ENTERPRISE-WIDE SECURITY

A NAC POSITION PAPER

FIRST EDITION
JULY 14, 1996

ABOUT NAC

The Network Applications Consortium (NAC) is a strategic end-user organization dedicated to improving the interoperability of mission-critical applications in a heterogeneous inter-enterprise computing environment. The Consortium’s goal is to influence the strategic direction of vendors developing enterprise application and infrastructure technologies. NAC focuses on providing vendors with input and feedback regarding product development and marketing strategies by:

- publishing papers that state NAC’s strategic vision of the industry’s direction;
- educating vendors on end-user enterprise-wide computing requirements;
- promoting, facilitating, and documenting collaboration among NAC members and vendors;
- advising distributed computing vendors on marketing and product development strategies.

NAC members include:

ABB Power T&D Co.
American Bureau of Shipping
American Medical Securities
Australian Bureau of Statistics
Bell Atlantic Mobile Systems, Inc.
Carolina Power & Light Co.
Compaq Computer Corporation
Continental Grain Company
Federal Deposit Insurance Corporation
International Finance Corporation
MCI Telecommunications
Michigan Department of Commerce
Nike, Inc.
NYNEX
Pacific Bell
Pacific Gas & Electric
Pennsylvania Blue Shield
Public Service Electric & Gas
United States Marine Corps
University of Michigan
World Bank

This paper is the result of NAC’s Strategic Interest Group (SIG) process, a collaborative effort of a subset of NAC members whose mission is to provide a cohesive NAC viewpoint on a particular industry sector or technical topic. The following NAC members were instrumental in writing this paper:

Carolina Power & Light	*Steve McGehee*
Compaq Computer Corporation	*Mike Wilhite*
Federal Deposit Insurance Co.	*Chuck Taylor*
MCI Telecommunications	*Brian Plackis, Jerry Robinson, Sam Rockwell, Ron Thomas*
Pacific Bell	*James Brentano*
Pennsylvania Blue Shield	*Todd Sebastian*
Public Service Electric & Gas Co.	*Randy Back, SIG Leader*
United States Marine Corps	*Mark Johnson, Janet Palmer*
University of Michigan	*Gordon Leacock*
NetResults	*Doug Obeid*
Author:	*Kelli Wiseth*

We welcome your feedback about this paper. For more information contact:
Doug Obeid, Executive Director
Network Applications Consortium
c/o NetResults
5214-F Diamond Heights Blvd., Suite 705
San Francisco, CA 94131

Contents

Executive Summary

As a corporate asset, information is unique in that its value depends chiefly on an organization's ability to keep it private. But the need to secure information competes with the need for ready access by the appropriate parties. In this paper, NAC discusses some of the key issues relative to computer data security in the enterprise-wide environment today. Some of the issues are organizational; some are technical. As will be seen in NAC's security framework, presented in this paper, an organization's culture and security policy are inter-related issues that ultimately affect the success of the security strategy.

Security mechanisms should function behind-the-scenes, virtually invisible to end-users. Everyone in the organization should have access to all the information they need to function as an empowered participant in the organization's goals. At the same time, threats to the information should be minimized in a cost-effective manner. Security should be applied consistently across all platforms and data in the organization, and security analysts and administrators should be able to set, monitor, and administer the security policy from a single user interface.

But that's not the case: Security implementations are numerous and each application has its own mechanisms and methods for configuring and enforcing security policies. Users are faced with many logon scenarios and must remember several, sometimes dozens of passwords to perform different job functions. This is burdensome and frequently causes users to circumvent security measures by writing down IDs and passwords; thus security is all too often something that's avoided rather than embraced.

No one disputes that data security is a critical component of a distributed data environment, yet keeping information secure is an enormous expense to organizations in terms of lost productivity, increased training costs, and exorbitant administrative expense. The multitude of authentication mechanisms in place and used on a daily basis by millions of computer users every day highlights these issues. This paper provides a background discussion on the topic of enterprise-wide security with particular emphasis on authentication services, which NAC will explore in greater detail in its next paper.

Introduction

Keeping information assets secure — yet readily available to those who are authorized to use them — is an enormous challenge in the distributed, client-server computing environment, where resources are geographically dispersed across heterogeneous platforms.

As an asset, information has a unique quality: if an organization doesn't have the information exclusively, the information may lose its value. For example, two competing organizations may have comparable capital investment in plant, equipment, and other tangible resources without directly affecting each other's market position. But if an organization's strategic marketing plan falls into the hands of a competitor, the organization loses its competitive advantage.

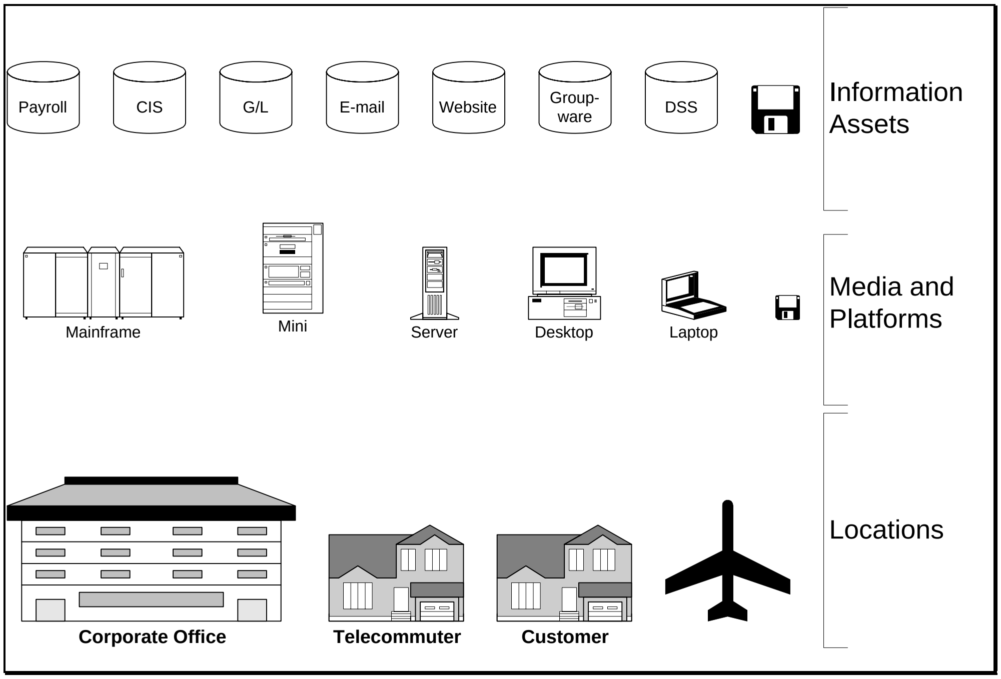
Thus, securing information is imperative: Information must be kept from those who would steal it, destroy it, modify its meaning, or otherwise cause the information to lose its value to the organization. Yet at the same time, information must be readily available to the appropriate parties within and outside the organization who need it to support the organization's goals.

A decade ago, meeting these two conflicting goals — securing information yet assuring its availability — was much easier to do. Data processing resources and access to them were centralized, physically secured, and tightly controlled. But in today's geographically dispersed business environment, computing resources are distributed throughout heterogeneous environments that defy boundary lines.

For example, many organizations are turning to telecommuting as a means of lowering real estate costs at the same time they respond to environmental legislation. As a result, employees are dialing-in to the organization's network at all hours of day and night, and on weekends. Business travelers around the globe dial-in from airports, restaurants, and hotels, checking their e-mail, posting messages on the organization's bulletin board, downloading extracts from the customer information database before a sales call.

As barriers within organizations disappear, inter-company walls are eroding as well. For example, rather than maintaining large inventories, many retailers communicate electronically with suppliers when it's time to restock items. Government, academia, and private industry use the Internet daily to communicate via e-mail about highly confidential research and development projects. Organizations are setting up web sites and ftp sites as a means of providing marketing materials, product information, and technical support on an anytime, anywhere basis.

Figure 1. Information Assets are Distributed across Heterogeneous Platforms



Information assets are distributed across heterogeneous platforms and a variety of physical locations, and thus are more difficult to control. Information assets are also very portable: critical information, such as a spreadsheet containing salary data or a report discussing key marketing objectives, can be easily copied onto a floppy and slipped into a shirt-pocket or purse and taken right past the guard station.

Furthermore, the notion of "information" has changed both quantitatively and qualitatively: Consider the volume of e-mail messages spread throughout an organization, which may likely contain some of an organization's most sensitive information.

Without question, information assets must be secure. Unprotected information assets may cost organizations not only in terms of strategic advantage, but also in direct dollar losses. In a well-publicized case last year, Citibank suffered close to a half-a-million-dollar loss before catching the high-tech thieves who manipulated Citibank's funds transfer system. Sometimes hackers break-in just to prove that they can, as was the case when Netscape security was compromised last year.

According to the FBI, computer crimes average about $450,000 each and represent a far greater risk to organizations than fire or other types of hazard, with estimates as high as $5 billion per year.[1]

But the majority of threats to an organization's information security aren't primarily from outside hackers, rather from an organization's own employees. In the latest Ernst & Young Information Security Survey,[2] 54% of the respondents indicated an information security-related financial loss in the last two years, although of this number, only 10% were able (or willing) to quantify the loss.

Nonetheless, it's worth noting that "twenty respondents indicated loss in excess of $1 million." Malicious, external acts comprised less than 15% of the total, while malicious, internal acts comprised almost 20%. Threats to security can be unintentional as well as deliberate, however, and in fact, Ernst & Young's survey reported "inadvertent error" as the largest single contributor to loss.

In this paper, NAC looks at security in the distributed enterprise-wide computing environment. Basic information security requirements, organizational issues, and technical concepts from a high level are first presented in NAC's Security Framework. The focus of the technical discussion is on the areas of authentication and authorization; authentication includes authenticating user identity as well as validating the source of information.

With an understanding of some of the technology available, NAC then uses a scenario to illustrate one of the problems created by the many authentication mechanisms in the workplace today, that of multiple logons and passwords. Because the mechanisms that exist aren't interoperable,[3] they hamper user access to data; increase the potential for security breaches; increase administrative costs; and ultimately impair the organization's ability to meet its objectives.

[1] Deborah Russell and G.T. Gangemi, Sr. *Computer Security Basics.* O'Reilly & Associates, Inc., 1991.

[2] Ernst & Young LLP. Chicago, Illinois. Third Annual Information Security Survey. Conducted between August and September, 1995. Suvey questionairre sent to 13,000 *Information Week* subscribers in North America. Over 1,300 respondents, 85% of whom were either IS heads or CIOs, direct reports of CIO, or heads of information security.

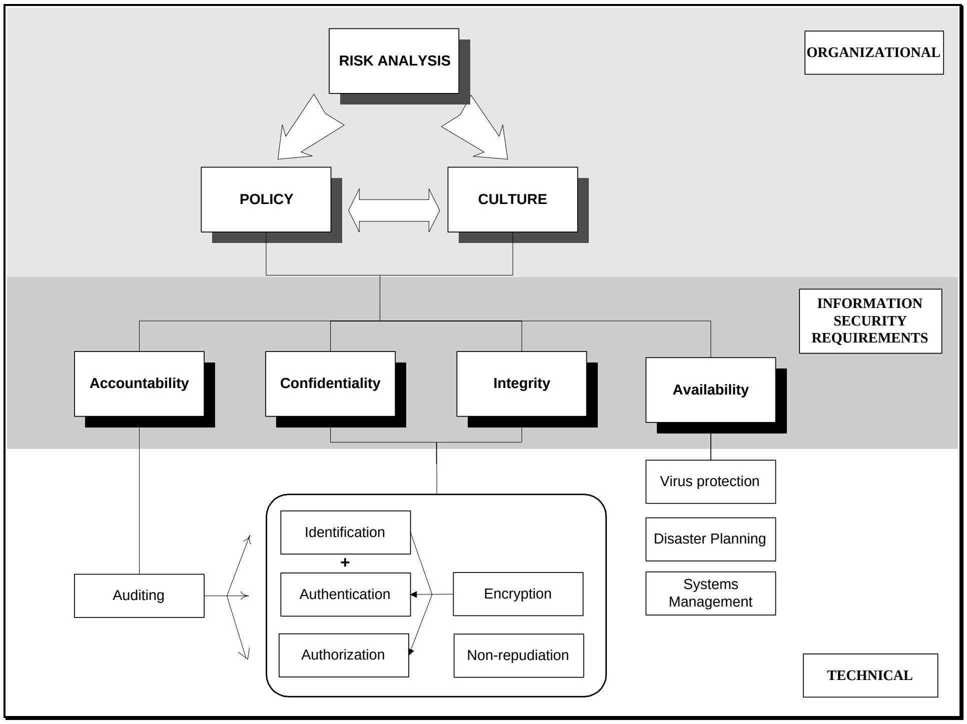
[3] The Network Applications Consortium has been evangelizing the benefits of interoperability since 1992, encouraging vendors to develop applications and common services using standard interfaces. For a summary of NAC's position statement and definition of interoperability, see *Appendix A.*

Enterprise-wide Security Framework

Security policy governs how an organization manages, protects, and distributes sensitive information. A key implication of the NAC Security Framework is that an organization's culture plays an important role in developing any security policy.

Information security encompasses many disciplines, including computer science, cryptography, information systems management, as well as corporate, military, and civil law. To present a coherent viewpoint and simplify discussion, NAC examined several security taxonomies, including the *Department of Defense Trusted Computer System Evaluation Criteria*, the OSI security architecture, and META Group's MESA (META Enterprise Security Architecture), before deriving the Security Framework below. The Framework is a mechanism for guiding the discussion, not a technical representation.

Figure 2. The NAC Security Framework



Security policy provides for the confidentiality, integrity, and availability of information with a measure of assurance that the mechanisms used are working (accountability).

The NAC Security Framework distinguishes between organizational and technical areas. The organizational elements — Risk Analysis, Policy, and Culture — determine how an organization will view information security requirements. That is, depending upon the type of organization or the particular context, one requirement may be more important than another.

For instance, in a top-secret government or military endeavor, confidentiality may be the most important requirement — more important than, say, availability at a given moment in time. In addition, depending upon the sensitivity of the information, systems may be required to meet specific criteria, such as a particular security rating as determined by the US *Department of Defense Trusted Computer System Evaluation Criteria* (TCSEC, or the “Orange Book”) rating level, or the *European Information Technology Security Evaluation Criteria* (ITSEC)[4].

As an example, highly sensitive information in either the public or private sector may require the mandatory access control mechanisms imposed by a B1-evaluated product, as opposed to the discretionary access control mechanisms of a C2 or C1 class product. (See *Authorization* later in this paper for more information about access control mechanisms.)

The differences between the ratings is beyond the scope of this paper, but it’s important to note that the specific processes and mechanisms of the Technical section of the NAC Security Framework must be implemented according to the specific requirements of a given organization.

Organizational Concepts

Risk analysis is the starting point for developing a security policy, and must take several factors into account, including the value of the information; the potential value of the information to those outside the organization; the cost to the organization for the loss or damage of the information; and the cost to secure the information.

Basically, the cost to protect the information shouldn’t exceed the value of the information to the organization. Information must also be evaluated relative to perceived threats both inside and outside the organization; the information may be of great value to the organization but of little value to anyone outside the organization; securing information that falls into this category might cost less because of a lower risk associated with loss of the information. Information assets throughout the organization should be evaluated, taking into account all such factors before attempting to develop a security policy.

An *Enterprise-wide Security Policy* is the set of overarching guiding principles that drives all security-related decisions; it is the set of rules and practices that

[4] Common European security criteria developed by Germany, the UK, France, and the Netherlands.

governs how an organization manages, protects, and distributes sensitive information. The security policy should be published, distributed, discussed, and promoted throughout the organization.

Policies about information ownership and use should be clearly articulated, including the extent to which employees are at liberty to use the tools or the information for their own purposes, if any. Privacy issues should also be directly and coherently addressed in the policy: What happens when personal e-mail messages end up in the wrong hands? How much access to the data on an employee's workstation does management have? How will the organization handle potential freedom of speech conflicts and censorship issues?

Clearly, a wide range of issues must be dealt with by a security policy, and it's critical to get buy-in on an organization-wide basis before publishing a document, which is why *culture* is a cornerstone of the NAC Security Framework: In order to be effective, a comprehensive security policy must be incorporated as an attitude shared by users, administrators, middle- and senior-level management. And in order for "security" to be a shared cultural value, people must have a stake in developing the policy, and believe that it's in their interests to keep the organization's information secure. This can only be achieved when security policy is created with active participation and representation from throughout the organization.

Information Security Requirements

An overarching security policy has three primary goals: to ensure the availability[5], integrity, and confidentiality of an organization's information assets. A fourth requirement, assurance, ensures that the measures taken to provide confidentiality, integrity, and availability include a mechanism — such as auditing logs — to prove that the mechanisms are working. These requirements are stated briefly below.

CONFIDENTIALITY

An organization's information assets must remain confidential. Information should not be disclosed to anyone, whether inside or outside the organization, who is not authorized to access it.

[5] Information *availability* encompasses policies and processes regarding data backup; archival procedures; data destruction; virus detection, elimination, and prevention; disaster recovery; load balancing; and many other activities. These issues, while important, are beyond the scope of this paper.

INTEGRITY

An organization’s information assets must maintain their integrity. The system must not corrupt information or allow any unauthorized malicious or accidental changes to it. Information integrity also encompasses communications integrity — ensuring that network communications are transmitted accurately and are not forged or modified during transmission.

AVAILABILITY

An organization’s information assets must be readily available to those people, processes, or agents (whether internal or external to the organization) that are authorized to use them. The opposite of availability is referred to as “denial of service,” when authorized system users aren’t able to get the resources because of system failures or deficiencies. “Denial of service” attacks are those that focus on occupying system resources so that this situation occurs — flooding an e-mail gateway with traffic until it crashes or occupying a file server with a batch-file that executes an endless loop of commands would be examples of such attacks.

ASSURANCE

An organization must have a means of ensuring that the mechanisms implemented to ensure the confidentiality, integrity, and availability of information assets are working. Auditing facilities, systems management tools, logs, alarms, and a variety of tools and procedures provide this measure of accountability. In the NAC Security Framework it’s presumed that each of the mechanisms that provides confidentiality and integrity includes an auditing capability. Auditing ensures that if security is compromised, it can be traced to the source.

* * *

Keeping information confidential and maintaining its integrity is the focus of the remainder of this paper.

Functional Processes

The enterprise computing environment is heterogeneous, composed of network operating systems, distributed client-server applications, databases, and a range of network services, such as file and print, directory, and messaging, to name a few.

Two basic processes which ensure that only the appropriate users get onto the network, look at information in a database, open files on a network drive, access a mailbox in their name, and so on, include authentication and authorization.

Authentication is a process which occurs in three different security contexts:

- verifying user identity
- verifying the origin of a message
- verifying message content

Verifying message origin and content are discussed below under "*Encryption.*"

☞ Throughout the discussion the word "***user***" refers to any client process, whether initiated by a human user[6] or non-human processes, programs, or agents.

☞ Throughout this discussion, the word "***message***" refers to network packets, e-mail messages, inter-process communications, remote procedure calls — computer-related communications of any kind and at all layers of the OSI protocol model.

IDENTIFICATION AND AUTHENTICATION

Verifying user identity prior to providing access is typically referred to as *Identification and Authentication,* or *I&A.* It's a two-step process that validates a user's purported identity (subject) prior to providing that user access to a resource (object). For example, when users wish to connect to a client-server database application, they enter a user name or ID, in conjunction with a password.

Identification and authentication processes can rely on combinations of one or more of the following:

- User logon plus password
- User logon plus third-party generated password or token (for example, Security Dynamics ACE/Server and SecurID token-card combination, or a cryptographic authentication token such as that created by Kerberos or another cryptographic authentication protocols)
- Biometric-based authentication (fingerprints, handprints, voice recognition, retina scans)

[6] A user can also be thought of as a *subject* that is acting upon network *objects,* objects being files, directories, devices, sockets, and messages.

Each of the techniques listed is increasingly more secure. For example, a user logon and password combination, the most common authentication method, can be enhanced by the addition of a token-card mechanism. This adds another factor — the token-card — to the password: in order to prove identity, the user must not only know something (the password), he must also have something in his possession, specifically, the token-card. A two-factor authentication mechanism is stronger than a single factor mechanism, that is, a password alone.

A third way of proving identity is through biological information that is unique to each individual, such as the fingers, voice, or eye. Biometric-based authentication have historically been too costly for anything but environments demanding the highest degree of security, but these are coming down in price and being adapted for client-server environments. Desktop fingerprint-scanning devices with PC interfaces specifically for use with distributed computing environments are on the market.

In a distributed, client-server environment, the identification and authentication process occurs at several stages. First, a user must logon to the network, and then logon and provide a password (or other mechanism) to access e-mail, database applications, groupware, and so forth.

Authentication Services

Rather than authenticate user identity on an application-by-application basis, an authentication service provides a focal point for authentication, ideally across all applications. An authentication service has a database of user accounts, passwords, and information about services. When a user logs on, the request is sent to the authentication service, which issues a ticket or some other such “credential” after authenticating the user. The credential then authenticates the user to all services and applications for a specific period of time.

Kerberos, a cryptographic authentication protocol that was devised as part of MIT’s project Athena[7], is a widely used authentication service. The protocol has been implemented by several vendors, including CyberSafe, and has also been adopted by the OSF for its DCE, although the DCE version of Kerberos is at this point not compatible with MIT Kerberos; implementations include both versions 4 and 5.

[7] An experimental distributed computing environment begun in the early 1980s at MIT in conjunction with Digital Equipment and IBM.

Other authentication services available include various vendor products, such as Bull Access Master Service, ICL/Access Manager, and SESAME. An authentication framework, X.509, is specified as part of the X.500 directory service recommendation. The NetWare 4.x directory service includes a ticket-based authentication function that is conceptually similar to Kerberos.

The table below lists some of the identification and authentication mechanisms in use today. The list below is by no means complete, but is merely a representative example of the situation in the industry today.

Table 1. Identification and Authentication Mechanisms

Product	**Identification & Authentication**
Network	
Banyan VINES	VINES Security Service (formerly Vanguard)
H-P Unix	HP/DCE Logon
IBM/SNA	RACF
Novell NetWare	NWDSLogin, NWDSAuthenticate
OSF/DCE	DCE/Kerberos
Unix	Unix Logon
Windows NTS	NT Logon + LANMAN Password
Database	
Oracle Oracle7	O3LOGON (or optionally, subsystem OS, Kerberos, DCE/Kerberos, SESAME)
IBM DB2	Subsystem + DB2

AUTHORIZATION

After a user's identity is validated, he or she can access the network object within the parameters specified for that user via the network object's *authorization*, or *access control*, mechanism. In the case of file share or file service, the authorization mechanism is referred to as an *access control list (ACL)* or *access rights list (ARL);* the list contains a list of users that can access the file share and specifies the level of access that they each have; for example, *Read Only, Write, Execute, Delete*, or a combination of whatever rights the file share supports.

The two major types of access control mechanisms, discretionary access controls and mandatory access controls, offer different degrees of security. *Discretionary access controls* function at the discretion of the person who owns the entity. With discretionary access controls, security can be circumvented, albeit indirectly: a user who has rights to an object can transfer those rights to a third entity, who by rights may not necessarily have access. For example, if User A has rights to File X on a file share but User B doesn't have rights, User A could conceivably reset the rights or copy the File X to a diskette and give it to User B.

On the other hand, mandatory access controls enforce security based on a system of labels and clearance levels. Using the example above, in a mandatory access control environment, User A and User B would each have security clearance levels and File X would have a security label — Top Secret, Secret, Confidential, Unclassified, for example — attached to it. Presuming User A's clearance allows him to access File X but User B's security clearance isn't high enough, even if User A copies File X to disk and hands it to User B, User B won't be able to access File X.

Supporting Mechanisms

ENCRYPTION

Encryption is the process of encoding material (text, files, databases, network packets, passwords, for example) so that is meaningless until it is decrypted. Encryption techniques are used at various points in the distributed computing environment to ensure that information will remain confidential, even if it falls into the wrong hands.

For example, an application might encrypt all data that traverses the network, so that if any packets are intercepted, they won't give away any information. Passwords should be encrypted before they're stored in the password file, whether the password file is on a stand-alone computer or a network server. When users enter their logon IDs and passwords to access an application, the IDs and passwords should ideally both be encrypted before being sent over the network.

The encryption process involves using an algorithm and a *key* (a unique bit string as seen below) on the material in question.

Figure 3. Sample of a Key

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.5
mQBNAjFyflAAAAECAJUMDBWXr2aHJELdDiNNO2rNDTIrYQOgAKa1nTAL3LQjXEET
E6iLNFUKBAgksV3FEXHbkA42V3ihVPx2notDk+EABRG0IUtlbGxpIFdpc2V0aCA8
a3dpc2V0aELarry29rZWQubmV0Pg==
=68Kt
-----END PGP PUBLIC KEY BLOCK-----
```

For example, a file containing only the words "*NAC evangelizes the benefits of interoperability*" (called the *plaintext*), was encrypted[8] with the key shown above. After encryption, the file contained the *ciphertext* shown below:

Figure 4. Sample of Encrypted Sentence (Ciphertext)

```
-----BEGIN PGP MESSAGE-----
Version: 2.5

hEwCVPx2notDk+EBAgCTZB1R59uTaf93B5FsXJ3DYNXu3JrmVT5WSofTlM0IhFK7
5lec5yww+45BVHT1z1UPxdSIdATzlx3oMnYU/l10pgAAAFfWOGolGf0PR/dmuogi
p4SdmSzsR0qugHqPQDvRVoaQ+p0r9jtR3AQ2gZxwivv1+WZkdK5fkU1cU+xGLzbC
KE324xoTyJMWm65GJWj18iZYSY5GMCoCmwg=
=5RBW
```

[8] Encryption was performed using the PGP application. PGP is free encryption software (with a history marked by controversy) that uses the RSA algorithm to encrypt files and provide many of the functions, such as digital signatures, needed to meet other security requirements such as integrity and non-repudiation.

```
-----END PGP MESSAGE-----
```

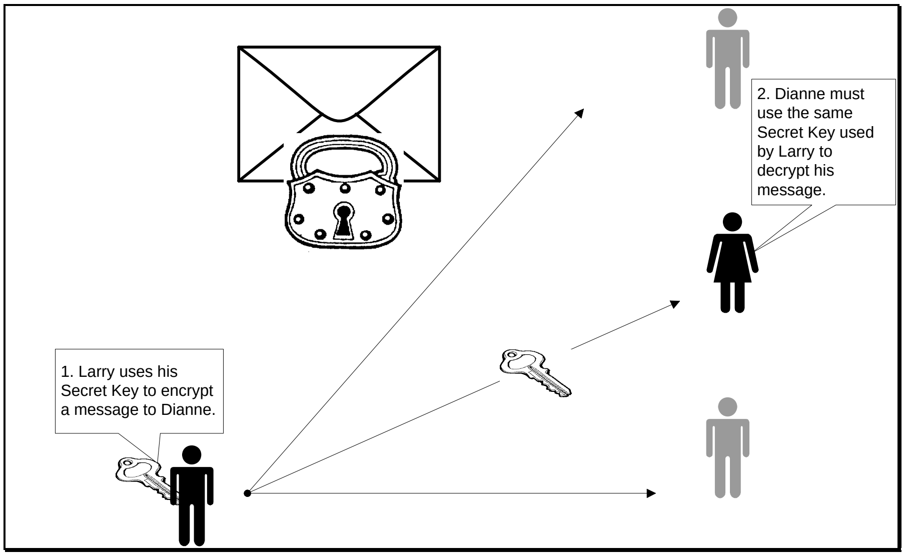
If the packets that comprised this message were intercepted during the course of being transmitted over a network, or if the encrypted file itself were found on a hard-disk drive, neither would be very meaningful to the interceptor without the proper key and the encryption algorithm.[9]

Encryption mechanisms use either public-key technology, secret-key technology, or a combination of both types. Each is described briefly below.

Secret key

In secret-key encryption, two users share a "secret," that being a key. (Although some texts refer to this method as "private key," to avoid confusion it's best to refer to this method as "secret key" to distinguish it from the "private key" used in public-key systems.)

Figure 5. Secret-key Encryption



2. Dianne must use the same Secret Key used by Larry to decrypt his message.

1. Larry uses his Secret Key to encrypt a message to Dianne.

In secret-key, or "symmetric" systems, the same key is used for encryption and decryption. The DES (Data Encryption Standard) is a secret-key algorithm that's been the U.S. standard for non-classified government and sensitive private sector information since 1977.

[9] Of course, an interceptor might *replay* the encrypted logon session and attempt to gain access that way. But there are ways to prevent playback attacks from succeeding beyond a single session.

The DES standard is documented in FIPS 46 and adopted by ANSI as X3.92. The algorithm was originally designed by IBM, but when adopted by NIST it was modified (the key length was shortened), evidently with influence by the NSA. The banking and financial services industry relies heavily on DES, and DES is the algorithm at the heart of Kerberos.

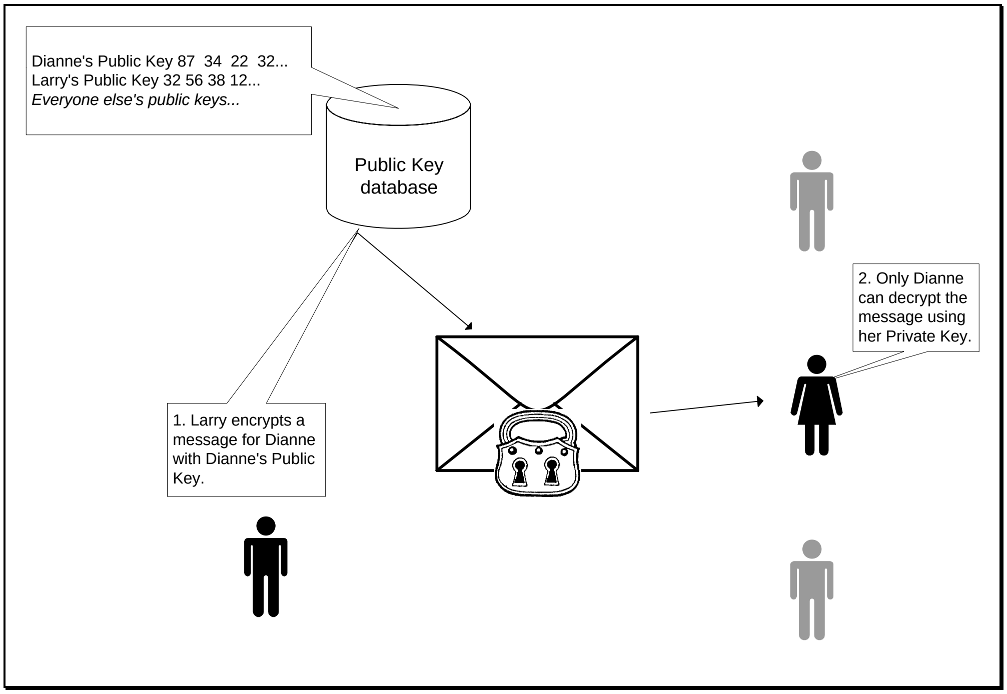
DES breaks up the contents of a file or message into 64-bit blocks and performs a series of substitutions and permutations on each block using the key; the end result is that the contents are scrambled and meaningless until the same key is used again to reverse the process. Although the length of the DES key has been criticized in recent years — cryptographic experts believe that it's too short — its 56-bit length provides 70 quadrillion possible keys.

Since the same key must be used to decrypt communications as is used to encrypt them, a couple of problems arise. For example, how does one securely get a secret — the key — to someone in advance of sharing a secret? And secondly, how does one securely do this with everyone with whom one wants to communicate? Thus, distribution of secret keys can be a problem. In the example above, Larry would need to distribute a different secret key to every with whom he wishes to communicate so that they could decrypt his messages.

PUBLIC KEY

Public-key, or “asymmetric” systems, use two different mathematically related keys; whichever key is used to encrypt, the other key is used to decrypt. The “public key” can be (and must, in order for this system to work) made public. The companion key is called the “private key,” and the only person (or network entity) who can have access to it is the owner. It doesn’t matter which key is the public one and which one is the private one, as long as one remains private.

Figure 6. Public-key Encryption


Dianne's Public Key 87 34 22 32...
Larry's Public Key 32 56 38 12...
Everyone else's public keys...
Public Key database
2. Only Dianne can decrypt the message using her Private Key.
1. Larry encrypts a message for Dianne with Dianne's Public Key.

Named for its inventors, Ron Rivest, Adi Shamir, Leonard Adelman, RSA is a de-facto public-key system in wide use today. For example, RSA encryption is at the heart of the Visa-MasterCard SET (Secure Electronic Transaction) specification. A major benefit of public key technology is that it eliminates the secret-key distribution problem, that is, the need to distribute keys in an absolutely secure, secret way.

Another key management issue emerges, however, and that is ensuring that a particular public key is indeed the public key of the person or entity that it attests to be, and not a public key created by an impostor. To provide authenticity, public key systems must rely on an infrastructure of “Certificates” and a “Certificate Authority.” The hierarchy and infrastructure required for public-key management, or the “public-key infrastructure” (PKI), is a primary requirement for the success of public-key systems.

A spin-off of RSA Data Security, Inc., Verisign, (with financial backing from Mitsubishi Corporation, Visa International, and Security Dynamics, among others) has begun to position itself as a Certificate Authority for electronic commerce; it was spun-off from RSA specifically for this purpose. Verisign provides certificates (which it calls “Digital IDs”) to corporations and individuals. The company also sells hardware for third-parties to use in-house or with the public to generate certificates. Anyone can buy this technology — Verisign’s CIS (Certificate Issuing System) — to create digital certificates for in-house or external use.

That said, an infrastructure still doesn’t exist. The US Post Office has announced its intentions to be *the* Certificate Authority, and expects to begin a pilot in the summer of ’96.

RSA Data Security, Inc. provides software development tool-kits built around RSA’s PKCS (Public Key Cryptography Standards). PKCS encompasses several public-key algorithms and related functions, including PKCS #1 (RSA Encryption Standard); #3 (Diffie-Hellman Key-Agreement Standard); PKCS #5 (Password-based Encryption Standard), and several others.

Public-key and secret-key techniques are not mutually exclusive, and in fact are typically combined. See the example below for “Digital Envelopes.”

CRYPTOGRAPHIC AUTHENTICATION TECHNIQUES

In addition to ensuring secrecy by encrypting data for transmission or storage, cryptographic systems can provide a number of security-related functions in a distributed computing environment, such as verifying user identity, verifying data integrity, and verifying data origin, in addition to ensuring confidentiality. Cryptographic keys can be used to verify user identity.

For example, in a public-key system, since only the user should have the private key, a server or other process can verify the identity of the user by using the public key to encrypt a message and send it to the user. If the user can decrypt the message and send back an appropriate response, his or her identity is then confirmed, and communications can proceed. This is basically what cryptographic authentication protocols are about. As mentioned above, Kerberos performs this type of service. Some other types of authentication, beyond user identification, are discussed below.

DIGITAL ENVELOPES ENSURE CONFIDENTIALITY

Both public-key and secret-key can be combined to create a “digital envelope” which ensures the confidentiality of the contents of a message. Say Larry wants to send an encrypted message to Dianne. Larry will:

- Encrypt the message with DES, using a randomly selected DES key
- Get Dianne’s public key from the public database
- Encrypt the DES key with Dianne’s public key
- Send the message to Dianne

When Dianne receives the message, she uses her private key to decrypt the envelope and then uses the packaged DES key to decrypt the message. If it’s not for her — if Larry used Dion’s public key by mistake, instead of Dianne’s public key, for example — then Dianne can’t decrypt the message.

DIGITAL SIGNATURES AUTHENTICATE MESSAGE ORIGIN

A *digital signature* authenticates, or verifies the origin of a message. Say Larry is sending an email message to Dianne. (Again, the “message” can be any client-server communication).

- The message gets run through a message digest[10] algorithm to create a 128-bit number.
- The message digest is then encrypted with Larry’s private key to create Larry’s digital signature.
- The digital signature and original message are sent to Dianne.

When Dianne receives the message, she uses Larry’s public key to verify his signature. If the signature verifies correctly, Dianne proves two things: that the message hasn’t been corrupted, and that Larry sent the message. Notice that in this scenario, the message itself is not encrypted, so confidentiality is not provided.

DIGITAL SIGNATURES PROVIDE NON-REPUDIATION

[10] A message digest, also called a hash function or hash algorithm, generates a unique fixed-length value from an input of any length. The point of this mechanism in a security context is that it provides a “digital fingerprint” (it’s unique to the original) of an input which cannot be reversed; that is, such algorithms can prove integrity without disclosing the contents of the original.

In the example above, anybody with Larry’s public key could also prove that the message came from Larry by verifying the signature with Larry’s public key. This last feature is called *non-repudiation:* Larry cannot deny that he sent the message because the digital signature was created using his private key, which only he has. And if Larry claims that the message was altered, the message digest can be used to prove or disprove it either way.

* * *

With an overview of the technologies available, we turn to a real-world scenario from the enterprise-wide computing environment to see how these mechanisms are implemented, from an end-user’s perspective. Vendors have implemented a variety of authentication mechanisms and technologies as well as proprietary solutions in their products. How do all these mechanisms “play out” in terms of user accessibility to information, administration, and the like? As the following scenario describes, not very well.

Enterprise-wide Security in the Real World

Security implementations are numerous. Each application has its own mechanisms and methods for configuring and enforcing security policies, resulting in enormous expense to organizations in terms of lost productivity, increased administrative expense, and user frustration. Security mechanisms have created such a burden to end-users that they frequently find ways to circumvent them.

On a typical Monday morning in the marketing department, the regional manager of the North America division arrives with only minutes to spare before the executive committee's quarterly planning session. The manager turns on his workstation so he can print a copy of the detailed report that the field office posted the night before in the collaboration database.

As the computer boots up, the manager pulls out a sheet of paper from his top desk drawer that contains a list of logon IDs and passwords. As always, his morning begins with a "logon ritual," starting with the power-on password for his Compaq DeskPro, his Windows NTS user ID and password for the corporate LAN, then his logon ID and password for the SNA environment, then a logon and password to the Oracle database, and finally, a logon and password to the Lotus Notes discussion database.

He stops short when a prompt warns him that his UserID for the collaboration database — containing the report he needs for the meeting — is invalid. He double-checks his password list, which contains close to a dozen UserIDs and passwords, but he doesn't see anything wrong. Blood-pressure rising, he calls the Help Desk to find out what's wrong with the collaboration database.

The Help Desk has an arsenal of administration tools at its disposal: after all, the technical support folks are responsible for Windows NT Servers, Oracle databases, Sybase databases, PROFS e-mail system, LAN e-mail systems, and groupware applications, to name just a few. It takes close to 10 minutes for the technical analyst to determine precisely which system the marketing manager is having a problem with, but eventually she's able to say confidently (and with the usual sigh of exasperation) "there's nothing wrong with the collaboration database — are you sure you're using the right password?"

Just then, the director's assistant walks in and slides a copy of the report in question under his nose. Hanging up the phone, he asks her how she got a copy of the report. Turns out the assistant had worked late the night before and decided to do the manager a favor. Since she knows where he keeps his password list (as does the temp who occasionally fills-in when she's on vacation) she was able to logon, download, and print his confidential report almost without any problem.

"But by the way," she says, "your password expired so I had to reset it — it's now *BillG*..."

Relieved to have his report in hand, the marketing manager gets to the meeting with just minutes to spare, making a mental note to give his assistant a raise for showing such initiative…

As he begins to scan the report, he soon discovers that the numbers make no sense. For starters, several offices that had been closed last quarter list record-high sales, while several offices that had been consolidated into one show no sales at all. Confused and concerned, he pulls his head away from the report and looks around the conference room.

He then notices that the lead security analyst has been sitting at the table as well, explaining to the committee that "several of the company's key reporting systems contain corrupted data," which the marketing manager quickly translates into the more meaningful "our information is garbage." Evidently a database system that was installed a month ago inadvertently gave super-user access to 87 people in telemarketing. Somehow — the security analyst isn't sure how it happened because there's no audit trail... the logging capability of the new system wasn't configured according to the standard — what should have been simple queries were posted to the accounts in the new system. Unfortunately, the data from the new system fed into five other reporting systems — including the marketing department's database used to build the report.

*** * ***

The Issues

This scenario highlights several important issues. At the same time they attempt to secure information from outside access, enterprise-wide security mechanisms may result in:

- compromised security
- lost productivity
- increased administration expense

COMPROMISED SECURITY

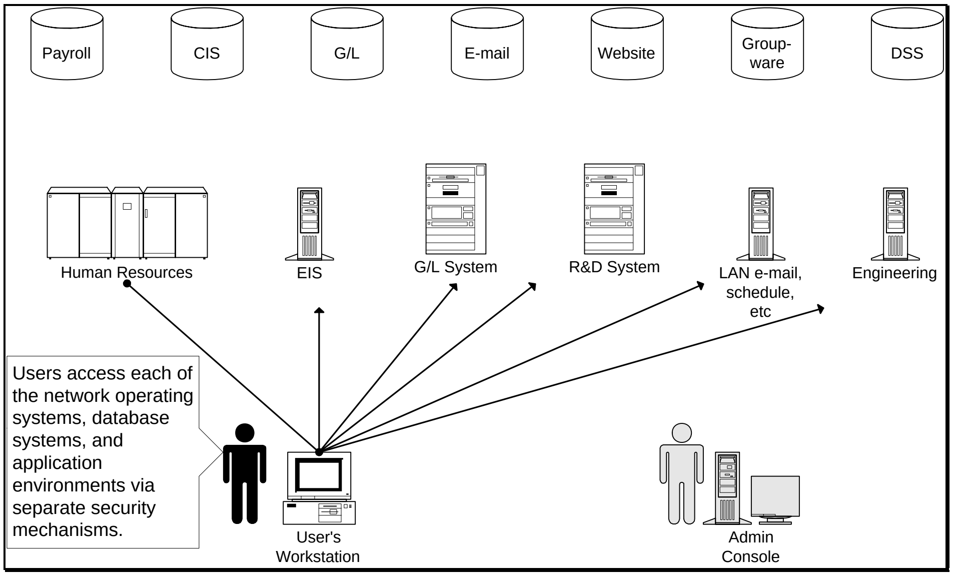
Multiple logons and passwords increase security risks. The situation described in the scenario — a user who keeps a list of all his logons and passwords right at his desk — is not uncommon. Multiple logons and passwords interfere with productivity by requiring users to remember different ones for every system, so users look for ways to simplify the process, such as writing them on paper or tacking Post-It notes with passwords to the computer monitor.

In so doing, users circumvent the very security these measures wish to impose, as seen in the scenario: the authentication process became meaningless the moment one individual became aware of another person's logon ID and password.

LOST PRODUCTIVITY

Multiple logons and passwords impede productivity. As shown in the figure below and as described earlier in the scenario, every different network, database, and network or enterprise-wide application typically has its own security mechanisms with its own attendant logon and password.

Figure 7. Multiple Applications, Multiple Logons


Payroll
CIS
G/L
E-mail
Website
Group-ware
DSS
Human Resources
EIS
G/L System
R&D System
LAN e-mail, schedule, etc
Engineering
Users access each of the network operating systems, database systems, and application environments via separate security mechanisms.
User's Workstation
Admin Console

Given today's client-server, distributed computing environment with a full complement of databases, e-mail systems, and legacy applications, logons per user are numerous. Winn Schwartau, a Florida-based security expert, reports of computer network managers with as many as 48 passwords.[11]

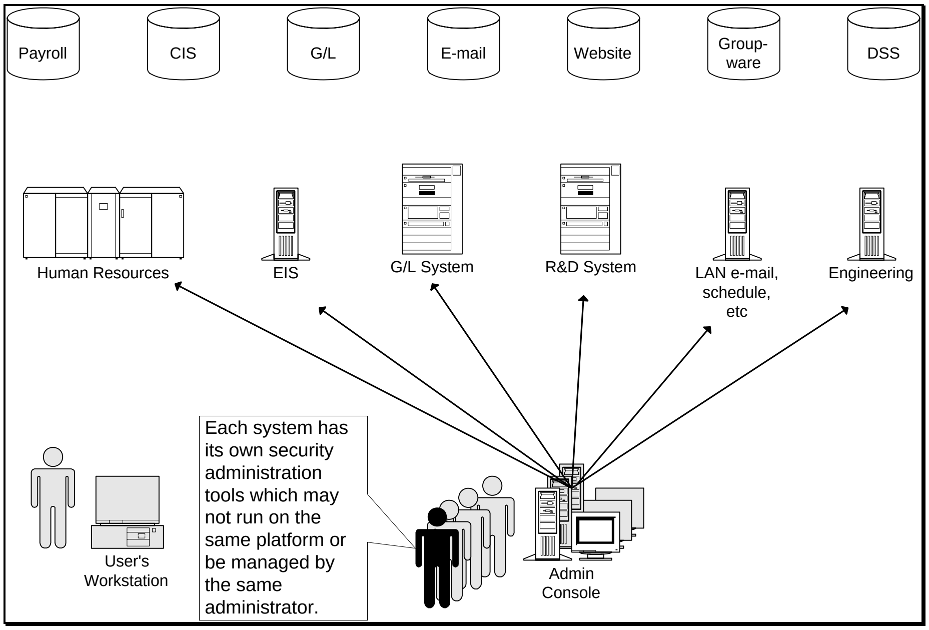
[11] "Halt! Who Goes There? Electronic security has led to growing insecurity for those who can't remember passwords." William M. Bulkeley. *Wall Street Journal*. April 23, 1995.

INCREASED ADMINISTRATION COSTS

In addition to imposing a burden on users to remember a variety of logons and passwords, multiple security mechanisms imposes an enormous burden on an organization in terms of administration. Every system that has its own means of identifying, authenticating, and authorizing users must be configured appropriately, typically via independent mechanisms. That is, one system's administration tool doesn't fit all systems.

Administrators are responsible for managing user logons, passwords, profiles, and access lists. Administrators must keep up with all the changes logons, passwords, profiles, and access lists in organizations where change is the only constant. Keeping up with adds, moves, changes, according to Gartner Group and others, is the most costly aspect of network computing.

Figure 8. Multiple Systems to Administer



Each system has its own security administration tools which may not run on the same platform or be managed by the same administrator.

The issue is not whether information security is important, but rather that the raft of security mechanisms available don't interoperate with each other or with existing common network services. The next scenario describes NAC's vision of interoperable security services.

Enterprise-wide Security: NAC’s Vision

Security mechanisms function behind-the-scenes, virtually invisible to end-users. Everyone in the organization has access to all the information they need to function as an empowered participant in the business. At the same time, threats to the information are kept at bay in a cost-effective manner.

On a typical Monday morning in the marketing department, the regional manager of the North America division arrives with only minutes to spare before the executive committee’s quarterly planning session. The manager turns on his workstation so he can print a copy of the detailed report that the field office posted the night before in the collaboration database.

As the computer boots up, the manager enters his UserID and password. Behind the scenes, his encrypted UserID and password are sent to the security service, which interacts with the directory service to validate the marketing manager by comparing credentials held in the directory service. Upon validation, the security service generates an encrypted *security ticket* that will provide the manager with entry to all the systems that he is authorized to used during the next four hours.

The expiration limit is based on the manager’s role as configured by the security service. Other roles, such as that specified for an order-entry clerk in the telemarketing department, would have different time, place, and expiration limitations. When the manager accesses the collaboration database, instead of entering another logon and password, his encrypted ticket is used by the collaboration database to validate his identity. He launches the collaboration database application, finds and prints his report.

Meanwhile, elsewhere in the company, the lead security analyst is implementing a new database application. The analyst has a list of all users who are entitled to access the new system and the data that they will need to access. In addition, he has a list of a dozen other applications that will interface with the new one. At his disposal, the analyst has a security policy services application that enables him to configure the breadth of general security policies that might apply to the new application — mandatory or discretionary access controls, authentication techniques and services, password policies (in the event that passwords are used as an authentication technique), session time-outs, intrusion detection, audit record store and retention.

When the marketing manager learns of the new application via an automatically generated e-mail message from the application to the appropriate set of new users, he can access it automatically without any additional logon facility.

Conclusion

Since 1992, NAC has been acting as a catalyst for the development of interoperable network applications with the goal of providing rational, coherent information technology solutions to the enterprise. Generally speaking, information technology managers must choose from a multitude of technical solutions, too few of which interoperate. In the last few years, NAC has examined the areas of messaging and directory services. As is the case with these services, security services also represent an infrastructure-level service that must interoperate with and support services throughout the enterprise.

But as in the case of messaging and directory, security services are characterized by a proliferation of mechanisms, each of which must be purchased, installed, configured, and maintained. One of the key tasks performed by security services is authenticating users prior to enabling access, and it is authentication process that presents a sizable burden in terms of administration and management. Non-interoperable authentication mechanisms also impose a sizable burden on the end-user, who loses ease-of-access to information and must spend inordinate amounts of time logging on.

NAC will explore issues more specific to authentication services in its next paper. The significant issues that begin to emerge include the need for a general security API, common administration tools, and market convergence around a standard security model.

Appendix A. NAC’s Definition of Interoperability[12]

Interoperability enables organizations to mix-and-match components — hardware platforms, operating systems, and applications — without loss of functionality or duplication of services: any client works with any server, and both use common enterprise services.

From a non-technical, business perspective, the concept of *interoperability*” enables an organization to leverage investments made in the “building blocks” that comprise its IT infrastructure. The building blocks are the diverse group of networks, platforms, operating systems, and the applications that run on them. A conceptual view of these components is shown in Figure A below. (Note that this is an oversimplified view; the enterprise environment might include different layers, or the layers might be organized differently. For simplicity’s sake, the example is only a base upon which to build a definition of interoperability.)

Saying that all these components should *interoperate* has shades of meaning that depend upon your vantage point. For example, in an individual client workstation, an application, such as a word processor, must interoperate only with the given operating system; in turn, the operating system must run on the hardware platform in question. In addition, if the workstation must access a service over the network (and is thereby acting as a client), the network layer must be compatible with all these components.

Implementing a single application such as a word processor on a single hardware platform or a particular operating system is one thing. But, as shown in Figure A below, in the process of implementing enterprise applications, one discovers a great number of choices exist at every layer of the model:

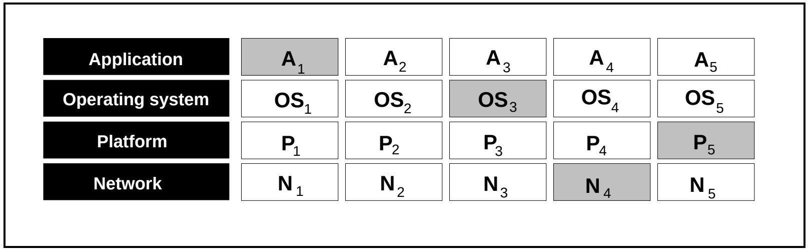
Figure A. Many Choices at Every Layer

Application	A_1	A_2	A_3	A_4	A_5
Operating system	OS_1	OS_2	OS_3	OS_4	OS_5
Platform	P_1	P_2	P_3	P_4	P_5
Network	N_1	N_2	N_3	N_4	N_5

[12] Excerpted from “*Interoperability: A NAC Position Paper*” August 1994.

Interoperability would enable IT managers to choose any component from within any layer without loss of functionality. For example, after selecting any application[13] — A_1 for example — you may also choose OS_3, P_5, and N_4 as the other components of choice with the assurance that all will work together.

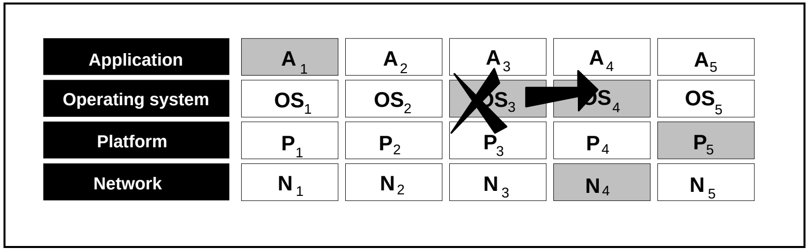
Figure B. Choose Freely Among all Layers



Application	A_1	A_2	A_3	A_4	A_5
Operating system	OS_1	OS_2	OS_3	OS_4	OS_5
Platform	P_1	P_2	P_3	P_4	P_5
Network	N_1	N_2	N_3	N_4	N_5

Interoperability exists when choices can be made from any layer and matched with choices in any other layer. So if a decision were made to move from one operating system to another — for example, from OS_3 to OS_4 as in Figure C below — the operating system layer would be interoperable with the application and platform layers if the change could be made without affecting either of these layers. For example, the e-mail client application would still run on the selected hardware platform and the newly chosen operating system, and the network operating system would still deliver messages.

Figure C. Choices Work with All Choices Across Layers



Application	A_1	A_2	A_3	A_4	A_5
Operating system	OS_1	OS_2	OS_3	OS_4	OS_5
Platform	P_1	P_2	P_3	P_4	P_5
Network	N_1	N_2	N_3	N_4	N_5

Moving beyond a single workstation to the enterprise at large, interoperability would enable an organization to implement a given hardware platform and e-mail client application for one department, and select a completely different hardware platform and different e-mail client application for another department with complete assurance that both e-mail clients could communicate with each other.

However, being able to mix-and-match components and applications isn't enough. Any enterprise application requires certain basic functionality, including the ability to secure the application so that only the appropriate parties can access

[13] Note that the *application* can be either the client or server application code.

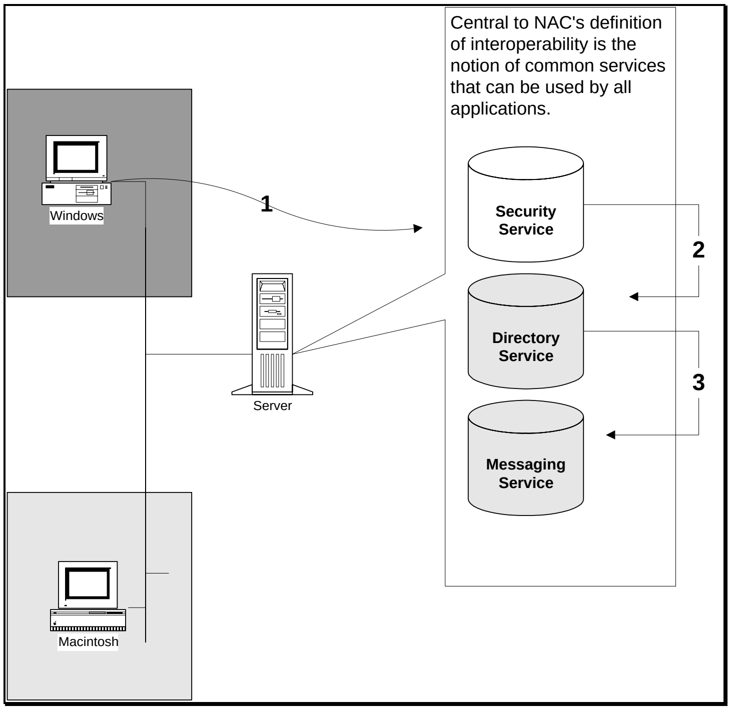
it, the ability to locate the application on the network, and the ability to transmit information from the starting point to the appropriate endpoint. These *common services* — security, directory, and messaging, to name a few — are required by many enterprise applications.

Therefore, in order for the environment to be truly interoperable, organizations must be able not only to mix-and-match building blocks on any level, but the basic functions — common services — that provide the foundation for the enterprise environment should be seamlessly used by all applications. Not doing so means duplicate services, schemes to connect and synchronize multiple systems, added administration, excessive training and support costs, and so on.

Thus, NAC's concept of *interoperability* has two dimensions. Interoperability provides IT managers with:

- the ability to mix-and-match the building-block components and applications that comprise the IT infrastructure
- the use of a common set of service functions shared by all applications

Figure D. Enterprise Computing Requires a Set of Common Services


Central to NAC's definition of interoperability is the notion of common services that can be used by all applications.
Windows
1
Security Service
2
Directory Service
3
Server
Messaging Service
Macintosh

The common services interoperate with each other as well as with the network applications that need them. In the figure above:
❶ the user logs onto the enterprise network;
❷ the security service and directory service interact to validate the user by comparing credentials held in the directory service; and
❸ the message and directory services work together to resolve addresses for messages transmitted by users (and processes).

Appendix B. Glossary

acceptable use policy
An official policy statement which describes the type of activities that may occur on a network.

access control list
An authorization mechanism, specifically a list attached to a particular network resource, which specifies the users (human, agent, process) that can access the resource as well as the level of access that they have. For example, an ACL attached to a directory on a file share will list the user accounts. Access levels typically provide either read, write, execute, modify, delete, or create permissions, or combinations of these.

access controls
Mechanisms that specify and oversee the specific ability of one entity to access another. Such mechanisms may include hardware features, software features, operating procedures, management procedures. For example, a bootup password on a desktop computer is a hardware-based access control mechanism while a screen-saver password prompt is an operating system access control mechanism.

access
The ability of one entity in a distributed computing system to view, change, or communicate with another entity.

ACL
Access control list.

active threat
A threat that not only intercepts information (as does a passive threat) but alters it and reinserts it into the information stream.

asymmetric cipher
An asymmetric cipher uses two keys that are mathematically related in such a way that whichever key is used for encryption, the other key is used for decryption.

authentication
A process in which one entity verifies the identity of another entity.

authorization
The process of determining whether a client may use a service, which objects the client is allowed to access, and the type of access allowed for each.

cipher
A cryptographic system in which units of plain text of regular length, usually letters, are arbitrartily transposed or substituted according to a predetermined code; the key to such a system; a message written or transmitted in such a system.

client
An application that initiates a connection to a server. More generally, a computer that requests and receives information from another system through a network.

Clipper chip
A high-speed encryption chip designed by the National Security Agency. A Clipper chip encrypts data with the Skipjack algorithm (a classified algorithm that isn’t published, unlike all other encryption algorithms). Clipper-encrypted communications can be decrypted by law enforcement agencies or others who possess the chip’s secret key.

cryptography
The process of communicating in or deciphering secret writings or ciphers.

cryptosystem
A system whose intent is to disguise a message so that only someone who knows the code can read it. The original message is a plaintext. The disguised message is a ciphertext.

DES
Data Encryption Standard. A data encryption standard developed by the US government based on classified research.

digital signature
Encrypted data appended to the end of a message (or accompanying a binary file) that functions as a signature to attest to the authenticity of the file. Digital signatures are created with a private key and are verified with a public key. If any change is made to the signed file, the digital signature does not verify.

EES
Electronic Escrow Standard. A key escrow standard promulgated by the U.S. government that requires that the encryption keys used by telecommunications equipment be split into two parts and stored at different escrow agencies.

encryption
Any process that converts plaintext into ciphertext.

firewall
A network protection mechanism that selectively prevents or permits traffic between internetworks, monitors access to network services, and provides an audit trail.

GSSAPI
Generic Security Service Application Programming Interface.

IDEA
International Data Encryption Algorithm. A symmetric private-key cryptographic algorithm developed in Switzerland and licensed for non-commercial use in the PGP mail-encryption package. IDEA uses a 128 bit user supplied key to perform a series of nonlinear mathematical transformations on a 64 bit data block.

ITAR
International Traffic in Arms Regulations. ITAR are the regulations covering the exporting of weapons and weapons related technology from the United States. The government treats data encryption as a weapon under the ITAR regulations.

Kerberos
An authentication protocol that defines a series of messages that enable a client to acquire a security ticket. Secret-key based.

key certification
A process by which a public key is certified to belong to a particular individual or organization. Frequently, key certification is done with the digital signature of a trusted individual.

key escrow
Any system for making a copy of an encryption key so that it can be accessed at a later time by authorized individuals.

LEAF
Law Enforcement Access Field. A block of data generated by Clipper chips that contains a copy of the current session key that has been encrypted with the chip’s master encryption key.

non-repudiation
A characteristic provided by means of a digital signature which proves the identity of the originator of a message.

NSA
National Security Agency. The official communications security body of the U.S. government, founded in the early l950s.

passive threat
Intercepts information from a data stream but doesn’t alter it.

PEM
Privacy Enhanced Mail. An Internet standard for sending public-key encrypted mail over Unix systems. In the United States, the most popular implementation is RIPEM. RIPEM uses DES for encrypting the body of the message, and RSA to encrypt the message key.

PGP
Pretty Good Privacy. A public-key encryption system based on RSA encryption that was made widely available — for free — via the Internet. PGP is a strong encryption method, but lacks the support and infrastructure to make it suitable for commercial use.

public key encryption
A security technique that uses two keys: a public key and a private key. The public key is published and is used to encrypt data, while the private key must be known only to its owner. Messages encrypted with the public key can only be decrypted with the associated private key. Conversely, messages encrypted with the private key can only be decrypted with the public key.

RACF
Resource Access Control Facility

RC2, RC4
Proprietary bulk ciphers invented by RSA. RC2 is block cipher and RC4 is a stream cipher.

RIPEM
An implementation of Privacy Enhanced Mail written by Mark Riordan that is based upon RSAREF.

RSA
RSA is a public-key encryption algorithm. (RSA are the initials of the algorithm’s developers, Rivest, Shamir, and Adleman.) .

server
The server is the application entity that responds to requests for connections from clients.

spoofing	Fooling a user or system by imitating the actions of another person or process.
symmetric cipher	A symmetric cipher uses the same key for decryption and encryption. Some examples of symmetric ciphers: IDEA, RC2, RC4, and DES.
X.509	The authentication framework specified in the ITU-T X.500 directory recommendation.

Appendix C. References

Building Internet Firewalls Chapman and Zwicky. 1995. O'Reilly & Associates, Inc. Sebastopol, CA 95472

Computer Security Basics. Russell and Gangemi Sr. 1991. O'Reilly and Associates, Inc., Sebastopol, CA 95472

Department of Defense Trusted Computer System Evaluation Criteria. DoD 5200.28-STD, December 26, l985

Distributed Computing: Implementation and Management Strategies. Raman Khanna, editor. 1994. Prentice Hall. Englewood Cliffs, NJ 07632

Distributed Computing: Implementation and Management Strategies. Khanna, Raman (Editor). PTR Prentice Hall. Englewood Cliffs, New Jersey, 1994.

FIPSPUB 46-2: Data Encryption Standard.

FIPS PUB 48: Guidelines on Evaluation of Techniques for Automated Personal Identification.

FIPS PUB 74: Guidelines for Implementing and Using the NBS Data Encryption Standard.

FIPS PUB 81: DES Modes of Operation.

FIPS PUB 83: Guideline of User Authentication Techniques for Computer Network Access Control.

FIPS PUB 112: Password Usage.

FIPS PUB 113: Computer Data Authentication.

FIPS PUB 171: Key Management Using ANSI X9.17.

FIPS PUB 180: Secure Hash Standard.

FIPS PUB 185: Escrowed Encryption Standard. 2/19/94

FIPS PUB 186: Digital Signature Standard. 5/19/94

FIPS PUB 190: Guideline for the Use of Advanced Authentication Technology Alternatives. 9/28/94.

Firewalls and Internet Security: Repelling the Wiley Hacker. Cheswick and Bellovin. Addison-Wesley Publishing Company. 1994.

Guide to Writing DCE Applications. Shirley, Hu, Magid. O’Reilly and Associates, Inc., Sebastopol, California, 1994.

Network Security. Data and Voice Communications. Simonds. McGraw-Hill. 1996.

Network Security. *Private Communication in a Public World.* Kaufman, Perlman, and Speciner. Prentice Hall. Englewood Cliffs, NJ 07632. 1995.

NIST Special Publication 500-166 “*Computer Viruses and Related Threats*”

NIST Special Publication 500-169 “*Executive Guide to the Protection of Information Resources*.”

NIST Special Publication 500-170 “*Management Guide to the Protection of Information Resources*.”

NIST Special Publication 500-171 “*Computer Users Guide to the Protection of Information Resources*.”

PGP: Pretty Good Privacy. Garfinkel. 1995. O’Reilly and Associates, Inc., Sebastopol, CA 95472

Power Programming with RPC. Bloomer. O’Reilly and Associates, Inc., Sebastopol, California, 1992.

RFC1244 (Informational) “*Site Security Handbook.*” Holbrook, Reynolds. 07/23/1991.

RFC1319 (Informational) “*The MD2 Message-Digest Algorithm*.” Kaliski. 04/16/1992.

RFC1320 (Informational) “*The MD4 Message-Digest Algorithm.*” Rivest, 04/16/1992.

RFC1321 (Informational) "*The MD5 Message-Digest Algorithm*." Rivest, 04/16/1992.

RFC1411. "*Telnet Authentication: Kerberos Version 4.*" D. Borman, Editor. January 1993.

RFC1507 (Informational) "*DASS (Distributed Authentication Security Service).*" C. Kaufman. September 1993.

RFC1508 (Standards Track) "*Generic Security Service Application Program Interface*." Linn. 09/10/1993.

RFC1510 (Standards Track) "*The Kerberos Network Authentication Service (V5).*" Kohl, Neuman. September 1993.

RFC1511 (Informational) "*Common Authentication Technology Overview*." Linn, 09/10/1993.

RFC1636. (Informational) "*Report of IAB Workshop on Security in the Internet Architecture*."R. Braden, D. Clark, S. Crocker, C. Huitema. June 1994.

RFC1704. (Informational) "*On Internet Authentication*." R. Atkinson. October 1994.

RFC1825 (Standards Track) "*Security Architecture for the Internet Protocol*." Atkinson. 08/09/1995.

RFC1862 (Informational) "*Report of the IAB Workshop on Internet Information Infrastructure, October 12-14, 1994*." McCahill, Schwartz, Sollins, Verschuren, Weider. 11/03/1995.

RFC1865 (Informational) "*EDI Meets the Internet: Frequently Asked Questions about Electronic Data Interchange (EDI) on the Internet*." Houser, Griffin, Hage. 01/04/1996.

RFC1898 (Informational) "*CyberCash Credit Card Protocol Version 0.8*." Eastlake, Boesch, Crocker, Yesil. 02/19/1996.

Security Architecture for Open Distributed Systems. Muftic, Patel, Sanders, Colon, Heijnsdijk, Pulkkinen. John Wiley and Sons, England. 1993.