an ftp gateway proxy can provide control of the ftp commands that may be used between hosts.

The AT&T Secure Systems Engineering Department is developing a new security gateway product, referred to as NetWatch™ Trusted Security Gateway. This software program provides both packet-filter and application-gateway capabilities. The packet-filter mechanism uses a new "smart-packet" filtering capability, which provides much greater security control.

Additional information concerning firewalls can be found in a book, written by W. Cheswick and S. Bellovin of AT&T Bell Laboratories, titled "Firewalls and Internet Security."[7] It discusses how to build a firewall and identifies the many available products, some of which are even free of charge.

Encryption. In a network environment, such cryptographic techniques as encryption can be used to provide privacy, integrity, identification and authentication (I&A), and sender-nonrepudiation services. Privacy (preventing the disclosure of information) and integrity (preventing or detecting the modification of information) services can be provided using encryption and cryptographic *checksum* techniques. Privacy and integrity services can be provided at the link, network, or application levels, depending on the network security policy. A prerequisite, however, is that both parties (the user and network resource) must share a secret in the form of a *cryptographic key.*

Sender nonrepudiation can be provided by attaching a digital signature to a message. I&A of the claimed user identity is typically a prelude (either explicitly or implicitly, through prior possession of a cryptographic key) to other security services. In addition, in the approaches discussed in the following paragraphs, key management is integrated into the I&A process, enabling the subsequent use of privacy and integrity services.

Several cryptographic solutions have been developed to provide I&A, data privacy, and integrity. The Kerberos authentication system uses a trusted third party as an intermediary between a user and network resources for I&A purposes.[8] Kerberos authentication relies on encryption using the data encryption standard (DES) algorithm to protect I&A information.[9] It also establishes a shared encryption key between a user's work station and the network server (this key can be used to protect subsequent communications). The security services in the Open Software Foundation's distributed computing environment (DCE) standard are based on the Kerberos model.[10] The AT&T Surety™ Data Network System addresses these issues with a combination of public-key cryptography (the RSA algorithm) for key distribution and the DES algorithm for the subsequent provision of data privacy.[11]

Secure Protocols. Protocol-level security software provides two basic services:

- Transport of information to be used by other security controls on a host; and
- Access control of network resources.

Controls implemented in the protocol software can be used to manage access to other hosts and networks, as well as the route taken to arrive there.

A good example of protocol security is provided by the AT&T MLS/TCP software program. This product is an enhanced version of Wollongong's TCP/IP software that works together with the AT&T System V/MLS secure operating system. For the operating system, MLS/TCP software provides security labels associated with each network packet, as well as address information for auditing. It also facilitates complete control of communication between hosts, including the types of network services allowed (for example, telnet and ftp).

Advancements in protocol security have been extremely slow because they require changes to existing standards. Changing standards or creating new ones require consensus among multiple companies, as well as commercial and government organizations. Such approvals are difficult to obtain. It is very important not to purchase a product, however, that defines its own standard. This commits the purchaser only to that product, which may soon become obsolete. MLS/TCP uses two well-known standards to pass security labels: the Basic Security Option (BSO), and the Common IP Security Option (CIPSO).

Secure Windowing Terminals. A network having intelligent windowing terminals, such as X terminals,

presents an additional problem in trust-domain mediation. In this situation, it is usually possible to render data from various trust domains on a single windowed terminal. The windowing-system software driving the intelligent terminal has access, therefore, to information from all of these trust domains.

While the windowing-terminal software may allow simultaneous access by multiple users, it should not allow these users to access each other's data—at least not without appeal to some authority, typically the person using the terminal. This type of separation requirement is referred to as *user-based isolation*. Often, this requirement can be reasonably strengthened to restricting the access of one application to another (referred to as *client-based isolation)*.

These approaches, alone, do not solve the basic discretionary-access security problem. Individual users (and programs operating with their authority) are still capable of subverting either of the isolation schemes. For this reason, various nondiscretionary solutions have been proposed and developed, such as:

- The Compartmented Mode Workstation[12], which separates trust domains directly in the windowing-system software based on the Bell-LaPadula model; and
- The AT&T MLS/Xwin package that works by multiplexing the trust domains on the application side, thus presenting the windowing-system side with a single, virtual trust domain. MLS/Xwin also uses the Bell-LaPadula model to implement the application-side multiplexer.

Any secure windowing solution must be able to cut and paste information between windows safely. For the Compartmented Mode Workstation and MLS/Xwin, this capability is provided by an application-side program that intercepts the cut data before it can be pasted. The program can then apply whatever policy is appropriate for data movement between trust domains. This combination of a nondiscretionary policy and a method for mediating data movement across trust domains results in a safe windowing system.

Conclusion

It is possible to enhance the security of an existing network without disrupting people's work. A goal, a process to reach the goal, and the right tools are all that is needed. User acceptance—the biggest hurdle—can be achieved by implementing only the precise degree of security required, introducing new mechanisms in stages, and initially testing them on a small segment of the network . As with most security programs, the safeguarding of network resources is a continuous process, particularly due to the ever-changing nature of networks.

Acknowledgments

The authors wish to express appreciation to Chuck Flink and Ed Amoroso of AT&T Bell Laboratories for many helpful discussions on these matters and for their careful review of this paper. The information presented represents a portion of the collective knowledge of the AT&T Bell Laboratories Secure Systems Engineering Department.

References

1. D. Farmer and E. H. Spafford, "The COPS Security Checker," USENIX Conference Proceedings, 1992, pp. 165–190.
2. S. A. Kapilow and M. Cherepov, "QUEST - A Security Auditing Tool," *AT&T Technical Journal*, Volume 67, Number 3, May/June 1988, pp. 65–71.
3. D. A. Curry, "UNIX System Security: A Guide for Users and System Administrators," Addison-Wesley Publishing Company, 1992.
4. E. Amoroso, "Fundamentals of Computer Security Technology," Prentice Hall Publishing Company, 1994.
5. C. W. Flink and J. D. Weiss, "System V/MLS Labeling and Mandatory Policy Alternatives," *AT&T Technical Journal*, Volume 67, Number 3, May/June 1988, pp. 53–64.
6. C. Dowell and P. Ramstedt, "The ComputerWatch Data Reduction Tool," Proceedings of the 13th National Computer Security Conference, October 1–4, 1990.
7. W. R. Cheswick and S. M. Bellovin, "Firewalls and Internet Security," Addison-Wesley Publishing Company Inc., 1994.
8. J. G. Steiner, C. Newman, and J. I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," Proceedings of the Winter USENIX Conference, Dallas, Texas, 1988, pp. 191–202.
9. FIPS PUB 46-1, Data Encryption Standard, January 1977.
10. W. Rosenberry, D. Kenney, and G. Fisher, "Understanding DCE," O'Reilly & Associates, Inc., 1993.
11. R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," CACM, Volume 21, February 1978, pp. 120–126.
12. J. Woodward, "Security Requirements for System High- and Compartmented-Mode Workstations," MITRE MTR 9992, Revision 1, November 1987.

(Manuscript approved August 1994)

Ronald L. Sharp *is a member of the technical staff in the Secure Systems Engineering Department at AT&T Bell Laboratories, Whippany, New Jersey. He is currently performing research and development in the area of UNIX Multilevel Secure Networking and firewall technology. Mr. Sharp holds a B.S. degree in computer science from the University of Arkansas in Fayetteville, and an M.S. degree in computer science and systems design from the University of Texas in San Antonio. He joined AT&T in 1986.*

Steven R. Eisen *is a technical manager in the Secure Systems Engineering Department at AT&T Bell Laboratories, Whippany, New Jersey. He is currently responsible for management of secure product development and security systems engineering-services projects. Mr. Eisen, who is a Certified Project Management Professional, has a B.S. in electrical engineering and computer science from Columbia University in New York City, an M.S. in computer science from the University of California in Berkeley, and a Master's Certificate in Commercial Project Management from The George Washington University in Washington, D.C. He joined AT&T in 1979.*

W.E. Kleppinger *is a member of the technical staff in the Secure Systems Engineering Department of AT&T Bell Laboratories, Whippany, New Jersey. He is currently performing system security engineering on a variety of government and international projects. Mr. Kleppinger holds an A.B. degree from Princeton University in New Jersey, and M.S. and Ph.D. degrees from Stanford University in Palo Alto, California, all in physics. He joined AT&T in 1985.*

Mark E. Smith *is a member of the technical staff in the Secure Systems Engineering Department at AT&T Bell Laboratories in Greensboro, North Carolina. He is responsible for the development of systems software and windowing software within the department. Mr. Smith has a B.S. degree in computer science and mathematics from Vanderbilt University in Nashville, Tennessee, and an M.E. degree in computer science from Cornell University in Ithaca, New York. He joined AT&T in 1980.*