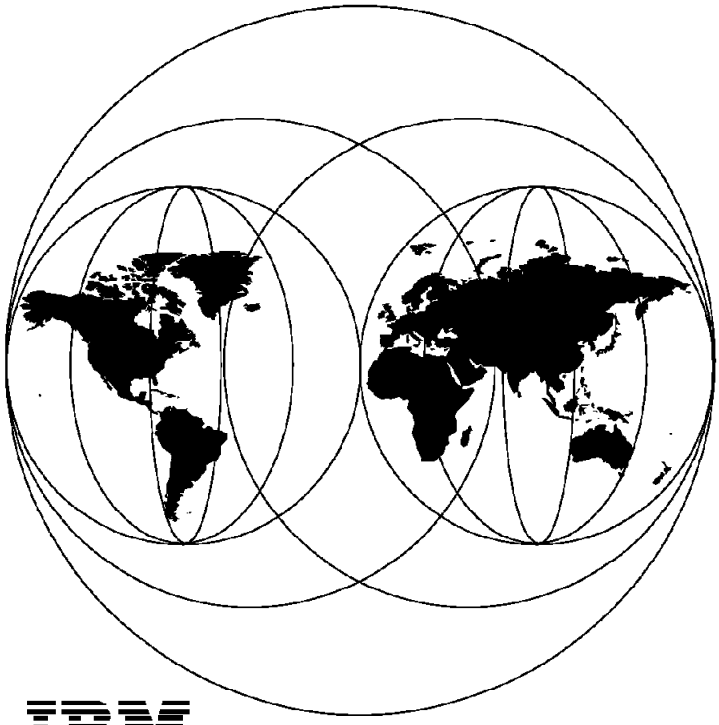
SG24-4568-00

# OS/2 Security Enabling Services

September 1995



IBM

International Technical Support Organization

Boca Raton Center

SG24-4568-00

# OS/2 Security Enabling Services

September 1995

**Take Note!**

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xiii.

**First Edition (September 1995)**

This edition applies to OS/2 Security Enabling Services for use with the OS/2 V2.11 plus a fixpak. There is a security fixpak available for OS/2 V2.11 only.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. JLPC Building 14 Internal Zip 5520
1000 NW 51st Street
Boca Raton, Florida 33431-1328

# Abstract

This document is unique in its detailed coverage of OS/2 security enabling services. It focuses on security enabling services, which are an add-on in OS/2. It provides information about this security enabling services in order to have a basic understanding of how security enabling services works together with Independent Software Vendors (ISV) applications.

This document was written for IBM customers, system engineers, software developers, consultants and anyone else requiring information on OS/2 security. Basic understanding of OS/2, DOS and Windows operating systems is assumed.

(98 pages)

# Contents

# Figures

# Tables

# Special Notices

This publication is intended to help IBM customers, dealers and IBM system engineers to provide an introduction to OS/2 security. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/2 security enabling services. See the PUBLICATIONS section of the IBM Programming Announcement for OS/2 Security Enabling Services for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the

examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX
CUA
Operating System/2
Personal Security
PS/ValuePoint
PS/2
ThinkPad
Workplace Shell
Common User Access
IBM
OS/2
Personal System/2
PS/1
RACF
ValuePoint

The following terms are trademarks of other companies:

Windows is a trademark of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Other trademarks are trademarks of their respective companies.

# Preface

This document is intended to provide an overview of OS/2 security enabling services. It contains information about the security enabling services which have to be installed on top of OS/2. This product is available as a fixpak in OS/2 V2.11 only.

This document is intended for IBM customers and employees requiring an overview, quick reference, and installation guide for security enabling.

## How This Document is Organized

The document is organized as follows:

- Chapter 1, “Information Security”

  This chapter provides an overview about the various aspects of why we need security.

- Chapter 2, “Security Environment”

  This chapter describes the security environment. It starts with a description of how to create an information security policy and how to establish it in the enterprise.

- Chapter 3, “OS/2 Security Enabling Strategy”

  In this chapter we look at the strategy that has been applied to OS/2, and take a brief look at client/server security.

- Chapter 4, “Security Enabling Services”

  In this chapter, we take a close look at the SES components and how they interact with both the OS/2 kernel and the Installable Security Subsystem (ISS).

- Chapter 5, “Installable Security Subsystem”

  This chapter discusses the relationship between SES, ISS and the security-dependent applications that must work together in a secured OS/2 workstation.

## International Technical Support Organization Publications

- *Security Enhancement Solutions for Workstations*, SG24-4569
  Available at a later date.
- *Security Overview of Open Systems Networking*, GG24-3815

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

*International Technical Support Organization Bibliography of Redbooks,* SG24-3070.

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOCAT TXT. This package is updated monthly.

**How to Order ITSO Redbooks**

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and MasterCard are accepted. Outside the USA, customers should contact their local IBM office. Guidance may be obtained by sending a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called BOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

## ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser (such as WebExplorer from the OS/2 3.0 Warp BonusPak) to the following:

```
http://www.redbooks.ibm.com/redbooks
```

IBM employees may access LIST3820s of redbooks as well. Point your web browser to the IBM Redbooks home page:

http://w3.itsc.pok.ibm.com/redbooks/redbooks.html

## Acknowledgments

This project was designed and managed by:

**Lajos Damen**
International Technical Support Organization, Boca Raton Center

The authors of this document are:

**Henning Borchers**
IBM Germany

**Graham Cogle**
IBM UK

**Ramon Gonzalez Compta**
IBM Spain

**Hartmut Schmidt**
IBM Germany

**Reinder Wiersma**
IBM Netherlands

This publication is the result of a residency conducted at the International Technical Support Organization, Boca Raton Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

John Divers
IBM UK

Harry Benas
Security Development, IBM Boca Raton

Bill Coltin
Security Development, IBM Boca Raton

Tony DiDaniele
Security Development, IBM Boca Raton

Mickey Galper
Security Development, IBM Boca Raton

Bruce Scheer
Security Development, IBM Boca Raton

Doc Shankar
Security Development, IBM Boca Raton

# Chapter 1. Information Security

Protecting enterprise's information is more and more essential in maintaining organizations competitive edge. IBM offers a wide range of products, technology, solutions and services to protect customer's information assets where they are located: on operating systems, in servers, in databases and across networks. Through our commitment to open systems, IBM is offering solutions to better protect information in open, heterogeneous, client/server environments.

In general information security means to avoid affects against confidentiality, integrity, and availability of data stored in computers.

Other risks in computer environments possibly are incorporation of destructive program code such as computer viruses or trap doors and misuse of software licenses.

## 1.1 Risks

The risks are varied. On one hand, operating errors and accidental data loss can significantly increase costs for support and maintenance. On the other hand, there is the danger of deliberate attacks, data manipulation and theft.

### 1.1.1 Integrity

Data must not be destroyed or harmed in its integrity neither by accident nor by mistake.

As the number of PC users in the environment rises, the costs for PC-support increases due to accidentally erased or changed configuration files of the operating system or applications.

The intended destruction or corruption of files (sabotage) is often less than the accidentally destruction, but is not to be ignored as it can cause the ruin of the enterprise.

An example of the importance of integrity is an electronic bank transaction. In this case falsification of data can cause the loss of money.

### 1.1.2 Confidentiality

Data must be protected from unauthorized disclosure.

The unauthorized inspection of data cannot be noticed if an information security subsystem is not used. Only information security subsystems will track opening and copying of files.

Again the electronic bank transaction provides an example. If a customer's PIN is spied out, an unauthorized person may gain access to his or her account.

### 1.1.3 Availability

Data must not be hidden or held back by an unauthorized measure.

Data may not be moved to other directories where the authorized user cannot see it.

### 1.1.4 Incorporation of Destructive Programs

There are many different types of destructive programs. First of all there are computer viruses. Their number have grown explosively in the last years. More than 6000 are known in 1995. Computer viruses are easily incorporated into computers by accident or intentionally.

Other species of destructive programs are trap doors and logic bombs.

### 1.1.5 Unauthorized Use or Abuse of Resources

Computer resources can easily be misused. Doing some private work, for example typing a letter, is quite harmless and may not result in any punishment.

However, activities that are not management approved on the company's computers is unauthorized use of computer resources.

Other abuse of computer resources is:

- Destroying computer equipment
- Interrupting power lines
- Interrupting data transmission lines

### 1.1.6 Misuse of Licenses

Copying licensed software for home use is often done. This is in many countries a crime.

On the other hand, money can be saved by paying only for licenses that are actually used. Lots of software is not used when people are on vacation or ill but it cannot be used because people cannot share their PCs.

There is another problem with licenses. Programs will be brought from home, for example user ordered a program for business usage at the purchase department. It may take some time until it arrives and in the meantime a user brings the program from home to use it in the office. Probably at that time it is then copied between users.

## 1.2 Security Drivers

The requirements that drive the security strategy and architecture have come from several different sources. This section presents a view of the requirements of the major sources.

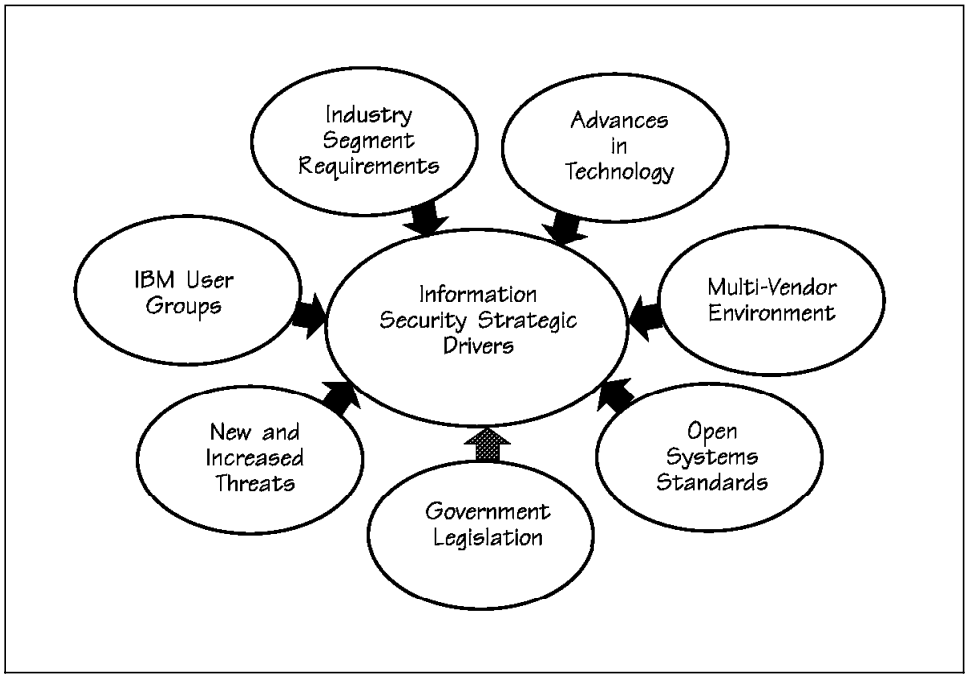



*Figure 1. Drivers of Information Security*

### 1.2.1 New or Increased Threats

The threats that exist in a centralized computer environment are well known. In addition to central processing on mainframes and minicomputers, personal computers and PC networks are being used increasingly for individual data processing. The two worlds are converging to form heterogeneous networks which are organized as client/server environments. Due to their open architecture, PCs are particularly susceptible to attacks on information security.

### 1.2.2 Multi-Vendor Computing Environments

Most enterprises today have multi-vendor computing environments. In the past, they may have operated as autonomous domains. Recent networking solutions provided the base for communications interoperability and new applications allow increased processing interoperability.

### 1.2.3 Advances in Technology

Evolution of the computing environment, from centralized operations centers to the world of fully distributed capabilities, has significantly changed the requirements of information security. Both as host terminals and as clients in networks, PCs have access to large amounts of data. PCs normally do not offer protection against attacks to data′s integrity, confidentiality, or availability.

### 1.2.4 Legal Reasons

The increasing focus of many governments on the implications of computer technology to privacy issues demands increased security. This demand becomes even more critical as multi-national enterprises implement world-wide networking of computers.

In some countries protection of personal data is required by law. Personal data may be, medical and health data, income data, or data used by authorities.

But also protection of business data may be required by law. That is, book keeping data and other information that has to be provided for control by authorities.

### 1.2.5 Business Needs

There are only few legal reason to protect data. Most data used in business has to be protected because of its value to the business itself.

To survive in today′s fast-paced world, businesses are trying to be the first in the marketplace with new products, or provide a higher level of service at a lower price. As the business evolves, dependence on electronic data processing and information security increases to get the right information to the right people at the right time.

Valuable data includes contracts, accounts, and other customer-related data. Loss of this data could result in the bankruptcy of the company, if the recovery of that data is impossible or would cost too much time and money.

# Chapter 2. Security Environment

This chapter describes the security environment. It starts with a description of how to create a information security policy and establish it in the enterprise, followed by a portrayal of security levels.

## 2.1 Security Process Cycle

The process of securing an information system is a cyclical, on-going effort with involvement from all level of the corporation, from the highest level of management down to the end users and programmers. There are five primary stages in the security process cycle, as shown in Figure 2.

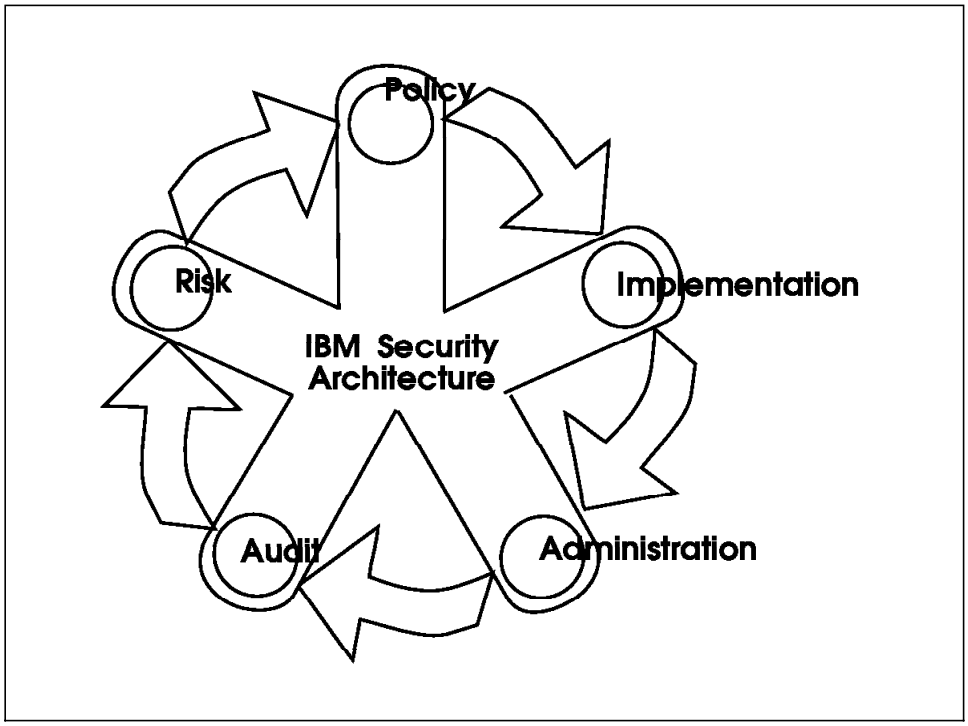



*Figure 2. The Ongoing Process of Information Security*

### 2.1.1 Risk Management

Risk management is the process that studies the potential security exposures and determines an acceptable level of security controls, implementation costs, and risk acceptance for those exposures that are not fully covered. Risk management includes:

- Identifying security exposures, such as natural disasters, external hackers, employee errors and sabotage along with the probability of occurrence.
- Identifying valuable business data, such as customer databases, research information or new product plans.
- Quantifying the value of potential loss for each exposure and valuable business asset.
- Determining the costs of implementing appropriate security controls.
- Weighing the costs of controls against the potential value of loss.
- Recommending changes in the security policy.
- Documenting acceptance of all risk exposure not covered by plans.

### 2.1.2 Defining Security Policy

A business must define its security needs in order to establish a management security policy and practice guidelines. The definition of security policy includes the following considerations:

- Not all data is treated in the same way. Some data is more important than other, so it is necessary to classify data. The following is an example of a possible classifications system:

  **Unclassified** — Everybody, even outside the enterprise, may see and handle this data.

  **Internal Use Only** — Only enterprise personnel may see and handle this data.

  **Secret** — Only personnel with a need to know may see this data. It requires special handling when sent by post both internally and externally.

  **Top-Secret** — Only personnel with a need to may see this data. Data may not be copied. Special handling when sent by post internally. May not be sent by post externally.

- Every single data or program file has its owner. This principle is called the data ownership principle. The owner is responsible for classification, backup, recovery planning when the data gets lost, and granting access to it. That does not mean that he has to do these things personally, but he has to initiate the appropriate processes, for example tell the manager of the DP center to backup this data once a week.
- High-level management commitment and responsibility for supporting the security policy.
- Documented procedures that deal with non-compliance.
- User education and a security awareness program.

### 2.1.3 Security Implementation

Security implementation is the process of procuring, installing and initializing the appropriate security products and system controls. The process of security implementation includes:

- Selecting security mechanisms appropriate for the security policy.
- Installing security hardware and/or software products.
- Defining system security controls and options.
- Grouping users and resources for effective administration.
- Classifying data and resources.

  See also *Security Enhancement Solutions for Workstations*, SG24-4569, for a discussion of administration for grouping users and resources.

### 2.1.4 Administration Functions

Administration is the process of applying the security policies and practices for an organization, which is largely the administration of security objects such as:

**Passwords** Every user ID has its own password to authenticate the person who uses this user ID.

**Users** Every person who work with a computer must have at least one user ID to identify her or him.

**Usergroups** User IDs can be gathered into groups when they need access to the same data or program objects, for example all members of the purchase department.

**Resources** User IDs may need access to the serial or parallel port of the computer.

**Access rights** User IDs need access to data or program objects. These rights are described in 2.2.2, “Authorized Workstation Access” on page 12.

**Desktop objects** Each user ID may have its own desktop.

**Trusted programs** When a user may not have access to the COM1 but the user who works with that user ID needs to transmit data via a modem, then the administrator will grant access to COM1 to a specific program. This program is then called a trusted program.

**Roles** Some user IDs ″perform″ a role within the information security subsystem. Roles are, as follows:

- Administrator
- Auditor
- Subadministrator
- Group administrator
- User

**Auditing information** Each event in the security subsystem has to be logged for further auditing. Audit events could be:

- Logon violation.
- Unauthorized access to a file.
- Administrator created a new user ID.
- User ID unused since 250 days is suddenly reactivated.

**Rules for passwords, working hours, etc.**
Password rules for administrators may differ from the rules for users; for example the administrators password must have at least eight characters instead of six for the user and also must consist of letters and numbers.

Please see *Security Enhancement Solutions for Workstations*, SG24-4569, for a more detailed discussion on this topic.

### 2.1.5 Security Audit

Audit is the continuous review of security controls and security events. Audit results are periodically reported to management and used as input for subsequent security process cycle efforts to update the security policy and implement new controls or enhance existing controls.

Security audits can include the following:

- Self testing or independent testing.
- Penetration testing.
- Internal compliance.
- External compliance.

The auditor should pay specific attention to logon and access violations, especially when there are a many of them in a short period of time and done by a single user ID. Normally a user ID should be revoked automatically by the information security subsystem after three attempts of logging into the system with a wrong password.

If a user ID tries to access files to which it is allowed, then the auditor should discover what is the reason for this access violation. Maybe the user is allowed to access that file, but the administrator missed to arrange the access in the information security subsystem.

## 2.2 Levels of Protection

There are four levels of protection. The first of them is ″no protection″, which is actually a protection level, but without any security.

### 2.2.1 No Protection

This level means exactly what it says: no protection, except against loss of data by backing up from time to time.

This protection level may apply to a private home user who has no need for security, because he has no valuable data on his computer. He need not protect his data against unauthorized modification (integrity) or disclosure (confidentiality) and the user can easily replace his data from other sources (availability), for example, obtain account data from the bank.

However, the user should be aware of the risks. With no protection, everybody who has access to the computer also has access to the data and

programs stored in the computer. Abuse of licences or unauthorized access to bulletin broadcasting systems is possible.

### 2.2.2 Authorized Workstation Access

If the user has valuable data on the computer or when the computer belongs to a company or it is a portable computer, there is an important reason to protect this computer against the risks described in 1.1, “Risks” on page 1.

Authorized workstation access is the lowest level of security. The computer is protected against booting by an unauthorized person. In general the computer will ask the user for a password, when booted or powered on. This is also known as ″pre-boot authentication″ which is described in detail in *Security Enhancement Solutions for Workstations*, SG24-4569.

Pre-boot authentication is not to be mixed up with the BIOS password. The BIOS password is easily deinstalled by unplugging the battery on the mainboard of the computer.

A little more attention is to be payed on the protection of the harddisk in the computer. If the computer is booted by diskette, the harddisk has to be locked, even if the correct password was entered. The password may be obtained by an unauthorized person.

The harddisk must be protected even when it is removed from that computer and installed in another. So encryption is part of the authorized workstation access. If the harddisk is encrypted, an unauthorized person cannot access the data on it.

There are many mechanisms to encrypt data. In the following are some of the most well known:

**DES** Data Encryption Standard, also known as the Data Encryption Algorithm (DEA). The best known and most widely known cryptographic algorithm. It provides a very high level of security and is used, for example to encrypt Personal Identification Numbers (PIN) or Electronic Funds Transfers (EFT). The disadvantage of DES is its slow speed.

**IDEA** IDEA is a new algorithm which is very secure and a little bit faster than DES.

**Blowfish** Also a new algorithm. It is less secure than IDEA, but faster.

**XOR** Very fast, but not actually an encryption algorithm. It is a modified exclusive OR Boolean operation.

It might be useful to encrypt data on diskettes. In addition, it should be possible to lock diskette drives in a manner that allows their use only with specially encrypted diskettes. This introduces a new function in securing data exchanged via diskettes. If all users in a company use the same key and algorithm to encrypt their data on diskettes, all data is completely interchangeable within the company, but not to the outside. If different groups of users use different keys, data is interchangeable within groups, but not between them. This helps to establish an easy to use data classification system and data ownership principle. See 2.1.2, “Defining Security Policy” on page 8, and *Security Enhancement Solutions for Workstations*, SG24-4569, for a discussion on that topic.

### 2.2.3 Multi-User Desktop Protection

If there is no protection on the computer and there is more than one user who uses it, they all share all data stored in it. This leads into new aspects of information security.

Since they have different jobs, they use different data files and/or programs. This results in different desktops users will use. These desktops must be secured. User A may not alter user B′s desktop, for example, moving, copying or deleting WPS objects.

Installing a desktop protection system is substantial if there is a number of users on one single computer. The desktop protection system has to manage the actions the users will use to handle their WPS objects. This means users may have different rights dealing with their or other user′s WPS objects,

- Create a new object
- Move an object
- Open an object
- Delete an object
- Access to pop-up menus

When users travel from one computer to another they expect that their desktop travels with them. It must be sure that the user finds his own desktop on every computer in his working area. The desktop protection system must therefore have advanced administration functions.

The administration of a desktop protection system could become very complex if the desktop security system is used in a network. There must be

a central security server within the network. Each workstation has to identify itself to the security server when turned on. This is necessary to make sure no unauthorized workstation enters the network and undermines the security system. Please refer to *Security Enhancement Solutions for Workstations*, SG24-4569, for more detail on administration.

### 2.2.4 Resource Access Control

If somebody intends to gain access to data or program files on a computer that has only multi-user desktop protection, he will be able to, if he uses methods which the desktop protection system is not designed to prevent.

Normally a desktop protection system can be deceived, because its meaning is to prevent desktops, but it has no access control.

Desktop protection only filters the view at resources and objects. Some applications give the user the possibility to gain access to data objects via their built-in macro language. A sophisticated user can go around the protection mechanisms of a desktop protection system.

To avoid this situation and for real access control, the operating system has to provide functions which the security system can use to protect the information on the computer.

OS/2 has these functions, called Security Enabling Services (SES).

#### 2.2.4.1 Separating Subject and Object

It is now possible to design an information security subsystem that has the capability to control access to data, desktops and other resources in a computer. That is crucial important if the operating system is a multi-tasking system.

Preventing unauthorized access to the computer or the data on it can be easily done as shown in 2.2.2, “Authorized Workstation Access” on page 12 and 2.2.3, “Multi-User Desktop Protection” on page 13. The real reason to create an information security subsystem is to:

1. Separate subjects and objects.
2. Establish a relationship between subject and object.
3. Access control to object via access control lists instead of hiding object on the desktop.

The objects of one user should regularly not be affected by another user. The access to objects has to be under control. It is then possible to distinguish the users and their actions from each other. This will help the

auditor to keep track of security-related events in the system and helps the support staff to keep the environment clearly arranged for easy maintaining.

But there are not only users busy on a computer. There are programs or tasks started by other users or programs or tasks. A lot of processes have child processes and so on. These together are called subjects. It is not anymore a user who wants access to a specific data file, but it is a subject that needs access to an object.

Subjects and objects have to be separated. If a subject needs access to an object, for example open a file for read, it asks the operating system to open that file. The operating system passes this request to the information security subsystem, which looks in it access control tables if the access is allowed or not. If the answer is yes, the file will be opened and the subject can read. This is described in detail in Chapter 3, “OS/2 Security Enabling Strategy” on page 21, Chapter 4, “Security Enabling Services” on page 33, and in *Security Enhancement Solutions for Workstations*, SG24-4569.

The main function of an information security subsystem is to keep subjects and objects separated, except when they are allowed to come together. This means to establish and maintain a relationship between subject and object as long as the subject exists in the operating system.

#### 2.2.4.2 Surviving Processes

User A may have started a process and then logged off the operating system. The process belongs to user A and is still alive, even when user B will log on to the system and start his own programs. The information security subsystem has to make sure that user B neither sees user A′s process nor has access to A′s data, even if he starts the same process as A did. That means that all processes in the operating system have to be personalized and separated.

An information security subsystem should also be capable of encrypting data files. Each user can be forced to encrypt his data file with a key. This key can be shared by a group of users so it is then possible to interchange data within this group, but not to another group or to the outside of the company. These confidentiality mechanisms protect data from unauthorized disclosure.

#### 2.2.4.3 Integrity

An information security subsystem should provide functions to safeguard the integrity of data. That means that given an original data object, a integrity object is calculated on the original data. The original data object and integrity object are kept together. Mechanisms such as a Cyclic Redundancy Code (CRC) check or a parity field check are appropriate to detect accidental modification of data. However, cryptographic methods must be used when there are concerns about the deliberate modification of data.

#### 2.2.4.4 Confidentiality

If the user deletes a file on the harddisk, the data will remain on the disk, because only the directory entry is deleted or the file is just marked as deleted. In addition the main storage of the computer and the SWAPPER.DAT should be cleared when a user logs off. The information security subsystem should provide functions to reliable delete the data on the disk, the main storage, and the SWAPPER.DAT.

#### 2.2.4.5 Single Signon

In today′s computer environment, the user has many different applications on different host systems. A password is needed for all of these systems. Things are getting worse when these password have different expiration dates. The user either has to remember different password or to write them on a sheet of paper. Normally the user will use only one password for all the systems he uses. If this password is inspected by an unauthorized person, this person gains access to all systems the user has access to. The same happens when an unauthorized person takes the recorded passwords of the user.

An information security subsystem should support the user by handling his passwords. The user need only to know the password for the information security subsystem. The subsystem will handle logons to other systems for the user and also keep track of the passwords.

#### 2.2.4.6 Administration and Audit

Attention should be payed on the administration of a high sophisticated information security subsystem and on auditing, as well.

Administration and auditing is very complex and the subsystem should support the administrator and the auditor with intelligent and easy to use functions to keep the system secure. Please see *Security Enhancement Solutions for Workstations*, SG24-4569, for more details.

## 2.3 Different Environments

Information processing has undergone rapid and significant change in the last few years. This change has caused a corresponding increase in concerns for the security of information assets.

### 2.3.1 Changing the Business Environment

In former days of computing life was easy. There was one big computer center with a few mainframe host computers surrounded by the input/output devices. RACF managed all information protection and the DP-center was a closed shop. In the offices were only 3270 terminals, no PCs.

But times changed. The big DP-center has been replaced by PC servers. In the offices we have clients with own input/output devices. In fact everybody who has its own PC in his office has become the manager of a DP-center with all its advantages, but also all responsibilities.

Traveling sales representatives often need to communicate with the enterprises main system from anywhere in the world via telephone lines or even by radio. This introduces new dangers and challenges regarding information security.

### 2.3.2 The Home User

The home user certainly needs only low-level security. Backing up data periodically will fit most users. But information security is a very individual thing.

Even a home computer user installs applications and uses the computer for personal information such as tax return, and it has users who will use it to play games. The owner may want to avoid users access to his data. This could be a good reason to install some security measures.

### 2.3.3 SmallOffice and HomeOffice (SOHO)

If the computer is used for business in small and home offices, more attention should be paid on security. The minimum security is authorized workstation protection. See 2.2.2, “Authorized Workstation Access” on page 12 for further discussion on this topic.

In addition, access to the computer may be controlled. For example, it may be placed in a room that can be locked.

If the computer is shared by more than one user, multi-user desktop protection should be used to secure the computer.

### 2.3.4 Client/Server Environments

Normally enterprises have more than one computer and they all are connected to each other via network facilities. Networks are spreading over the world. They cross city limits, state and national boundaries. There are two main classes of networks:

**Closed networks** These cannot be accessed from outside. Companies normally use this type of network. An example may be IBM′s internally used network.

**Open networks** These can be accessed by any user in the world via modem and telephone lines. The best example of an open network is the Internet.

Times are changing. Enterprises recognize the need to open their networks. Home workers and sales representatives have a need to access the enterprises network from any place in the world. So networks grow more and more from a closed into an open state.

Please refer to *Security Overview of Open Systems networking*,GG24-3815 for a more detailed discussion on this.

#### 2.3.4.1 Physical Security

Servers and mainframes, which are actually servers, too, should be located in a safe place. Only authorized personnel should have access to these rooms. The same applies to network components, such as communication controllers, modems or hubs, too, of course.

#### 2.3.4.2 Data and Desktop Security

All access to data or programs must be under control. Each user must be separated from the other. Only the owner for the data may grant access to it. Please refer to 2.1, “Security Process Cycle” on page 7, 2.2.4, “Resource Access Control” on page 14, and *Security Enhancement Solutions for Workstations*, SG24-4569.

#### 2.3.4.3 Administration

Attention must be payed to administrating information security subsystems. The security server has all information of all rights of every user stored in a secured and encrypted database. Any computer in the network may be the security server. Only the security administrator has complete access to all functions of the information security subsystem.

The administrator of the information security subsystem will carry out the change of the users access rights. See also *Security Enhancement Solutions for Workstations*, SG24-4569.

#### 2.3.4.4 Auditing

Auditing is a way of check and balance. To control the security administrator, each event produced by that user ID should be logged. This is not to control the person who is the administrator, but the role it has in the information security subsystem.

The role of the auditor is to do that controlling. In some cases he may not do that on his own. He has to consult a second auditor or the administrator to perform his job. That is called four-eyes principle. In some countries the works committee must agree when the auditor looks into such sensitive data, like who opened which file and when or at what time someone logged in or out.

#### 2.3.4.5 Management Process

The management has to define and establish organizational functions for security in the enterprise. That is also called security policy. This is discussed in detail in Chapter 1, “Information Security” on page 1, and 2.1, “Security Process Cycle” on page 7.

### 2.3.5 Portable Computers

These kind of computers require special attention. Portable computers are not connected to the security server all the time. They need their own security server, which is a subset of the security server at the home location. If they do not have a local security server, user rights cannot be controlled and the system is open. See also *Security Enhancement Solutions for Workstations*, SG24-4569. for more information about this topic.

# Chapter 3. OS/2 Security Enabling Strategy

In the previous two chapters we have defined and discussed information security and the customer's requirements. In this chapter we look at the strategy that has been applied to OS/2, and take a brief look at client/server security.

## 3.1 SES Strategy Overview

There are two basic approaches that can be taken by anyone who wishes to apply security to a workstation:

- Restrict a users access to a limited subset of the available applications, that is, simple multi-user desktop-based security systems.
- Allow the user full access to all applications, but enforce security policies at the operating system level, that is, full resource access control-based security.

These differing security requirements are addressed in OS/2: the operating system was extended to allow the addition of an external security component. This external software component is an Installable Security Subsystem (ISS). The OS/2 kernel, in cooperation with the ISS software component, will enforce workstation security policies. This approach satisfies a much wider range of customer requirements than any one application ever can, including: multi-user desktop protection, C2 level operating system security, compatibility with a wide variety of existing DOS/Windows security products; single signon, centralized security administration in a distributed computing environment, etc.

The ISS may be delivered as part of an ISVs security product, an IBM product, or a customer application.

This security strategy therefore does the following:

- Allows the customer the flexibility to select from any vendor with SES enabled products.
- Allows the customer the flexibility to select the product that suits their needs in terms of the level of security, and the type of product, for example, multi-user desktop.
- Enables security product consistency with other platforms, for example, DOS/Windows.

- Enables the industry workstation security experts to develop OS/2 products.
- Ensures that every security-relevant event in the operating system can be authenticated.

## 3.2 Open and Closed Security Architectures

When providing security enabling or a security solution at the access control level, a decision has to be made about whether to publish details of the interfaces, thus allowing third party software suppliers to utilize those interfaces and provide security solutions for the platform, or whether to restrict access to the interface and provide a purely proprietary solution. This section aims to discuss the advantages and disadvantages of the two approaches.

### 3.2.1.1 Closed Security Architectures

The closed, or propriety, approach to operating system security offers the customer little choice. The security interface is kept concealed which prevents, or at best severely restricts, external security software vendors from developing security solutions for that platform. The implications of this in the client/server environment are worthy of note:

For robust security in an enterprise environment which includes closed security architectures, all of the servers and clients have to be running secured operating system software from the same vendor. This is difficult to accept in the majority of enterprise environments where several different operating systems (DOS, DOS/Windows, OS/2, WindowsNT) need to interact in a network.

### 3.2.1.2 Open Security Architectures

An open architecture has details of the security interface published and available for use by anyone. This is the approach taken by IBM with its security strategy for OS/2. This allows the independent security software vendors the freedom to develop robust security applications that take advantage of the security services available via the SES interface.

This leaves ISVs free to expand their range of products to OS/2 platforms, and thus provide a consistent user interface to the security product across a number operating systems. If a consistent approach is made with regard to the software security vendor, an organizations investment in updating software releases, migrating to different platforms and training in the use of the security product can be controlled.

Table 1 on page 23 shows the differences between a closed and open architecture.

| Requirement | Closed Architecture | Open Architecture |
|---|---|---|
| Will your current security software vendor be able to develop to this architecture? | Possibly | √ |
| Is it possible for a vendor to offer a robust security solution in this environment? | Unlikely | √ |
| Will you be able to select the level of security that fits your needs? | Limited ability | √ |
| Will you be able to select a robust security solution that will work in a multi-OS networked environment? | | √ |
| Will you be able to select a robust security solution that will work with in a single OS networked environment? | √ | √ |
| Is it possible to develop your own security application to fit your specific needs? | | √ |

*Table 1. Comparison of Open versus Closed Security Architectures*

## 3.3 OS/2 Strategy

The protection of resources on workstations with operating systems such as DOS, Windows, OS/2, WindowsNT, AIX, etc. can be divided into two basic approaches:

1. Multi-User Desktop Protection

   Protection of workstation resources at the user interface (desktop) level. All security-relevant user activity (for example, clicking on an icon to access a file) is intercepted and interpreted at the desktop level where protection policies are enforced.

   Note that these protection services may be provided by the user shell itself, or the user shell may provide mechanisms for an external component to intercept, interpret, and filter user activity.

   One way to look at desktop security from a customer′s point of view is, to enforce desktop security policies, the operating system basically becomes a restricted system; that is, users are only allowed to invoke services and execute applications that a system administrator trusts will not break the security policies of the restricted desktop.

2. Resource Access Control.

   This is enforcement of security policies at the operating system level. All security-relevant operating system services (for example,an operating system call to open a file) make a privilege transition to the trusted computing base (trusted operating system code) where the security policies are enforced.

   Note that the security services may be provided by the operating system itself (for example WindowsNT, AIX), or the operating system may enable an external component to enforce its security policies (similar to RACF on VM or MVS).

   With this approach to security, the secured operating system still appears to be completely open. Users are therefore free to develop and execute any applications they want (this includes applications that may attempt to break the security policies enforced by the secured operating system).

These two approaches are very different (target market, level of protection, impact on performance and usability, cost, etc.). Each satisfies a subset of customer requirements for protection of workstation resources, and can be used separately or together to satisfy most OS/2 security requirements.

The OS/2 security enabling strategy facilitates both of these approaches and addresses the diverse set of workstation security requirements.

Table 2 shows the security functions available with various operating system platforms and who is responsible for providing those functions.

| Function | Windows 3.x | WindowsNT | OS/2 |
|---|---|---|---|
| Operating system integrity | | native | native |
| User identification and authentication | ISV | native | ISV |
| Resource access control | ISV | native | ISV |
| Audit and audit management | ISV | native | ISV |
| Multi-user desktop protection | ISV | native | ISV |
| Central site administration | ISV | native | ISV |
| Client logon integration | ISV | native | ISV |
| Standard API | | | native |
| Standard KPI | | | native |
| Compatible security applications | | | native |
| Target security evaluation | D2 in conjunction with ISV code | C2 | D2 in conjunction with ISV code |

**Note:**

**ISV -** This function would have to be provided by a vendor product. If the operating system supplies an API and/or KPI then the application can interface with that.

**Native -** This function is native to the operating system.

*Table 2. Comparison of OS/2 and Microsoft Security Strategy*

The diagram shown in Figure 3, shows the key components of the OS/2 security enabling strategy. These components may be provided as part of OS/2, other IBM products, ISV products, or customer applications.

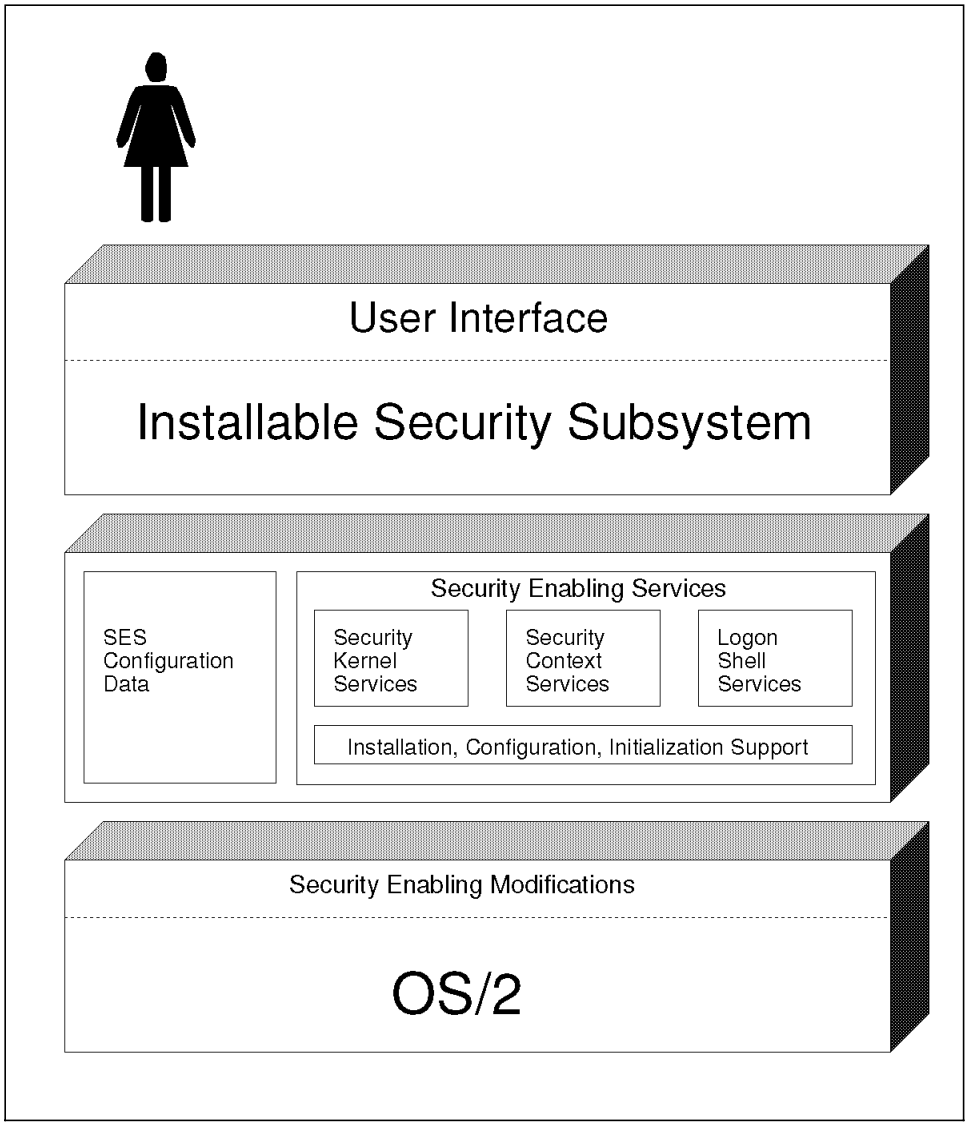
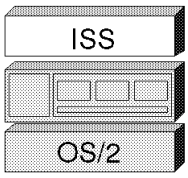



*Figure 3. OS/2 Security - Role of SES in a Secured OS/2 Workstation*

### 3.3.1 Installable Security Subsystem (ISS)



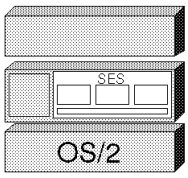

The ISS is a set of components that provide security features for a secured OS/2 operating system. This set of components will vary depending on the security features required by the customer, and may be developed by an ISV, by IBM, or by the customer themselves.

Services that may be included as part of an ISS:

- User identification and authentication (logon)
- Resource access control
- Audit
- Security context and trusted program support
- Security policy administration tools

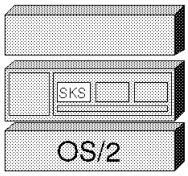
The components included in an ISS will depend on the security features that are required by the customer.

### 3.3.2 Security Enabling Services




SES enables an ISS to provide robust (C2 level) operating system security services (such as RACF on VM or MVS).

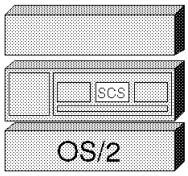
### 3.3.2.1 Security Kernel Services




The SKS is responsible for:

- Routing security relevant operating system events (such as file system access process creation, etc.) to an ISS to enable it to enforce security policies on those events.
- Providing operating system services for an ISS security kernel that aren't normally available to a device driver running at Ring-0 (such as file open, read, write, etc.).

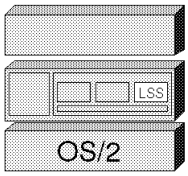
### 3.3.2.2 Security Context Services




The SCS allows:

- An ISS to establish an association between a process/thread and a subject handle, which the ISS can associate with a user's security credentials (such as user identity and group membership).
- An ISS to provide trusted program support (for example, a database application that can access the database file on behalf of a user even though the user is not authorized to access the database files directly).
- An ISS to provide concurrent multi-user access to an OS/2 workstation (for example, one local user and multiple remote TCP/IP users).

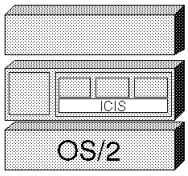
### 3.3.2.3 Logon Shell Services




The LSS enables:

- Alternative authentication mechanisms (such as smart card, finger print, etc.).
- The perception of a single signon by coordinating events that are relevant to a logon session (such as log on, log off, lock, and unlock) with cooperating local and remote security components.

### 3.3.2.4 Installation, Configuration, Initialization Support




The ICIS enables:

- An ISS to provide secure installation, configuration, and system initialization (boot) features.
- An ISS to provide trusted recovery services if the trusted computing base becomes corrupted.

### 3.3.2.5 SES and ISS Communication

Communication between SES and the ISS is accomplished through handles and programming interfaces.

***Subject Handles:*** OS/2 is a multi-process user environment. Because of this, it is necessary to have a method to associate users with the process or threads they′ve started. This is accomplished in OS/2 by the use of subject handles, where a subject may be defined as a user, group, or process.

Subject handles associate user identifiers and other credentials with actively running OS/2 processes. When a user logs onto the system, a unique handle is created for the lifetime of the user′s session. This handle is associated

with the user′s name and password. Any appropriately privileged program can find the handle for a client process, and can in turn find the user name and password of the client. Furthermore, other security dependent applications running under OS/2 can associate their own notion of credentials with a handle. For example, a LAN file server application could associate LAN credentials with a user′s handle.

***Security Context Inheritance:*** With OS/2 SES, the security context inheritance policy between parent and child processes is deliberately flexible. It is up to the ISS to determine whether the child should inherit its parent′s authority.

The SES system default inheritance policy is that child processes are untrusted; therefore children do not inherit authority from the parent process. In contrast, with the POSIX-compliant inheritance model, the child process inherits the security context of the parent process. With OS/2, there is an option which may be specified to enable POSIX-compliant inheritance on a per-process basis. There is not a system-wide definition for this, it must be specified each time it is required otherwise the default inheritance policy will apply, that is,no inheritance of authority.

***Programming Interfaces:*** ISS applications need to interact with predefined OS/2 services at an application and a kernel level. OS/2 provides programming interfaces for the following:

- Application (API) components
- Kernel (KPI) level components

Through the APIs and the KPIs, an ISS can create, delete, reserve, and examine handles for processes and threads, control processes, wait for events, determine the order of execution for specific authorities, and receive kernel-level event information.

## 3.4 Interfacing With Other Products

With the current release of SES, the ISS is responsible for providing an application programming interface (API) to allow security-dependent applications to invoke security services. With future releases, there will be the opportunity to expand the facilities available with SES, to include a security framework which will allow the standardization of this API. This will allow security-dependent applications to work with any ISS that supports the security framework services. There will also be the opportunity to include an administration graphical user interface (GUI) to allow users to invoke security

services through a standard graphical interface. This would augment or replace the administration interface provided as part of the ISS which may or may not be a graphical interface.

The diagram shown in Figure 4 illustrates this point.

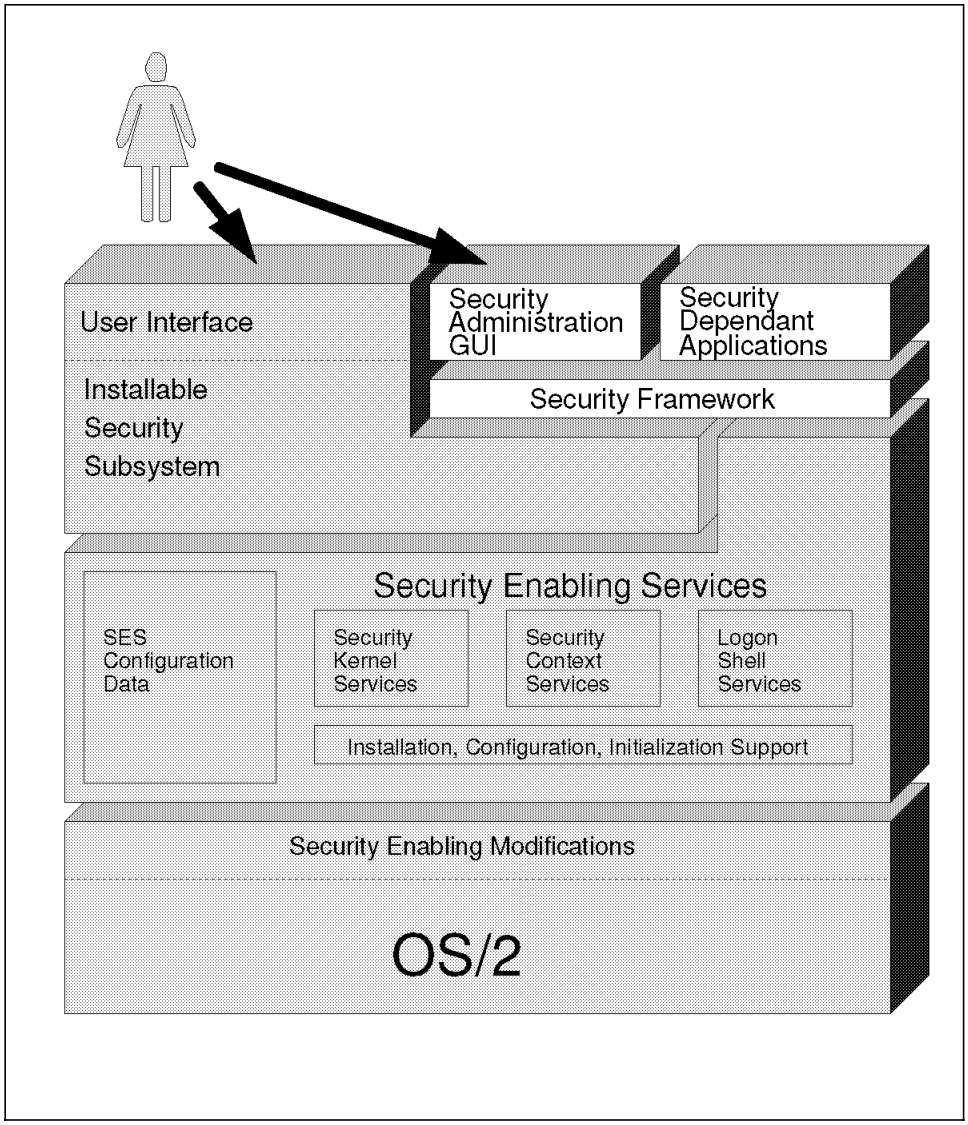



*Figure 4. API/GUI - One Possible Direction for SES*

#### 3.4.1.1 Security-Dependent Applications

These include network file/print/application servers (Novell, LAN Server), database managers (DB2), communications support (Communications Manager), banking applications, and insurance applications.

Security-dependent applications need security services such as user authentication or file protection, but typically don′t want to be dependent on a specific service provider′s API for these services.

# Chapter 4. Security Enabling Services

Only by utilizing the security enabling services (SES) of OS/2 can a truly robust security system be developed for OS/2. In this chapter, we take a close look at the SES components, and how they interact with both the OS/2 kernel and the Installable Security Subsystem (ISS).

## 4.1 SES Overview

OS/2 security requirements originate from a wide variety of customer environments ranging from the home and small office to large distributed computing environments that include a variety of client and server platforms.

The only truly secure way of dealing with these requirements also gives the customer the most choice is resource access control combined with an open security architecture. The resource access control is where security-relevant events such as file open, print, connect to a COM port, etc. are intercepted down within the operating system. It is only when the operating system becomes involved that a true C2 level security system can be developed.

With the realization of an open architecture, the freedom to exploit these security services is given to ISVs and customers alike, so robust, consistent (cross platform) security solutions can be developed. With the development of the SES interface for OS/2, IBM has delivered a resource access control level of security interface, combined with an open architecture.

The diagram in Figure 5 on page 34 illustrates the components that together make up a secured OS/2 operating system.

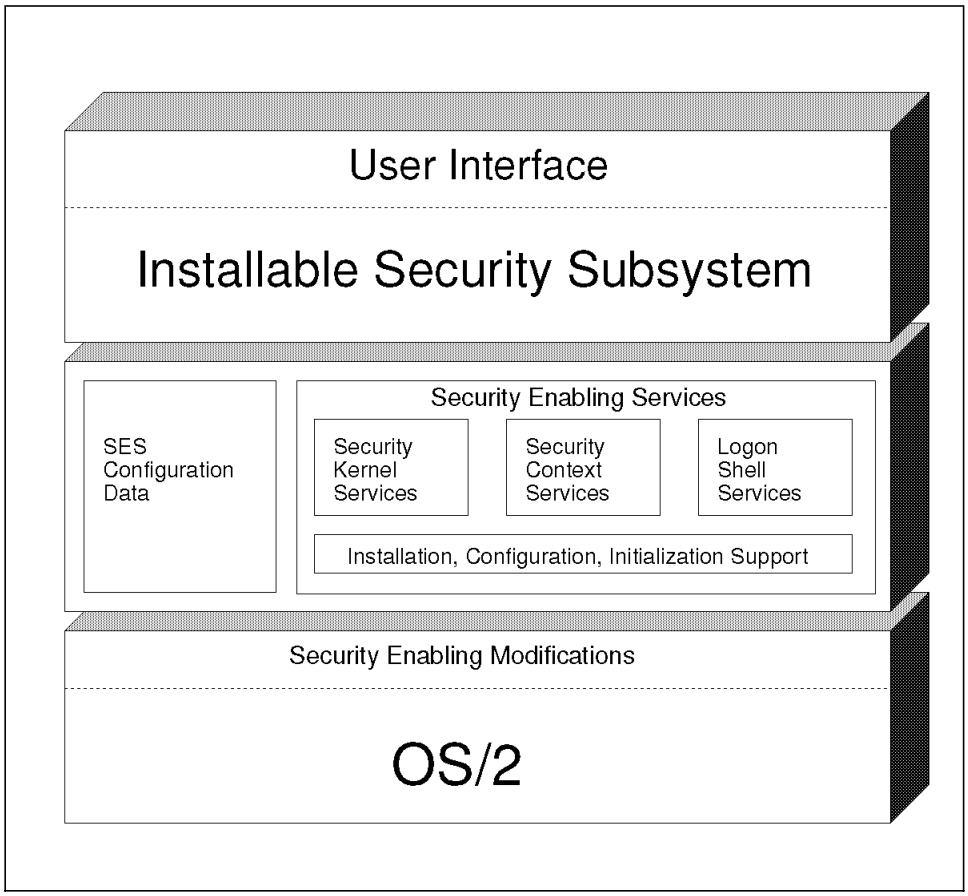



*Figure 5. SES - Key Components of the OS/2 Enabling Strategy*

- Security Kernel Services (SKS)

  This routes security relevant events to the ISS, such as file system access and process creation. It also provides operating system services to the ISS which would not otherwise be available at Ring 0, such as file system access.

- Security Context Services (SCS)

  This allows the ISS to associate a user's process, with that user's security credentials (user ID, group membership, trusted program privileges, etc.)

- Logon Shell Services (LSS)
- Logon Shell Services

In order to allow the perception of single sign on to the user, the LSS coordinates a number of components to log the user to the local security subsystem (ISS) and to other local/remote services. The authentication process can include the verification of smart cards, tokens, etc.

- Installation, Configuration, Initialization Support (ICIS)

  Secure installation, configuration, and initialization (boot) of an OS/2 system with SES, an ISS, and other security-related applications are handled by the ICIS component. This ensures the integrity of the system at these critical times, while minimizing the impact on the standard OS/2 installation, configuration and installation processes.

## 4.2 Security Kernel Services (SKS)

The SKS supports the ISS by:

- Intercepting and routing security relevant OS/2 kernel events to the ISS Security Kernel.
- Providing kernel level operating system services for the ISS Security Kernel.

The diagram shown in Figure 6 on page 36 illustrates the concept of the hooks and services that SKS provides for an ISS, using the OS/2 file system services as an example.

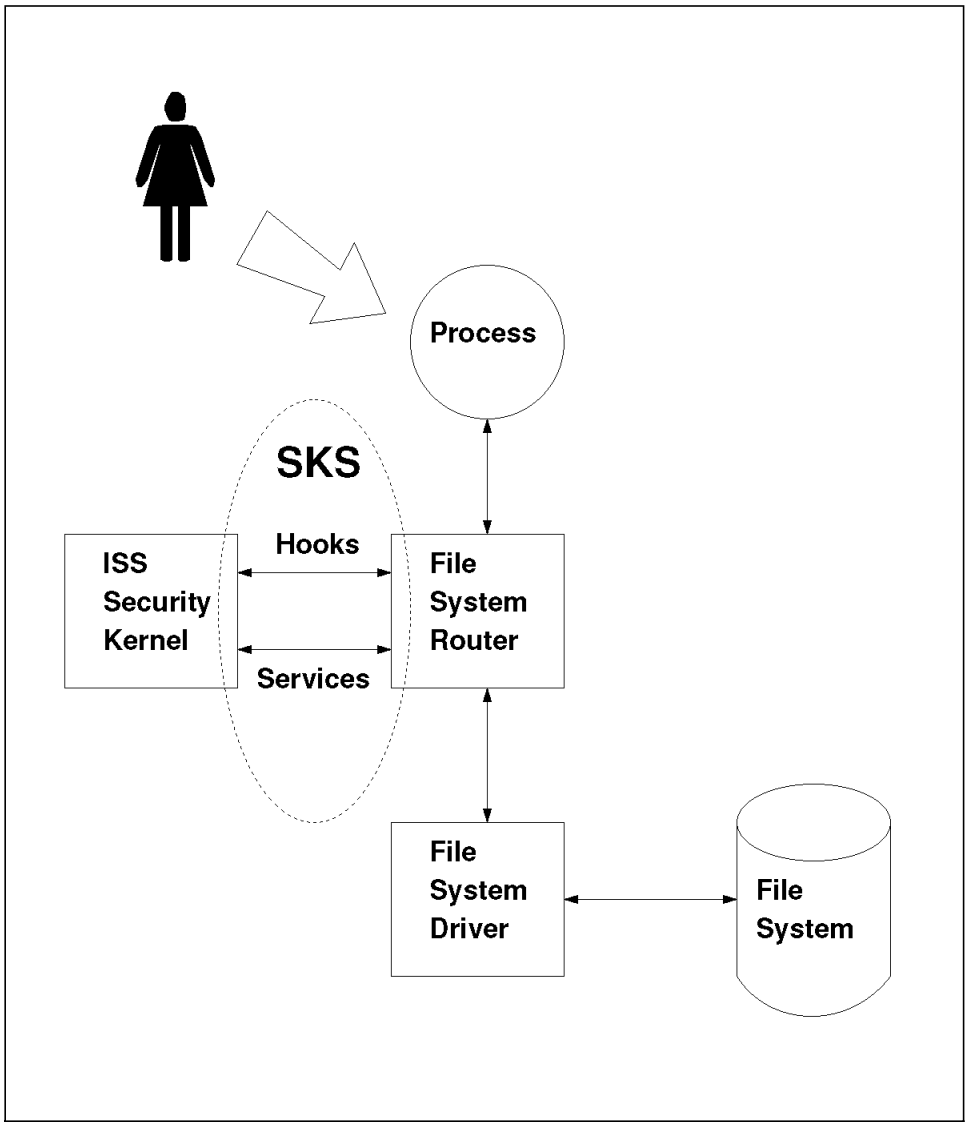



*Figure 6. SKS - Hooks and Services for the ISS Security Kernel*

### 4.2.1 Security Relevant Event Interception and Routing (Hooks)

The ISS security kernel device driver has to be notified of security-relevant events so that it can enforce access control and audit policies. This is performed by hooks into the operating system so that these events can be notified to the ISS and then acted on appropriately. The events that can be hooked directly via the SKS include:

- File access (open, read, write, close, change file pointer, delete)
- Directory access (make, remove)
- Load DLL module
- Execute program
- Callgate level support
- Multiple virtual DOS machine support
- Logon shell services trusted path support
- Security enabling services API audit support

### 4.2.2 Kernel Level Operating System Services

The ISS Security Kernel device driver runs at Ring-0, but has to be able to invoke some OS/2 services that are normally only available at Ring-3, such as file access for audit log updates. To minimize the performance impact this would otherwise cause, the SKS provides the following services to the ISS security kernel:

- File system access (open, read, write, etc.)
- Security context services

## 4.3 Security Context Services (SCS)

The objective of SCS is to enable an ISS (and other security applications) to provide C2 level security services, including enforcement of a Discretionary Access Control (DAC) security policy. A DAC security policy is based on a model for controlling access rights to objects based on the identity of subjects, where:

**Object:** Is defined as a passive entity (file, device, etc.) that contains or receives information. Access rights (read, write, execute, etc.) to an object implies access rights to the information in the object for use by the subject as specified by the access rights.

**Subject:** Is defined as an active entity (OS/2 process) executing on behalf of a user/group. The subject is associated with user/group/process credentials (user identification, group membership, trusted program privileges, etc.).

**Process:** Is defined as a program in execution, characterized by a single address space, execution state, and associated security context.

**Security Context:** Is defined as the information maintained for each process that enables security applications to associate the subject (process executing on behalf of a user/group) with the appropriate user/group/process credentials.

In an operating system that doesn't support multiple concurrent processes, such as DOS and Windows, the single process is always acting on behalf of the one-and-only local system user. No other users or processes can share the system. However, even in a single process system, multiple concurrently active security applications may need to maintain different definitions of user/group/process credentials.

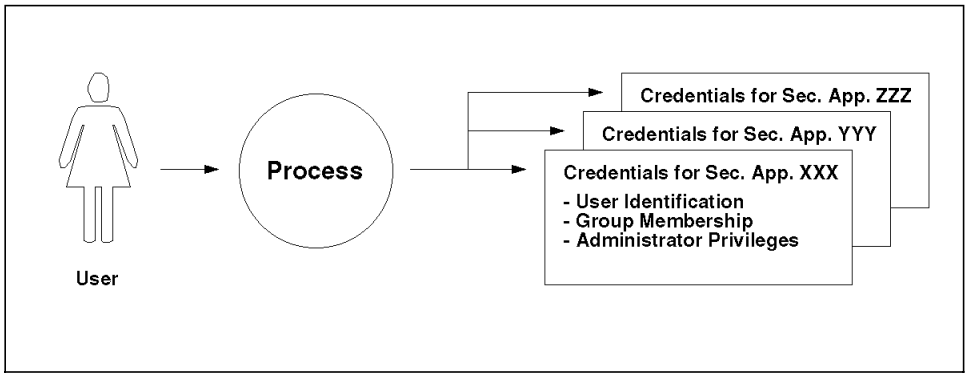



*Figure 7. SCS - Single Process Model*

In an operating system that supports multiple concurrent processes, such as OS/2, the potential exists for these processes to be acting on behalf of different concurrently active users (and/or trusted programs with their own user/group/process credentials). And, again, multiple concurrently active security applications may need to maintain different definitions of user/group/process credentials.

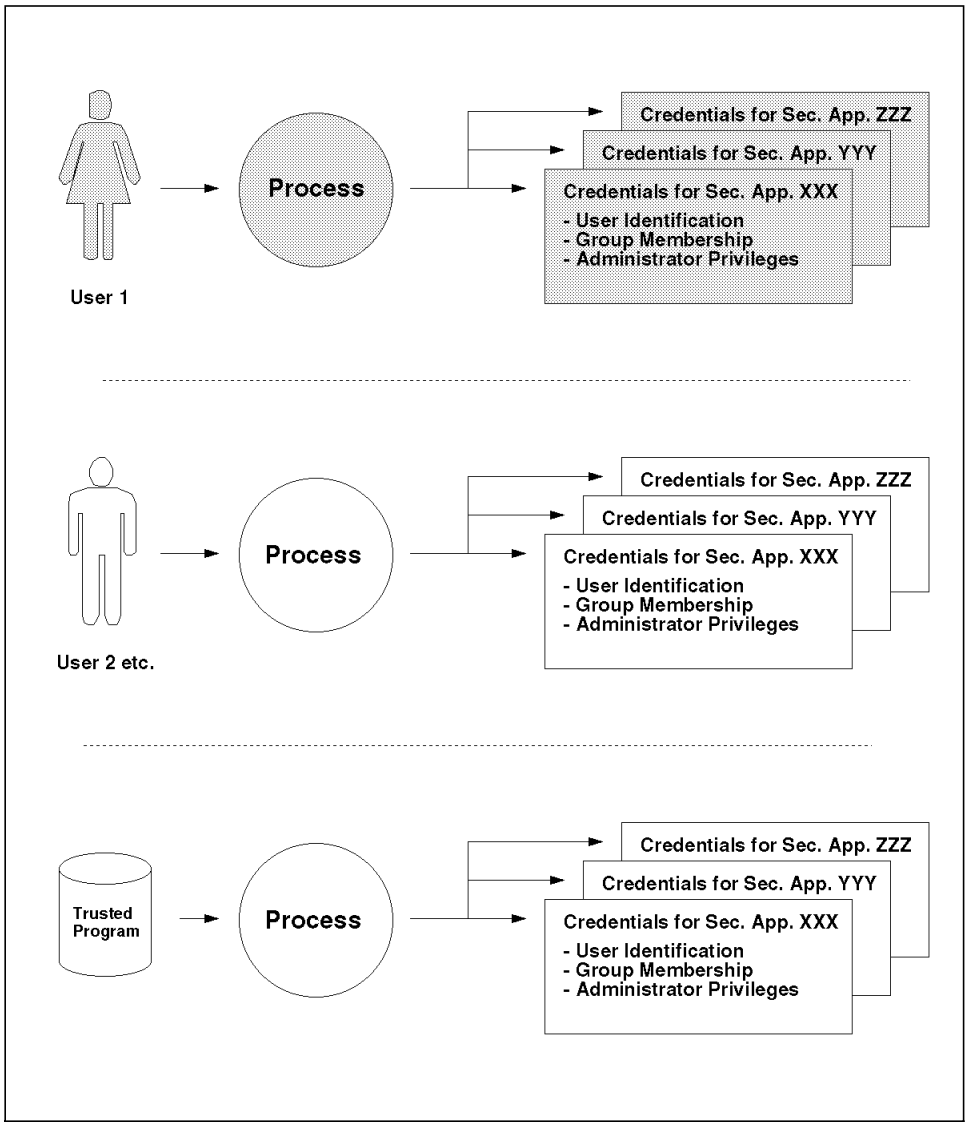



*Figure 8. SCS - Multiple Process Model*

In a single process operating system, it is possible for each security application to maintain its own user/group/process credentials for the one-and-only local system user without support from the operating system. However, in a multiple process operating system, where processes (and the associated security credentials) are dynamically created/terminated, the security applications need operating system support to manage the security credentials associated with each process.

To support dynamic management of user/group/process credentials associated with subjects (processes executing on behalf of users/groups) for multiple concurrently active security applications, SCS associates each process with a subject handle (dynamically-generated unique identifier) that each security application can associate with its own definition of the user/group/process credentials.

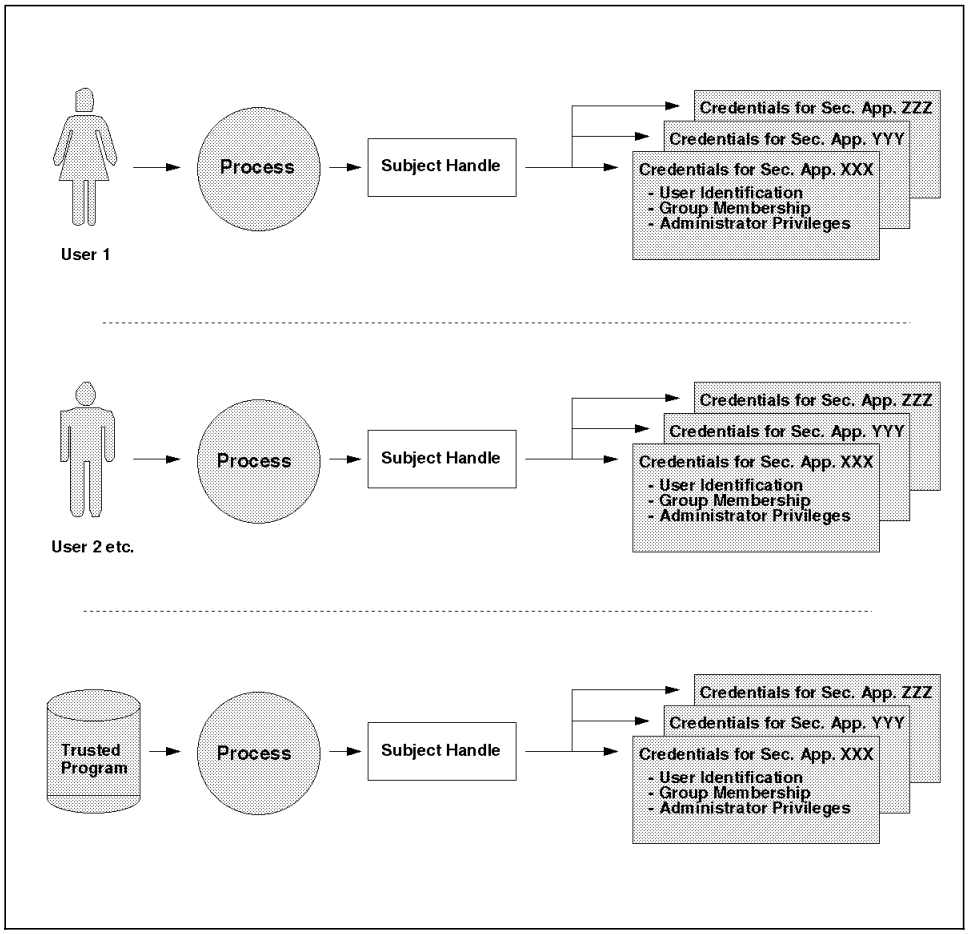



*Figure 9. SCS - Subject Handle Model*

To make the situation even more complicated, to support various trusted program/process models (for example: POSIX setuid, setgid, umask), each process potentially needs to be associated with several different sets of user/group/process credentials as follows:

- Client and agent user credentials (for example, POSIX uid) to enable a trusted process to act on behalf of the real user who requested the process services (client user) and the saved user associated with the corresponding program (agent user).
- Client and agent group credentials (for example, POSIX gid) to enable a trusted process to act on behalf of the real group of the user who requested the process services (client group) and the saved group associated with the corresponding program (agent group).
- Client and agent process credentials (for example POSIX umask and DCE tickets) to enable a trusted server process to act on behalf of a multiple clients (client process) and itself (agent process).

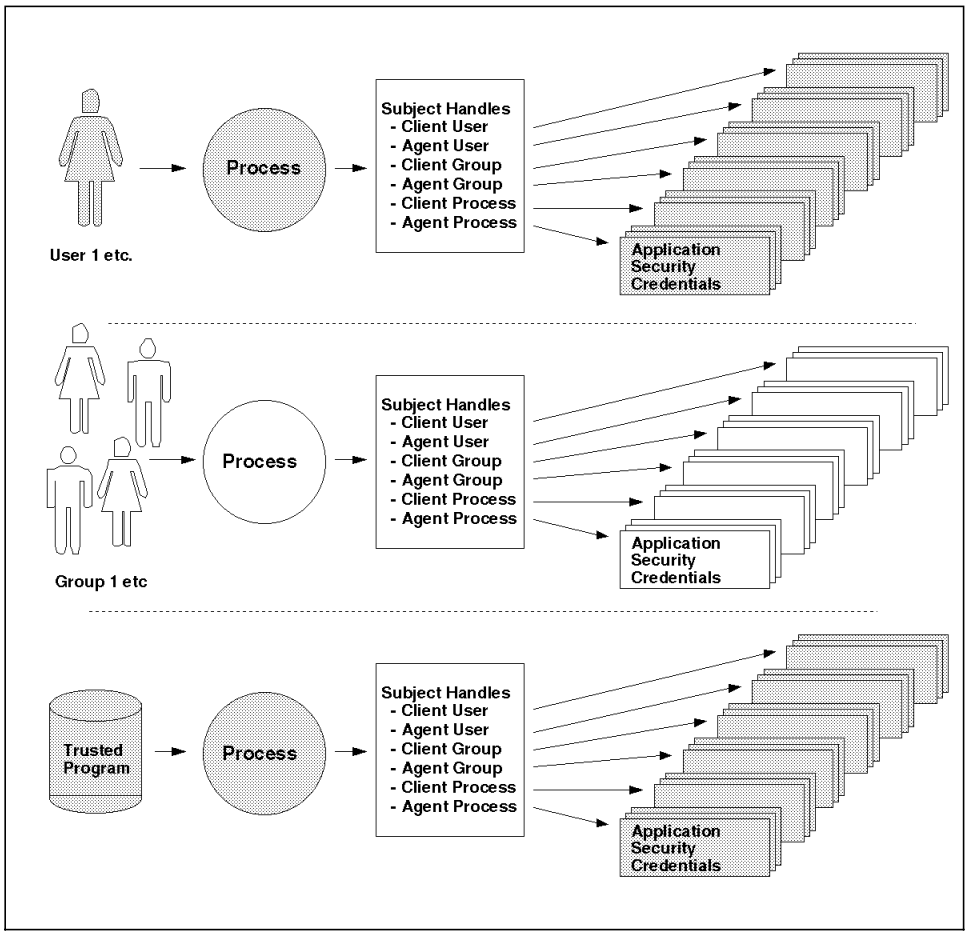



*Figure 10. SCS - Trusted Process Model*

To support either a process model (where all threads of a process share the same security context) or a thread model (where threads of a trusted process are allowed to maintain different security contexts), each process and each thread of a process potentially needs to be associated with different sets of user/group/process credentials.

However, even for the thread model, all threads of a process are executing the same program, and therefore have the same maximum security context (although the threads can have different effective security contexts). In addition, an OS/2 process is represented by Thread-1 (for example: if Thread-1 dies, the process dies), so the effective security context of the process can be represented by the effective security context of Thread-1.

SCS, therefore, maintains the following:

- One maximum security context for each process.
- One effective security context for Thread-1 (which represents the process).
- One effective security context for each additional thread if the thread requests its own security context (that is if it explicitly chooses a thread model instead of the default process model).

Please note that the vast majority of processes are untrusted, and consequently, all threads of the process will have the same effective security context as the maximum security context of the process. Only threads of a trusted process may have effective security contexts that are different from the maximum security context of the process.

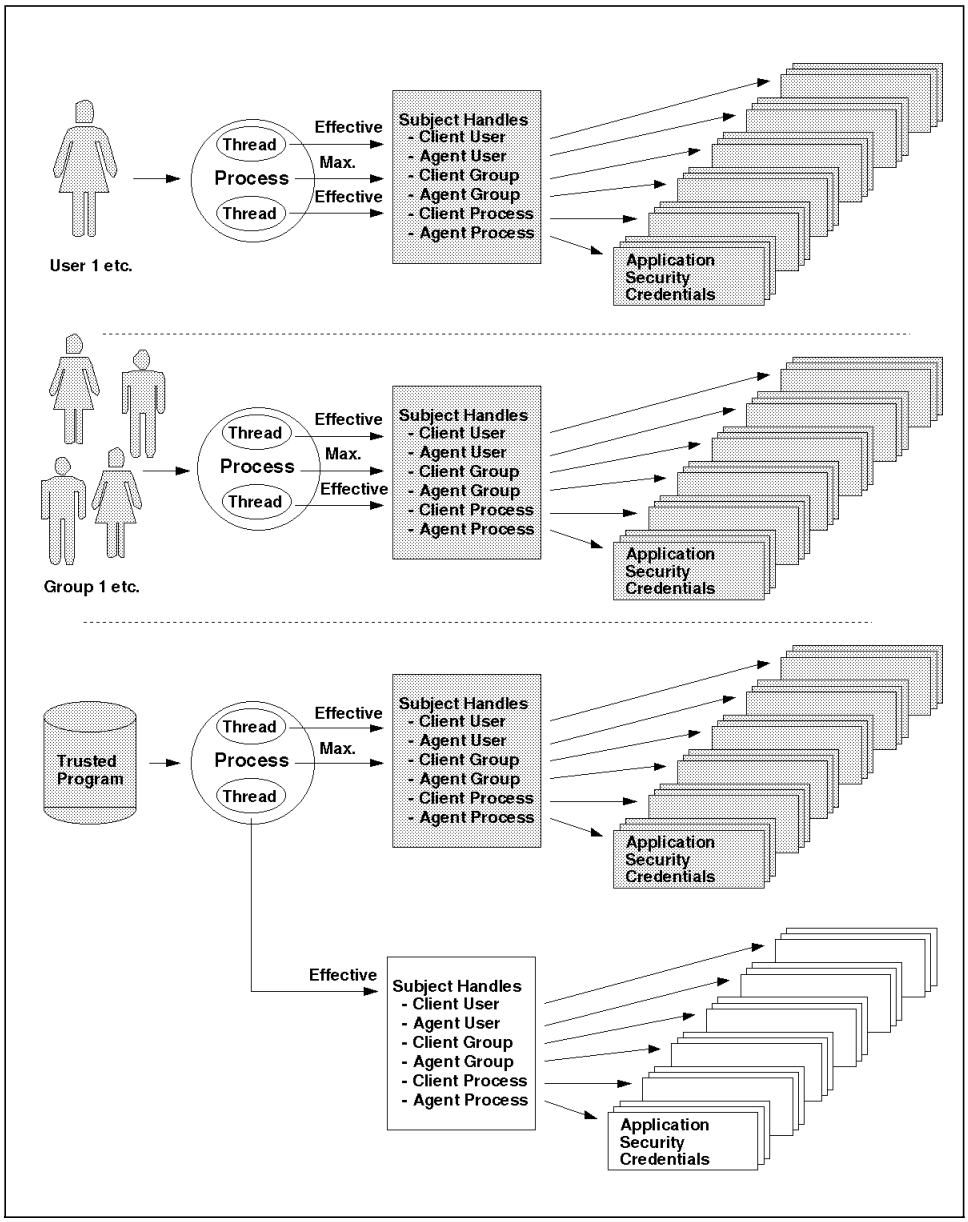



*Figure 11. SCS - Thread Context Model*

### 4.3.1 Multiple Concurrently Active Security Applications

SCS must enable the interoperation of multiple concurrently active security applications. For example, a distributed database manager could be installed on an OS/2 workstation that is secured by an ISS. The database manager might need to be able to start processes that can access files protected by the ISS, and the ISS might need to be able to create processes that can access records in the database protected by the database manager.

To enable the interoperation of the ISS and other security applications, SCS must do the following:

- Enable each security application to maintain its own definition of the security credentials associated with a process/thread. This is accomplished by associating each process/thread with a set of handles that each security application can then associate with its own definition of security credentials.
- Allow each security application to invoke the SES functions required to provide its security services (some SES functions can only be invoked by trusted security applications). Trusted security applications are recognized by SCS as having the authority to invoke privileged SES functions by maintaining the authority status of the security application in the security context of the trusted process. Each special authority that a security application might need to invoke privileged SES functions is associated with an authority flag in the security context of each process.

  **Note:** Any process that has an authority flag set in its security context is generically referred to as a security context authority. A process with a specific authority flag set in its security context is referred to by that flag name plus ″Authority″, for example: System Logon Authority (SLA), Access Control Authority (ACA), etc.

The services provided by security applications can be conceptually divided into three groups of functions that require special SES privileges (authority), as follows:

1. Establishing the association between a user′s security credentials (user identifier, group membership, administrative privileges, etc.) and the processes executing on behalf of the user. The first of these requires the identification and authentication of the user, and then the association of the user′s credentials with security context of the user′s processes.

   - System Logon Authority (SLA):

     A process that has the authority to establish the process-user association for the local system user. The local system user is

defined as the user who is associated with the OS/2 user interface services, that is Presentation Manager and Workplace Shell (PM/WPS).

- User Identification Authority (UIA):

  A process that has the authority to identify and authenticate a user for local system logon (the association of the user's credentials with the OS/2 PM/WPS user interface services is accomplished by the SLA as described above).

- Remote Logon Authority (RLA):

  A process that has the authority to establish the process-user association for a user who is not the local system user, that is a remote user who is accessing the local system through some interface other than the OS/2 PM/WPS user interface services, for example a user dialing in to the system through a TCP/IP connection.

  **Note:** An RLA is also responsible for identifying and authenticating remote users, that is UIAs can only identify/authenticate users for local system logon through the SLA.

2. Enforcing resource access control and audit policies for processes executing on behalf of a user. These policies may be enforced for local objects/services and for remote objects/services.

   - Access Control Authority (ACA):

     A process that controls access to (typically local) resources/services based on the security context established by an SLA or RLA.

   - Client Logon Authority (CLA):

     A process that controls access to (typically remote) resources/services based on its own authentication of a user, for example: Novell's client services that provide access to remote Novell servers. To enable the perception of a single signon for the local system user, the CLA can access the authentication information (for example: userid and password) provided by the user during local system logon.

3. Providing services for a client process that require the agent or server process to act on behalf of the client security credentials (in addition to acting on behalf of the agent/server security credentials).

   - Agent Process Authority (APA):

     A process that can act on behalf of a client's security credentials and/or its own (typically more privileged) security credentials.

- Server Process Authority (SPA):

  A process with threads that can act on behalf of multiple clients, security credentials and/or its own (typically more privileged) security credentials.

### 4.3.2 Multiple Concurrently Active Users

SCS must enable multiple concurrently active users. Although the primary target environment for OS/2 is the serial multi-user workstation with a single local keyboard,mouse, and display, support for multiple concurrently active users is required in the following environments:

- Background (Detached) Program/Process:

  An ISS may want to allow some processes to continue executing on behalf of one user (in the background) while another user is logged on. For example, a print spooler could run as a background process executing on behalf of one user at the same time that another user is logged on.

- Trusted Program/Process:

  A trusted program/process can execute on behalf of a trusted user who is not necessarily the user who invoked the program. For example, a change password program can be invoked by an untrusted user but could execute on behalf of the trusted user who has the authority to update the password database.

- Multi-User Application Server:

  A multi-user application server can execute on behalf of multiple client users through a client/server protocol. For example, a distributed database manager, a file/print/application server, or a multi-user shell that supports multiple user I/O streams (such as an X-Protocol I/O application server).

The key design point for the multi-user support is associating each process with the user on whose behalf the process is executing. SCS provides functions to associate a user (name) with a subject handle and to associate a subject handle with a process. The following scenario describes how SCS functions could be used by an RLA and an ACA process to associate a user′s credentials (user identifier, group membership, administrative privileges, etc.) with the user′s processes.

- The RLA establishes an association between a user (name) and a unique subject handle. When the user′s shell process is initiated, the RLA establishes the association between the user′s subject handle and the

user′s shell process. This association is inherited by all child processes of the user′s shell process.

- When the RLA establishes an association between a user (name) and subject handle, the ACA can be notified of the association. The ACA can then create the appropriate user credentials to be associated with the handle.
- When the ACA intercepts a request for access to a protected object, the ACA can retrieve the subject handle associated with the requesting process and use the associated user credentials to perform the access control check.
- When the last process referencing a subject handle is terminated, the ACA can be notified. The ACA can then delete the user credentials associated with the subject handle because it won′t be used again during this system boot.

### 4.3.3 Trusted Program/Process

A trusted process can be defined as a process that has the authority to act on behalf of a user other than the user who invoked the process. That is, the process has the authority to transition from executing with the privileges of one user to executing with the privileges of another user.

This privilege transition mechanism satisfies the following two key requirements:

- A trusted process must be able to control access to private data. For example, a database manager could be invoked by any client user to update records in the database. The database manager might need the ability to execute on behalf of the client user, but might also need to execute on behalf of a trusted agent user that has the authority to update the database files (even though the user who invoked the database manager doesn′t have the authority to modify the database files directly).
- A trusted server process must be able to impersonate client processes. For example, a multi-user application might need the ability to assume the security context of its client processes so that when it accesses resources protected by an ACA, the ACA will enforce the access control policies for the client′s user/group/process credentials (not the server′s user/group/process credentials).

SCS enables an ISS to implement trusted program support by associating each process/thread with client and agent user handles. The effective user handle can be set equal to either the client or agent user handle. In addition to supporting the association of each process/thread with client/agent user

handles, SCS must also support the association of each process/thread with client/agent group handles and with client/agent process handles. The effective group/process handle can be set equal to either the client or agent group/process handle. Two security context authority (SCA) roles are defined to satisfy the above requirements, without granting the trusted process unlimited super user powers:

- Agent Process Authority (APA):

  Enables a trusted process to execute on behalf of an untrusted client (for example, the user who invoked the process) and a trusted agent (for example, the owner of the trusted program).

- Server Process Authority (SPA):

  Enables a trusted process to have multiple threads that execute on behalf of different untrusted clients (who are not necessarily the user who invoked the process) and a trusted agent (for example, the owner of the trusted program).

## 4.4 Logon Shell Services (LSS)

A key requirement for SES is to enable the perception of a single signon in customer environments where user resources may be stored on the user′s local workstation and may also be stored on a variety of remote servers. The local workstation resources may be protected by local workstation security services that require identification and authentication (I&A) of the local workstation user, and each remote server may be protected by the server′s security services that also require I&A of client users. Users want the ability to enter their I&A information (for example, name & password) one time, and have this information accepted by all of the local/remote I&A mechanisms. LSS enables the perception of a local workstation user logging on to multiple local/remote services through a single signon event. LSS accomplishes this by coordinating the interoperation of the various security components that need to participate in the logon event to perform I&A of the local workstation user for access to:

- Local PM/WPS user interface services
- Local workstation resources protected by the ISS
- Other local/remote resources protected by other security services

In addition to enabling the perception of a single signon event, LSS enables the use of alternative authentication mechanisms (for example smart cards) and coordinates the interoperation of security components for other events related to a logon session:

- Logoff, shutdown
- Lock, unlock
- Identification and authentication (I&A)
- Change password, create user profile, delete user profile

PLEASE NOTE:

- IBM′s strategic solution for the single signon requirement is the third-party authentication protocol defined by the DCE (Distributed Computing Environment) technology. LSS facilitates the DCE strategy (and other single signon strategies) by enabling the interoperation of security components (for example: ISS and DCE).
- LSS does not provide generic single signon (I&A) services itself. It simply enables cooperating components to work together to provide the perception of single signon.

  SES does not:

  – Support DCE or GSSAPI logon services
  – Include any user registry/database
  – Include any authentication mechanisms
  – Include any services to synchronize user IDs, passwords, etc.
  – Include any facility to associate local users with security application credentials, so there is no logon notebook or personal logon facility to associate a local system user name with a remote server′s domain/userid
- LSS does not provide generic logon services for multiple local/remote users. LSS only provides logon services for the one and only local system user.

  To emphasize this point:

  – LSS functions can only be invoked for the local system user.
  – LSS events depend on the state of the local system logon session.
  – LSS event flows are integrated with PM/WPS services for the local system user.

  LSS functions can only be invoked by a process that can communicate with the local system user through PM/WPS user interface services. Security components participating in LSS events assume that they can

communicate with the user who invoked an LSS function via PM services (WPS may or may not be active depending on the state of an event).

For example, a security component that needs to authenticate a user for local system logon will most likely communicate with the user through PM services (for example through a dialog box). Consequently, LSS functions can only be invoked by a process that can communicate with the user through PM/WPS services. For example, a communications product that supports remote dial-in to an OS/2 workstation cannot invoke LSS functions for the remote user.

The following sections describe:

- The requirements of key security components that must cooperate to provide single signon support for the local system user
- The operational requirements for interaction between these key components to provide an integrated state machine for events related to the local system user′s logon session

### 4.4.1 Overview of Key LSS Components

The following figure identifies key security components that participate in LSS local logon session events:

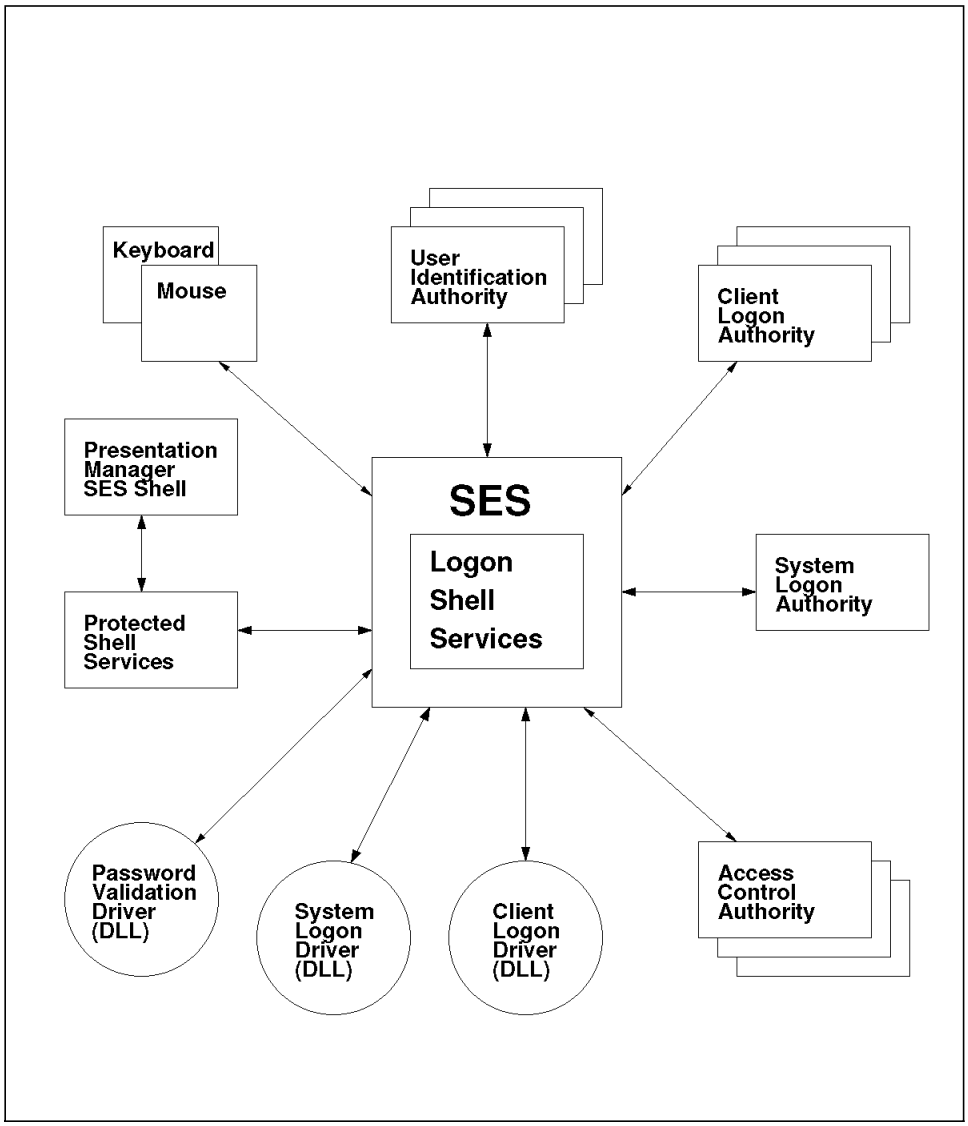



*Figure 12. LSS - Coordination of Logon Session Events*

#### 4.4.1.1 LSS Policy DLLs

To enable the perception of single signon in environments where multiple local/remote I&A is required, these I&A services can be divided into three major functions:

1. I&A for access to resources on the local system (local system logon)
2. I&A for access to remote resources where the server doesn′t accept local system I&A (client logon)
3. Administration of user authentication information (for example userid and password)

The LSS policies for these functions are encapsulated in DLLs that can be replaced by ISV security products or customers to satisfy specific security requirements.

***System Logon Driver:*** The System Logon Driver (SLD) defines/enforces the policy for SES interaction with UIAs during a local system logon event (and other events related to a local system logon session), so the SLD determines which UIAs participate in identifying and authenticating a local system (PM/WPS) user. The SLD is called by SES for the logon, unlock, identification and authentication, change password, create user profile and delete user profile operations.

**Note:** The default policy for the SLD supplied with SES is to process the UIAs in the order they appear in the SECURE.SYS file, and to stop when the first UIA returns either success or failure (it will only go on to the next UIA if the current UIA returns an error condition).

***Client Logon Driver:*** The Client Logon Driver (CLD) defines/enforces the policy for SES interaction with CLAs during a local system logon event (and other events related to a local system logon session), so the CLD determines which CLAs participate in events related to a local system logon session. The CLD is invoked by SES for the logon, logoff, lock, unlock, change password, create user profile and delete user profile operations.

**Note:** The default policy for the CLD supplied with SES is to process all of the CLAs listed in the SECURE.SYS file in the order they appear in the file. The return code is ignored.

***Password Validation Driver:*** The Password Validation Driver (PVD) defines/enforces the policy for creating/changing a local system user′s password. The PVD validates that a new password satisfies specified rules (for example, minimum length, composition rules, dictionary check, etc.).

The PVD is invoked by SES for the change password and create user profile operations.

**Note:** The default policy for the PVD shipped with SES is to allow any password.

#### 4.4.1.2 LSS Event Daemons

An LSS event (for example local system logon) requires cooperation between SES (the SES Daemon, PSS Daemon, SESShell Daemon, and SES Device Driver) and the various security components that must participate in the event. For example, for a logon event: UIAs identify & authenticate the local system user, the SLA establishes the security context for the local system user, CLAs provide single signon services for the local system user, and SES coordinates the whole event.

To enable interaction between SES and the cooperating security components, each security component must provide a daemon process that registers with SES and waits to respond to LSS events when invoked by LSS. These daemon processes require special SES privileges to participate in LSS events.

**Note:** No special privilege is required to start LSS events. For example, a smart card device could detect the insertion of a smart card and start a local system logon event without any special privileges. However, it would probably be better to include security applications (such as a smart card device that could provide I&A for local workstation users) in the set of cooperating LSS components so that they are aware of the current state of the local system logon session and can act accordingly (for example: logon versus unlock).

***User Identification Authority:*** LSS interacts with UIAs to provide I&A for the local workstation user for the logon, unlock, and I&A events. Each UIA invoked returns the status of its authentication attempt (for example: success, failure, error, etc.). The status from each UIA is processed by the SLD and the final user authentication status is determined by the SLD, which it returns to LSS.

***System Logon Authority:*** LSS interacts with the SLA to apply its security policy for logon, logoff, lock, unlock, and shutdown.

For logon and unlock, the SLA receives the final status determined by the SLD (based on the results of the UIAs that participated in the logon/unlock event). Given the results of the local I&A, the SLA can do one of the following:

- Return a status to indicate that the logon/unlock event failed.
- Return a status to indicate that the logon/unlock event succeeded. If the event is a logon, the SLA would create the local security context for the user.
- Return a status to indicate a Guest logon/unlock event. If the event is a logon, the SLA would create the local security context for the Guest user.

For lock, logoff and shutdown events, the SLA first receives a query event notification. The SLA may prompt the user to confirm the event. The SLA may then cancel the event or allow lock/logoff/shutdown to continue. The SLA will, if the event was not cancelled, then receive a lock/logoff/shutdown event notification. At this point, the SLA will perform any processing required for the requested event, for example with logoff, all processes that are running on the users behalf will be terminated.

**SLA** One reason for the query logoff/shutdown event notification is to give the SLA an opportunity to confirm that the user wants to take this action before it's gone too far. Also, for the lock event, the SLA may not want a Guest user to lock the system.

***Client Logon Authority:*** After local system logon/unlock, CLAs are provided the name and password entered by the local user during the local system logon/unlock event.

**Note:** The password is not permanently stored anywhere. It is maintained in kernel memory during the local system logon session so that CLAs may use it to provide the perception of single signon.

If names/passwords are synchronized between the local and remote systems, the CLA may be able to log the user on to the appropriate remote services using the local name/password without any further intervention by the user. If names/passwords are not synchronized, the CLA will need to obtain the correct name/password from the user.

**Note:** To expedite the logon process, CLAs should return status to LSS as soon as possible (before actually attempting to log the user on to a remote server). The intent is just to notify the CLA of the local system logon so that it can prepare to log the user on when needed (hopefully without any further intervention by the user).

#### 4.4.1.3 LSS Keyboard/Mouse Device Driver Support

LSS provides a trusted path service that enables a user to invoke the services of an ISS through a special key combination (for example: Ctrl-Alt-Del) that cannot be intercepted by applications. When the trusted path service is invoked, the ISS can take control over keyboard/mouse input and can ensure that the user's input is routed directly to the ISS.

When no local system user is logged on or the local system user interface services (PM and WPS) are in a locked state, a logon or unlock event must be initiated to start local system logon or to unlock the user interface services.

- If the trusted path service is not configured, LSS has control over user keyboard/mouse input and initiates a logon/unlock event when user activity (keystroke or mouse button) is detected.
- If the trusted path service is configured, LSS will not initiate a logon/unlock event when user activity is detected. When a user invokes the trusted path service, the ISS can take control over keyboard/mouse input and can initiate a logon/unlock event as appropriate.
- Whether the trusted path service is configured or not, user activity other than keyboard/mouse input can be detected and a logon/unlock event can be initiated by security applications. For example, a smart card reader could detect insertion of a smart card into the reader and initiate a logon/unlock event.

In addition to detecting trusted path invocation or user activity, LSS detects keyboard/mouse inactivity (keystroke or mouse button) to automatically lock the user interface services after a specified time period. This facility detects keyboard/mouse inactivity independent of the state of the user interface services (without regard to what screen group is active etc.).

### 4.4.2 Overview of Key LSS Operations

LSS supports the integration of multiple cooperating security components to handle local system logon session events (for example: logon, logoff, lock, unlock, etc.). The local system logon session refers to the time period between a successful logon event and a successful logoff event. During a local system logon session, the security context established for the local workstation user is associated with PM, the user shell (for example: WPS), and all untrusted processes created on behalf of the local workstation user. This security context will be referred to as the Local System Logon Session security context.

The requirement to support the integration of multiple cooperating security components to handle local system logon session events involves a fairly complex state machine. In addition, LSS supports several optional modes of operation:

- Trusted Path Support:

  1. LSS monitors keyboard/mouse activity to initiate logon/unlock events.
  2. ISS trusted path services initiate logon/unlock events (in lieu of LSS).

  This optional mode of operation is specified by a CONFIG.SYS environment variable: TRUSTEDPATH=NO|YES (default is NO).

- User Shell Process Handling for Logon/Logoff:

  1. Let the user shell continue to execute between local system logon sessions.
  2. Terminate/restart the user shell between local system logon sessions.

  This optional mode of operation is specified by a CONFIG.SYS environment variable: RESTARTUSERSHELL=NO|YES (default is YES).

- Guest Logon Support:

  1. Requires user to explicitly initiate a Guest logon event
  2. Automatically starts a Guest logon event (without explicit user action)

  This optional mode of operation is specified by a CONFIG.SYS environment variable: AUTOGUEST=NO|YES (default is NO).

To describe how the requirement to support these optional modes of operation impacts the technical requirements for the LSS state machine, we need to loosely define four of the key LSS local system logon session states:

- Explicit Logon State

  PM and the user shell process are active. PM and the user shell process are associated with the security context of the user (either an authenticated user or a Guest user) for whom a local system logon event was explicitly initiated (although the user doesn't necessarily need to take an overt action to initiate logon).

  For example:

  – LSS can initiate logon when keyboard/mouse activity is detected.
  – The ISS can initiate logon as part of its trusted path services.
  – A smart card device can initiate logon when a smart card is inserted.

Note that in this case, the SLA establishes the security context for the local system logon session. The security context associated with the local system logon session in the Explicit Logon State will be referred to as the Explicit Logon State security context.

- Auto-Guest Logon State

  PM and the user shell process are active. PM and the user shell process are associated with the security context that is specified for the default unauthenticated user as the result of LSS automatically initiating an Auto-Guest logon event.

  Note that in this case, the SLA does not establish the security context for the local system logon session. The security context associated with the local system logon session in the Auto-Guest Logon State will be referred to as the Auto-Guest Logon State security context.

- Lock State

  PM and the user shell process are active, but the user interface services (keyboard, mouse, display) are not available until an unlock event is initiated, that is keyboard/mouse input to applications is disabled and the display is covered with a customer-specified bitmap.

  Note that in this case, the security context associated with the local system logon session is not changed from what was established during the local system logon event. Consequently, the security context associated with the local system logon session in the Lock State is either the Explicit Logon State security context or the Auto-Guest Logon State security context.

- Logoff State

  PM is active, the user shell may or may not be active (depending on the mode of operation), and the user interface services are not available until a logon event is initiated. PM and the user shell (if active) are associated with the security context that is specified for this state.

  Note that in this case, the SLA may optionally define a security context for PM and the user shell (if active) when no local system user is logged on. The security context associated with PM and the user shell (if active) when no local system user is currently logged on will be referred to as the Logoff State security context.

The diagram in Figure 13 on page 58 shows a simplistic view of how the LSS state machine transitions between these states as a result of LSS events.

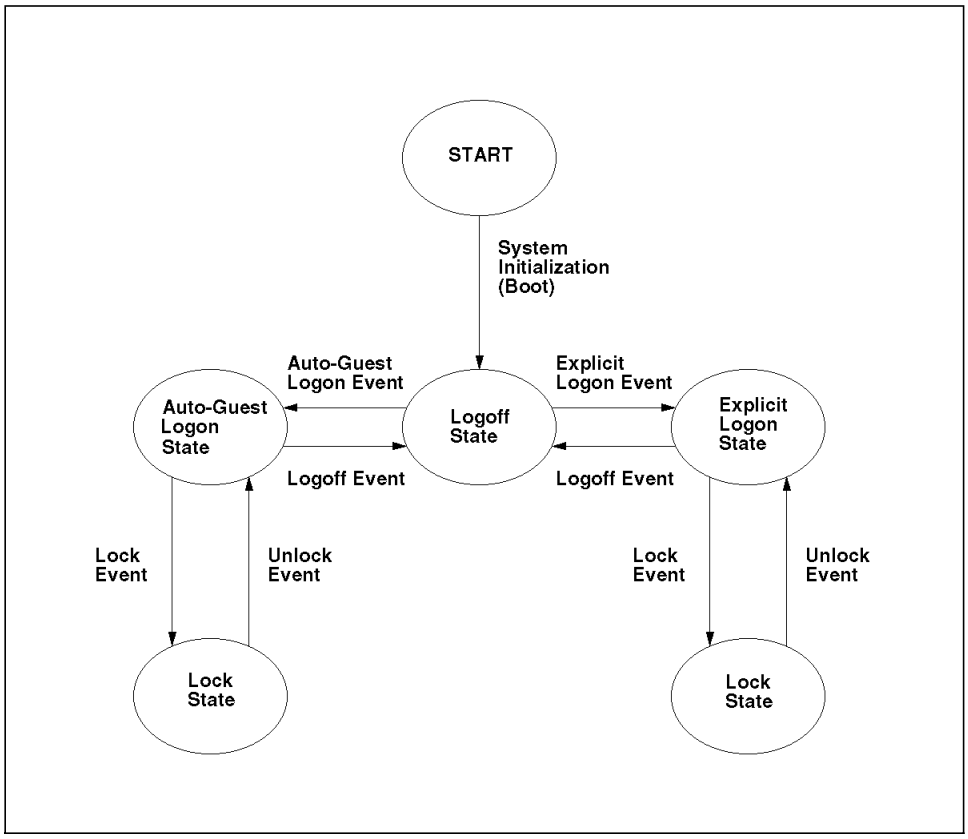



*Figure 13. LSS - Overview of Logon Session Events*

Please note there is no direct transition from the Auto-Guest Logon State to the Explicit Logon State (or vice-versa). To transition from one logon state to another requires a logoff event followed by a logon event. This may seem obvious, but it can cause some operational constraints that are not necessarily obvious (or desirable). The same is true for transitions between the Logoff and Lock states, that is direct transitions between these two states are not supported.

### 4.4.2.1 Initialization

Here we provide an overview of initialization to show the impact of the optional LSS modes of operation on system initialization requirements.

- DEFAULT OPERATION (TrustedPath=No, RestartUserShell=Yes, AutoGuest=No)

  1. The SES device driver and ISS security kernel (device driver) are loaded and initialized.
  2. The SES Daemon is started.
  3. The SESShell Daemon is started, which initializes PM.

     **Note:** The SESShell Daemon is the PM process.
  4. The PSS Daemon is started.
  5. The SESShell Daemon starts the security application daemons (for example: SLA, UIA, CLA, etc.) if configured.
  6. The initialization process is suspended until the SLA registers with SES (and optionally specifies a Logoff State security context). When the SLA registers with SES, the initialization process is allowed to continue.
  7. The LSS state machine ends up in the Logoff State, the user shell is not active and the user interface services are not available until a logon event is initiated.

- TRUSTEDPATH=YES

  System initialization is not affected by this option.

- RESTARTUSERSHELL=NO

  The user shell (for example: WPS) will be started during initialization, after SES, ISS, PM, SLA, etc. are ready, but before a logon event.

  1. No change.
  2. No change.
  3. No change.
  4. No change.
  5. No change.
  6. No change.
  7. LSS starts the user shell.

8. The LSS state machine ends up in the Logoff State, the user shell is active, but the user interface services are not available until a logon event is initiated,

- AUTOGUEST=YES

  An Auto-Guest logon event will be initiated as the last step of initialization.

  1. No change.
  2. No change.
  3. No change.
  4. No change.
  5. No change.
  6. No change.
  7. LSS starts the user shell if RESTARTUSERSHELL=NO.
  8. LSS initiates an Auto-Guest logon event (see next section for overview of logon).
  9. If the logon event is successful, the LSS state machine ends up in the Auto-Guest Logon State.

     **Note:** If the logon event is not successful, LSS ends up in the Logoff State.

#### 4.4.2.2 Logon/Logoff

Here we provide a Logon/logoff overview to show the impact of the optional LSS modes of operation on LSS state machine requirements.

- DEFAULT OPERATION (TrustedPath=No, RestartUserShell=Yes, AutoGuest=No)

  Logon:

  1. When the LSS state machine is in the Logoff State and the user shell is not active, LSS detects keyboard/mouse activity and initiates an explicit logon event.
  2. The UIAs are invoked (as determined by the SLD) to authenticate the local system user.

3. The SLA is invoked to establish a security context for the local system user.
4. If the logon event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services and the user shell is started.

   **Note:** If the logon event is not continued by the SLA, LSS returns to the Logoff State.

5. The LSS state machine ends up in the Explicit Logon State.

Logoff:

1. When the LSS state machine is in the Explicit Logon State, a logoff event is initiated.
2. The SLA is invoked to determine whether the local system user wants to continue with the logoff event or not.
3. If the logoff event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services and the user shell is terminated.

   **Note:** If the logoff event is not continued by the SLA, LSS returns to the Explicit Logon State.

4. The LSS state machine ends up in the Logoff State.

- TRUSTEDPATH=YES

  LSS will not initiate an explicit logon event when keyboard/mouse activity is detected while in the Logoff State (the assumption being that the ISS trusted path services will initiate an explicit logon event as a result of trusted path invocation).

  Logon:

  1. When the LSS state machine is in the Logoff State and the user shell is not active, LSS ignores keyboard/mouse activity. However, a trusted path invocation is detected and the ISS trusted path services initiate an explicit logon event.
  2. No change.
  3. No change.
  4. No change.
  5. No change.

The logoff operation is not affected by this option.

- RESTARTUSERSHELL=NO

  LSS will not terminate/restart the user shell between logoff/logon events.

  Logon:

  1. When the LSS state machine is in the Logoff State and the user shell is active, an explicit logon event is initiated.
  2. No change.
  3. No change.
  4. If the logon event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services. The user shell is already active.

     **Note:** If the logon event is not continued by the SLA, LSS returns to the Logoff State.
  5. No change.

  Logoff:

  1. No change.
  2. No change.
  3. If the logoff event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services. The user shell is not terminated.

     **Note:** If the logoff event is not continued by the SLA, LSS returns to the Explicit Logon State.
  4. No change.

- AUTOGUEST=YES

  LSS will automatically start an Auto-Guest logon event (without explicit user action) whenever the LSS state machine is in a Logoff State, with one key exception to allow a user to explicitly logon (either as an authenticated user or as an explicit Guest user) when the LSS state machine is in the Auto-Guest Logon state.

With this option, the LSS state machine is essentially always in a logon state (either Auto-Guest or Explicit), except for the brief time between logon states while LSS transitions through the Logoff State. Consequently, when a user wants to explicitly log on, a logoff event must be processed first.

To make this as painless as possible for the ISS, LSS handles the Auto-Guest Logon State a little differently from the Explicit Logon State:

- LSS allows an explicit logon event to be initiated while the LSS state machine is in the Auto-Guest Logon State, and automatically initiates an implicit logoff event before proceeding with the explicit logon event.

  **Note:** When the LSS state machine is in the Auto-Guest Logon State, the ISS should provide a convenient user interface for the Auto-Guest user to initiate an explicit logon (for example when adding a logon icon to the desktop or adding a logon menu item to the desktop context menu).

- LSS doesn′t allow a logoff event while the LSS state machine is in the Auto-Guest Logon State (since LSS would immediately start an Auto-Guest logon and return to Auto-Guest Logon State), except for the implicit logoff initiated by LSS for an explicit logon.

  **Note:** When the LSS state machine is in the Auto-Guest Logon State, the ISS should not provide a user interface for the Auto-Guest user to initiate a logoff.

Logon:

1. When the LSS state machine is in the Auto-Guest Logon State, an explicit logon event is initiated.
   a. LSS suspends the explicit logon event and initiates a logoff event.
   b. When the logoff event is completed, LSS allows the explicit logon event to continue.
2. No change.
3. No change.
4. If the logon event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services. The user shell may already be active or may be started at this time, depending on the RESTARTUSERSHELL option.

**Note:** If the logon event is not continued by the SLA, LSS goes to the Logoff State; however, an Auto-Guest logon is initiated immediately, so the LSS state machine ends up in the Auto-Guest Logon State.

5. The LSS state machine ends up in the Explicit Logon State.

Logoff:

1. No change.
2. No change.
3. If the logoff event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services. The user shell may or may not be terminated, depending on the RESTARTUSERSHELL option.

   **Note:** If the logoff event is not continued by the SLA, LSS returns to the Explicit Logon State.

4. The LSS state machine goes to the Logoff State, however, an Auto-Guest logon is initiated immediately, so the LSS state machine ends up in the Auto-Guest Logon State.

#### 4.4.2.3 Lock/Unlock

This is a lock/unlock overview to show the impact of the optional LSS modes of operation on LSS state machine requirements.

- DEFAULT OPERATION (TrustedPath=No, RestartUserShell=Yes, AutoGuest=No)

  Lock:

  1. When the LSS state machine is in a logon state (Auto-Guest or Explicit), a lock event is initiated.
  2. The SLA is invoked to determine whether the local system user wants to continue with the lock event or not.
  3. If the lock event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services.

     **Note:** If the lock event is not continued by the SLA, LSS returns to the logon state.

  4. The LSS state machine ends up in the Lock State.

Unlock:

1. When the LSS state machine is in the Lock State, LSS detects keyboard/mouse activity and initiates an unlock event.
2. The UIAs are invoked (as determined by the SLD) to re-authenticate the local system user.

   **Note:** During the processing of an unlock event for a Guest user (either Auto-Guest logon or explicit Guest user logon), the unlock event cannot require authentication; however, the lock event can be used as a screen saver function for Guest users.
3. The SLA is invoked to re-establish the security context for the local system user.
4. If the unlock event is continued by the SLA, the CLAs are invoked (as determined by the CLD) to provide single signon services.

   **Note:** If the unlock event is not continued by the SLA, LSS returns to the Lock State.
5. The LSS state machine ends up in a logon state (Auto-Guest or Explicit).

- TRUSTEDPATH=YES

  LSS will not initiate an unlock event when keyboard/mouse activity is detected while in the Lock State (the assumption being that the ISS trusted path services will initiate an unlock event as a result of trusted path invocation).

  The lock operation is not affected by this option.

  Unlock:

  1. When the LSS state machine is in the Lock State, LSS ignores keyboard/mouse activity. However, a trusted path invocation is detected and the ISS trusted path services initiate an unlock event.
  2. No change.
  3. No change.
  4. No change.
  5. No change.

- RESTARTUSERSHELL

  The lock/unlock operations are not affected by this option.

- AUTOGUEST

  The lock/unlock operations are not affected by this option.

## 4.5 Installation, Configuration and Initialization Support (ICIS)

The ICIS support for SES with OS/2 version 2.11 is very straightforward in the following areas:

- Installation

  The SES component for OS/2 version 2.11 is distributed as part of the OS/2 service stream, and is available with Fixpak XR_B100.

  The SES installation process simply copies the necessary files to the appropriate drive/directories. No modifications to CONFIG.SYS to enable SES are made during the SES installation process. When a customer installs a security product (ISS) that requires SES, the ISS installation process will make the appropriate modifications to CONFIG.SYS (and SECURE.SYS) to enable SES features.

- Configuration

  All SES configuration is accomplished by modifying the CONFIG.SYS and SECURE.SYS files. No GUI or APIs are provided for configuration.

- Initialization

  ISV security products typically provide boot protection which ensure continuous protection prior to OS/2 initialization. However, during OS/2 initialization (prior to the ISS being able to enforce its security policy), OS/2 ensures that the system is protected from unauthorized intervention. There are three phases of the initialization process that must be considered:

  1. Prior to processing of CONFIG.SYS, the user can interrupt initialization by typing ALT-F1, the ISS must be able to ensure that this interruption cannot be used to circumvent its security policies.
  2. During processing of CONFIG.SYS, SES and ISS components must be loaded and functional prior to allowing a user to log on. If not, the

ISS must be able to cause a default state where only an authorized administrator can intervene.

3. During processing of SECURE.SYS, SES ensures that security applications are assigned authorized privileges and that no applications are assigned unauthorized privileges.

# Chapter 5. Installable Security Subsystem

An ISS is a set of components that provides the security features for a secured OS/2 system. In this chapter we discuss the relationship between SES, an ISS and the security dependant applications that must work together in a secured OS/2 workstation. The diagram in Figure 14 illustrates this relationship.

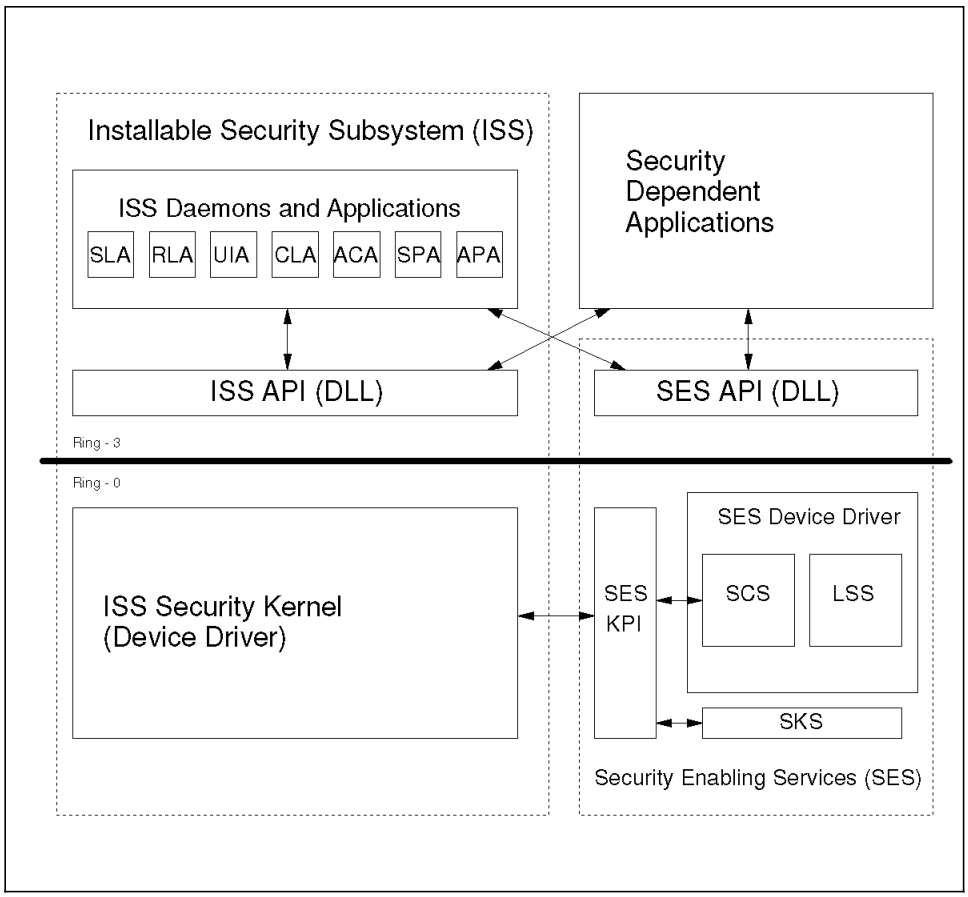



*Figure 14. ISS - Overview of a Secured OS/2 System*

## 5.1 What Is an ISS?

An ISS is a set of components that provides the security features for a secured OS/2 system. An ISS may contain components that perform or support I&A (such as password checking), DAC (such as file access control), system audit, single signon, trusted program support etc.

## 5.2 What Are the Typical Components of an ISS?

The components of an ISS will vary, depending upon the security features that the ISV needs to add to the OS/2 system to satisfy the customer set. An ISS may include the components depicted in the picture in Figure 15 on page 71.

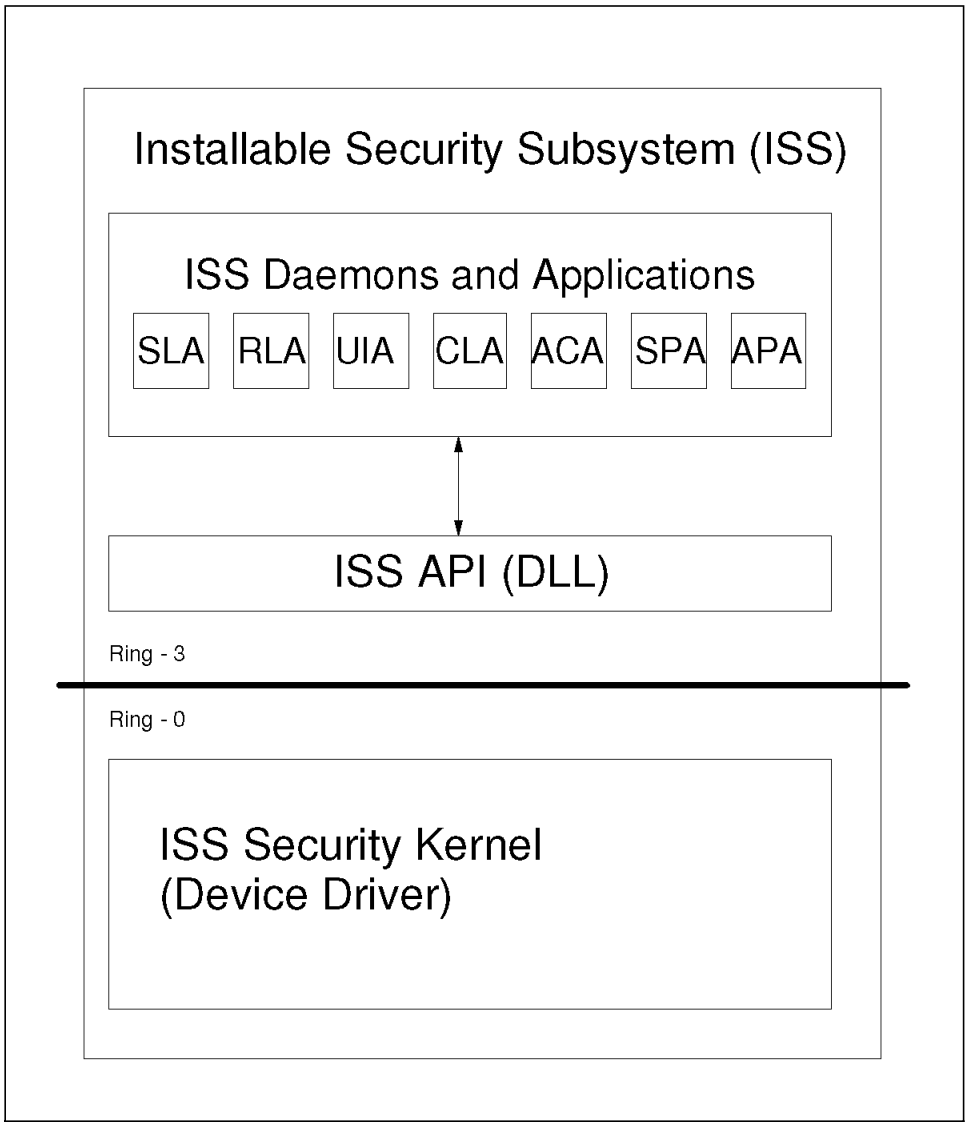



*Figure 15. ISS - Components*

- **Security daemons and applications**

  The application level components of an ISS are what the customers see. This is the user interface and/or application program interface. By exploiting the SES trusted process support, much of the logic required to implement ISS security features can be built as trusted ISS applications.

  **Note:** From an OS/2 customer's point of view, there is no distinction between a security daemon and a security application; both are a set of one or more software programs/components that provide a service to the customer.

  The SES definition of a security daemon is an application/program that executes as a trusted process with special SES privileges and interacts with SES via a captive thread that is blocked in the SES device driver until SES needs the security daemon to respond to a security relevant event.

  The ISS security daemons and applications may need special SES privileges to perform their responsibilities. The following table shows which privileges might be applicable to the various services provided by an ISS.

| Privileged ISS Daemon/Application | ISS service |
|---|---|
| Access Control Authority (ACA) | Resource Access Control |
| Agent Process Authority (APA) | Trusted Program Support |
| Client Logon Authority (CLA) | Client/Server Support |
| Remote Logon Authority (RLA) | Client/Server Support |
| System Logon Authority (SLA) | Local System Logon |
| Server Process Authority (SPA) | Trusted Program Support |
| User Identification Authority (UIA) | I&A (Local System Logon) |

*Table 3. Suggested Mapping of SES Privileges to ISS Functions*

- **Security kernel (device driver)**

  The function of the security kernel is to permit the ISS to establish communication with the OS/2 kernel. This enables the ISS to receive security event information from the kernel. Without a security kernel (implemented as an OS/2 device driver), an ISS cannot enforce its security policy on OS/2 kernel operations.

Typical OS/2 kernel operations include file actions such as open, read, write, close, change file pointer, delete, process creation, etc. Theseactions are referred to as security events. Notification of the occurrenceof selected security events is sent from the OS/2 kernel to the ISS security kernel, if the ISS has indicated that it wishes to receive the notification.

- **Dynamic link libraries**

  Three special DLLs are defined in the OS/2 security enabling services. Each has a well defined function in the processing of security events and each provides a default policy for processing security events. These DLLs, andthe corresponding policy for processing security events, can be replaced byISS DLLs:

  – System Logon Driver (SLD)

    The SLD determines the order in which UIAs are invoked during logon.

  – Client Logon Driver (CLD)

    The CLD determines the order in which CLAs are invoked during logon.

  – Password Validation Driver (PVD)

    The PVD verifies that a password issued during a change password request satisfies specified rules (composition, history, dictionary, etc.).

  In addition, the ISS can provide its own APIs, with which a security dependent application can invoke the security functions provided by the ISS. This is an optional component of an ISS.

  A trusted application is an external security component which depends upon the security services provided by the secure OS/2 operating system. It won't necessarily have been developed by the ISV who developed the ISS, but it may need to use services provided by the ISS. Access to these services is provided through the APIs supplied with the ISS.

  Why would an ISS want to provide APIs? By providing an API, a customercan develop security dependent applications that can invoke the security services provided by the ISS. This adds significant value to an ISS formany large OS/2 customers.

## 5.3 What Support Does SES Provide for an ISS?

SES provides an operational environment for the ISS with well defined security services and support.

### 5.3.1 Security Context

SES maintains a security context for each OS/2 process/thread. The security context contains subject handles that are associated with user/group/process credentials and information about privileges and status of the process/thread. The important point to note here is that each process/thread is associated with a security context that denotes its privileges, status, credentials, etc.

There are several ways the security context of a process/thread can be established/changed:

- Inherited from its parent process.
- Specified as having special SES privileges.
- Established by trusted components with appropriate SES privileges.
- Modified through SES APIs.
- Established by an ISS during system logon.

  **Note:** System logon is defined as associating a user's security context with the OS/2 user interface services (Presentation Manager and Workplace Shell).

### 5.3.2 Privileges and Authorities

A process that is active under OS/2 may be granted a specified set of SES privileges. These privileges control the execution environment of the process and determine what SES functions the process may access. Each privilege defined by a separate flag in the process/thread is security context, and is referred to as the security context authority (or authority for short). A program/process/thread that has one or more of these privileges is referred to as a Security Context Authority (SCA).

Since the processes are identified by the authority roles they may assume, they are often referred to by the name of the authority. For example, aprocess which has assumed the role of access control may be referred to as an ACA. Processes may assume multiple authority roles, depending upon the functions they need to perform. The roles are assigned during system initialization, in response to information the ISV adds to a system file named SECURE.SYS.

#### 5.3.2.1 ACA, SLA and UIA

The ACA/SLA/UIA roles are defined primarily to support local workstation security services.

- **ACA (Access Control Authority)**

  An ACA establishes the rules for access to protected objects. It is invoked each time an initial access request is made. It must have access to the credentials of the requester and the access conditions for each protected object.

  This is a means of implementing DAC in OS/2. For example, an ISS may contain an ACA which controls access to selected system files. The rules contained within the ACA would control requester access based upon the user′s credentials and the access control established for the file. The access control could be implemented in the form of an ACL and could restrict access to the file for a single user or a group of users.

  The specific privilege defined for an ACA is the right to register for notification of subject handle creation and deletion.

- **SLA (System Logon Authority)**

  An SLA determines whether or not a user should be allowed to logon to the local system and helps create the credentials which associate the user with the local system logon environment. An ISS may define/manage the user/group/process credentials that are associated with the process security context (set of subject handles, security context authority flags, and related status information). The SLA can associate the user with the processes executing for that user.

  The minimum set of authorities that an ISS should have are UIA and SLA. With these authorities, an ISS can grant a user access to the OS/2 system (associate the user′s credentials with the user′s system logon session), and can grant access to authorized workstation resources. Note that the ISS security daemon can be both a UIA and an SLA.

- **UIA (User Identification Authority)**

  A UIA performs identification and authentication for users logging onto the system. The UIA takes the user logon information as input, and

returns a status of User-Authenticated or User-Not-Authenticated (or other error return codes) to SES. An ISS should provide at least one authentication mechanism and corresponding UIA.

Note that there may be multiple UIAs in a system, depending upon a customer's needs. For example, a system that uses 2 forms of I&A, such as a password and a smart card, might use 2 different UIAs to perform the I&A. The order in which the UIAs are invoked to authenticate the user is defined in the SLD.

#### 5.3.2.2 APA and SPA

The APA/SPA roles are defined for trusted program support, again primarily for local workstation security services.

- **APA (Agent Process Authority)**

  An APA executes on behalf of a client user (who may be untrusted) and a trusted agent. When an APA program is invoked by a user, the APA swaps the user's set of security privileges (called the security context) for the user's process to a set of security privileges which can perform a security sensitive operation. When the requested operation is completed, the APA is terminated.

- **SPA (Server Process Authority)**

  An SPA executes on behalf of multiple client users and a trusted agent. An SPA maintains one or more threads that run with the user's security context rather than the SPA's. Each time a different user request is processed, the requester's security context is used to determine access rights.

APA and SPA are provided in order to allow a program the capability of being able to transition from executing with the privileges of one user, to executing with the privileges of another (pseudo) user. This allows a trusted program to execute on behalf of an untrusted client under certain controlled conditions. For example, a DBMS program might need to be accessible and process transactions on behalf of trusted and untrusted clients.

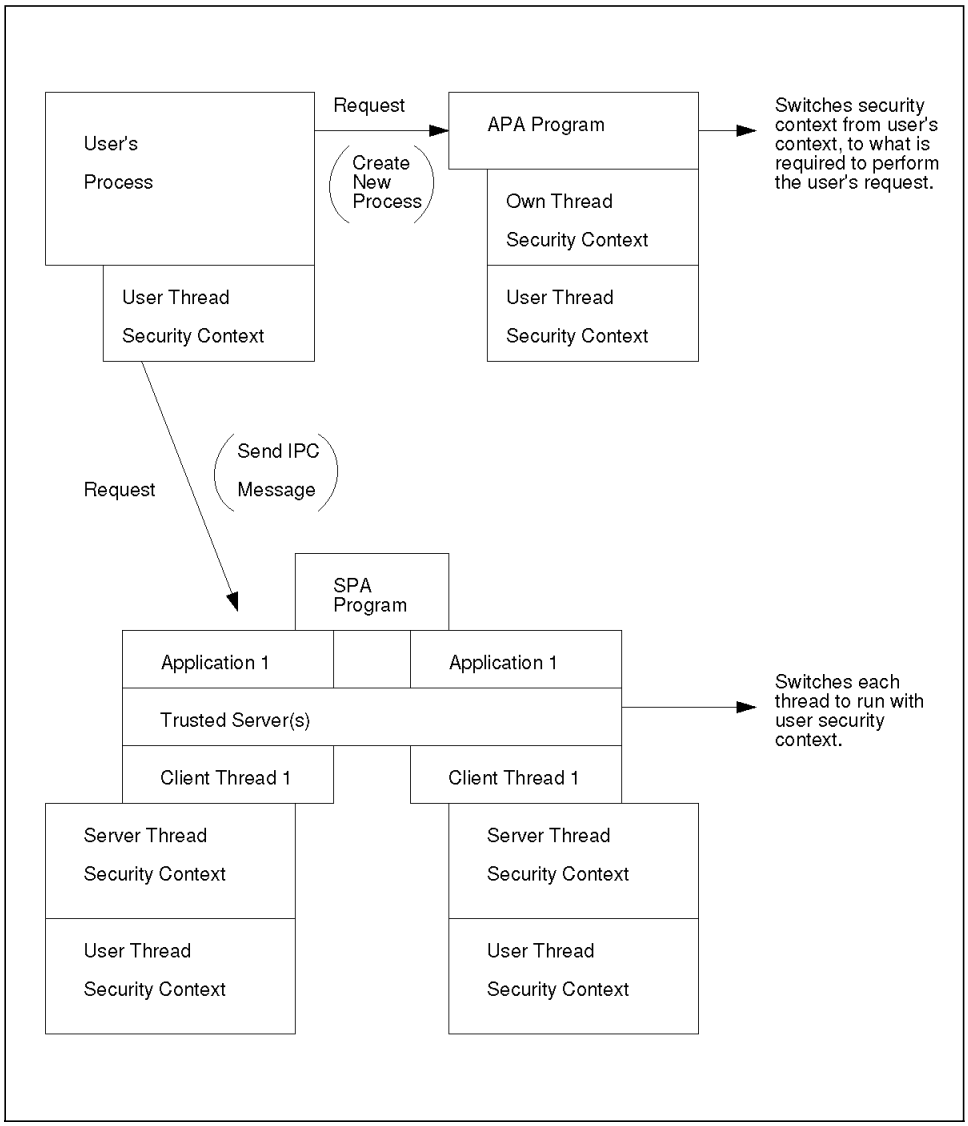



*Figure 16. ISS - APA and SPA*

#### 5.3.2.3 RLA and CLA

The RLA/CLA roles are defined primarily to support distributed (client/server) computing environment security services.

- **RLA (Remote Logon Authority)**

  RLA provides process-user association for remote system users and local system processes. For example, a remote user who wishes to logon to a local machine (perhaps through Telnet), would be authenticated to the local system by an RLA. The functions of an SLA and an RLA are similar. The actual association of the user with the processes executing for the user is performed by an RLA for remote users.

- **CLA (Client Logon Authority)**

  A CLA authenticates the local user onto a remote system. A CLA gathers up whatever information is necessary and propagates it to some remote node/server/etc. for authentication. If the remote authentication fails, the user may still be locally authenticated. Depending upon the ISS application, a CLA could also have UIA authority. There may be multiple CLAs in a system; the CLD determines the order in which they are called.

  A CLA is not needed on a stand-alone workstation, but should be present on client workstations in a client/server environment to facilitate the perception of single signon during the local system logon process.

#### 5.3.2.4 Interoperation of Security Context Authorities

How do the security context authorities (UIA, SLA, CLA, RLA, ACA, etc.) worktogether? Consider what happens when a user initiates a logon request.

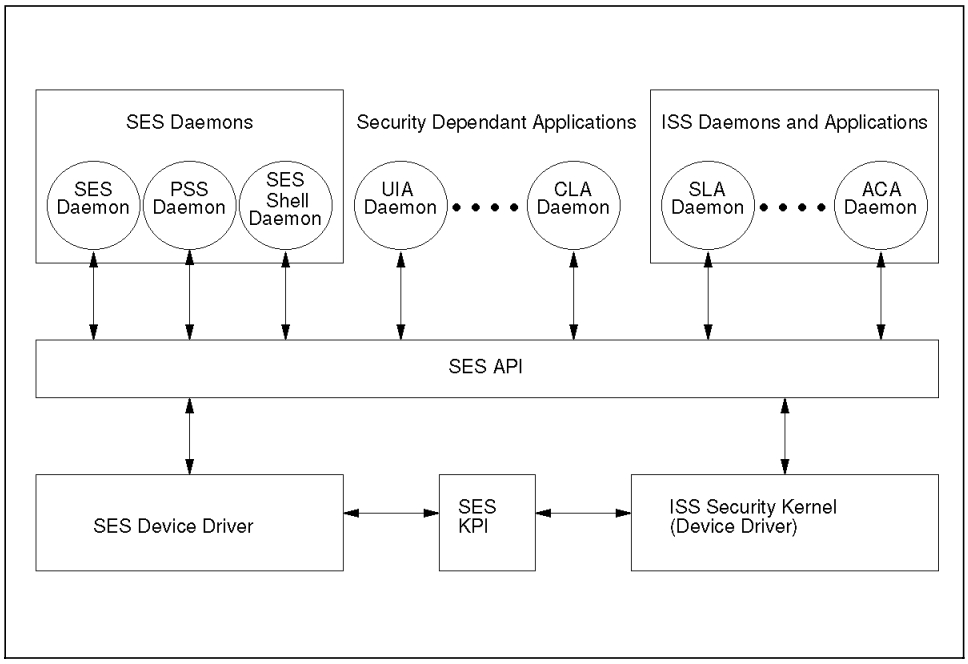



*Figure 17. ISS - Interoperation of Security Context Authorities*

After the logon request is made by the user, the SLD is queried to determine the order in which to notify system UIAs. The UIA obtains the user name and password (any Identification and Authentication mechanism could be used), and performs authentication of the user. The information is passed back to the system, and the SLA is notified of the logon attempt. The SLA applies the logon policy that has been defined by the ISV, and determines if system logon is to be performed. If it is, SES creates the security context (as defined by the SLA) to associate with the user′s credentials for the local system logon environment (typically PM and WPS).

The CLD is queried to determine the order in which to call CLAs. The appropriate CLA obtains the user logon information and propagates it to a remote resource for authentication and subsequent access to the remote resources. If authentication is successful, the user obtains access to the remote services.

The RLA receives a request to logon from a remote server, and performs authentication as needed. If authentication is successful, SES creates the security context (as specified by the RLA) to associate the user′s credentials with the process tree(s) created for the user by the RLA. When an ACA receives a request for access to protected resources from one of these

processes, the ACA can query the security context for the requesting process and apply the appropriate access control policy.

### 5.3.3 Programming Interfaces

ISS applications must interact with predefined OS/2 services at an application and a kernel level. OS/2 provides programming interfaces for application (API) and kernel (KPI) level components. The API provides security developers a means to create security applications; the KPI provides security developers a means to create ISS device drivers. For example, the ISS security kernel (device driver) communicates through defined KPIs to the kernel level security enabling services.

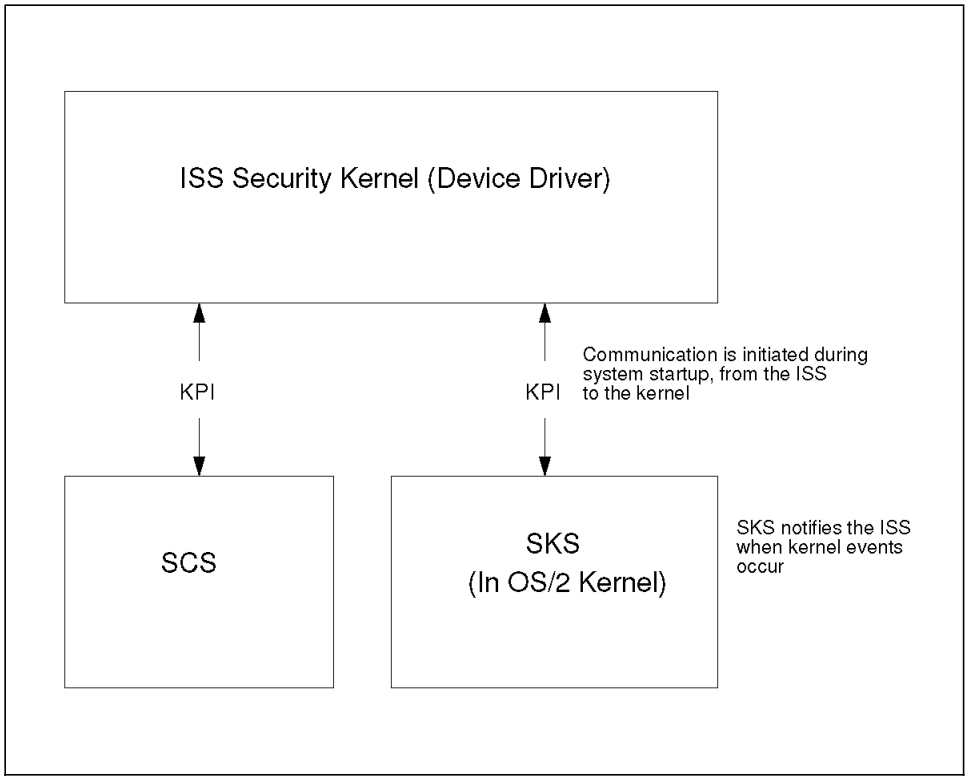



*Figure 18. ISS - Kernel Programming Interface*

Through the APIs and the KPIs, an ISS can create, delete, reserve, and examine handles for processes and threads, control processes, wait for events, determine the order of execution for specific authorities, and receive kernel level event information. APIs are made available to ISVs as Dynamic

Link Libraries (DLLs), which are loaded into the ISS through C language calls.

## 5.4 ISS Summary

OS/2 security is provided by an ISS, building on the services provided by OS/2 SES. The minimum set of components necessary for an ISS depends upon the services provided by the ISS, but would typically include:

- A security kernel (to enforce security policies at the OS/2 kernel level by interacting directly with the OS/2 kernel and SES security context services)
- A security daemon with UIA and SLA privileges (UIA to identify and authenticate local system users, SLA to establish the security context for local system logon), perhaps also with ACA privileges (for notification of subject handle creation/deletion)

ISVs can create unique applications which provide I&A, DAC, audit, single signon, or trusted program functions for the OS/2 system. These applications can be granted a defined set of privileges (ACA, SLA, UIA, RLA, CLA, SPA and APA). An application may be granted multiple privileges.

An ISS can include DLLs to replace/augment SES policies. The SLD and CLD DLLs determine the order in which UIAs and CLAs are called by the system. The PVD DLL validates password composition. The default SES DLLs (SLD, CLD and PVD) may be replaced by an ISS.

User credentials and handles are accessible to the ISS. Programming interfaces, called APIs and KPIs, are provided to enable the ISS to invoke the OS/2 security enabling services.

# Glossary

## A

**Access**. An interaction to obtain information from any source or to communicate.

**Access control**. A security mechanism to protect access to programs or information, for example RACF.

**Access method**. The technique or program used for moving information or communicating.

**Access time**. The time from issuing a command to read or write to a file on disk until the physical read or write is actually carried out.

**Access object**. Object of the rights administration on which subjects access. Access objects are generally files and interfaces.

**Administration object**. Rights administration object which contains rules for several users. The administration object can be assigned to domain, user groups, users, workstations and trusted programs.

**Adapter card**. A printed circuit card for attaching input/output devices to a computer.

**Address**. In data communication , the unique code assigned to each device or workstation connected to a network.

**AIX**. Advanced Interactive eXecutive: IBM′s version of UNIX.

**Algorithm**. A set of rules to solve a problem in a number of steps.

**API**. Application Program Interface. Interface through which programs can communicate.

**ASCII**. American National Standard Code for information interchange: a coded character set used on personal computers.

**Authenticate**. A process to verify the integrity of data or a message, or to verify the user of an information system or protected source.

**Authentication**. Confirmation of a given identity, using password, smart card or ID token.

**Authentication server**. Part of a trusted security base. Responsible for authenticating identities of clients. Maintains passwords and group membership information for users.

**Authorize**. Granting someone the right to use a computer, application or database; is also used in connection with programs to grant complete or restricted access to an object, resource or function.

**Audit**. Logging of user actions for audit purposes.

**Auditor**. Role with regard to rights administration or log evaluation. DP auditor, responsible for auditing DP systems.

## B

**Batch file**. On a personal computer, a file having the extension .BAT, which contains a list of commands that are executed when the file is called.

**Bitmap**. (1) An area of memory or storage that contains the pixels representing an image, arranged in the sequence in which they are normally scanned, to display the image.
(2) A representation of an image by an array of bits.

**BIOS**. The area of the computer that controls incoming and outgoing signals.

**Boot**. The process of starting up a personal computer.

**Boot drive**. Logical drive from which the operating system is loaded. Generally it is the disk drive (C). It can, however, also be a floppy drive (A).

**Boot protection**. Prevention of a system start from a medium other than the hard disk (or boot ROM). A system start with an operating system diskette is prevented.

**Bus**. In a processor, a physical facility to transfer data; for example, ISA, MCA. Adapter cards are connected to a bus.

**Byte**. A string that consist of a number of bits, treated as a unit, and representing a character.

# C

**CBC mode**. Cypher Block Changing. DES encryption mode in which not only the key, but also the last encrypted 8 byte are used for encryption.

**CCA**. IBM Common Cryptographic Architecture; the IBM architecture for the cryptographic Application Programming Interface (API).

**Client/Server system**. A client/server system is a local network where PCs are connected to a server.

**CD-ROM**. Compact Disc Read Only Memory. A Compact disc specifically for storing data.

**CD-ROM XA**. Compact disk read-only-memory extended architecture. A partial implementation of CDI and DVI standards.

**Channel**. A connection between a personal computer and one or more input/output devices.

**Checksum**. The sum of a group of data, used for checking purposes.

**Cipher**. A cryptographic system.

**Clip board**. A temporary storage area used to pass information within a program or from a program to another.

**CKDS**. Cryptographic Key Data Set: a file containing cryptographic keys.

**CMOS**. A chip technology that requires little power, used to store vital configuration data of a PC.

**COM**. Serial interface for data communication. Is used to connect a modem, for example.

**Confidentiality of data**. Protection against unauthorized reading. Can only be achieved by encryption.

**Configuration**. (1) An arrangement of physical or logical devices that make up a (sub)system. (2) The manner in which the hardware and software of an information processing system are organized and interconnected.

**Control vector**. In TSS: a 16 byte string that modifies the master key or a key-encrypting key to create another key that is used to encipher and decipher data or data keys.

**Controller**. A device that controls the operation of input/output devices.

**Conventional memory**. Random Access memory in a PC that DOS or OS uses as the first 640K byte.

**CRC**. Cyclic Redundancy Check. Checksum which is not cryptographic.

**Cryptography**. The principles and methods for encrypting plain text and decrypting cipher text to conceal its meaning.

**CVC**. Cryptographic checking of the contents of a magnetic stripe of a credit card (Mastercard).

**CVV**. Cryptographic checking of the contents of a magnetic stripe of a credit card (VISA).

## D

**Data encrypting key**. A key used to encipher, decipher or authenticate data.

**Data integrity**. Data intactness. The intactness is checked by an integrity check (checksum method).

**DEA**. Data Encryption Algorithm. Method for encrypting data using a 64 block cipher that uses a 64 bit key, including 8 parity bits.

**Decipher**. To convert encrypted text (cipher text) into the original text (plain text).

**DES**. Data Encryption Algorithm. Developed and published in 1977 by IBM. A US standard for encrypting data, available in two versions, full DES and commercial DES. Standard algorithm used by international banks. Symmetrical algorithm with a 56 bit long key. The CBC mode is regarded as secure.

**Device**. A physical unit of a computer system, often used for input/output operations, which can be used in a logical order or have a logical address.

**Directory**. A hierarchically structured logical area for storing files on a hard disk or diskette, which may include one or more sub-directories.

**DOS**. Disk Operating System: an operating system for personal computers.

**Domain**. Organizational unit which is commonly managed. Also known as system.

## E

**EBCDIC**. Extended Binary Coded Decimal Interchange Code: a coded character set used on main-frames.

**Emulator**. Imitator.

**Encipher**. To convert an original text (plain text) into encrypted form (cipher text).

**Encrypt**. Synonym of encipher.

**ESS**. Establish Secure Session; a cryptographic means by which hardware components establish.

**Extended attribute**. The OS/2 method of attaching additional information to a file object. Extended attributes can be used to store notes on file objects ( for example version, history), categorize file objects *for example file type, associations), describe the format of data contained in the file object, or append additional data to the file object. They are stored separately from the file object they are associated with and are managed by the file system attached to the file object.

**Extended data**. User-defined information, including multimedia information, about Light Table folder objects. Such information goes beyond what is available in OS/2 standard data. Extended data includes user-defined columns, and may come either from a supported database or from extended attributes.

**Extension**. In the name of a file, the three letters following the dot, which often indicates the type of file, for example, BAT in AUTOEXEC.BAT indicates batch file.

## F

**Folder**. A directory as represented on the OS/2 desktop.

**Font**. The characters available for text with a given set of attributes.

## G

**Generation**. The number of copies away from the original.

**Graphical User Interface (GUI)**. A type of computer interface consisting of a visual metaphor of a real world scene, often of a

desktop. Within that scene are icons representing objects, that the user can access and manipulate with a

**Graphic**. Any pictorial representation of information.

**Graphics**. Text or pictorial artwork created by a variety of means, such as electronic generated graphics software and the pressed onto the video-discs.

## H

**Hertz**. A measure of frequency equivalent to cycles per second.

**Host**. A host is a ″large″ computer which acts as a ″host″ for terminals or workstation PCs with terminal function.

**HPFS**. High Performance File System. HPFS provides long file name support and fast access to very large disk volumes.

## I

**I&A**. Identification and Authentication; see identification and authentication.

**Icon**. A pictorial representation of a function that you can select to carry out this function.

**Identification**. Identification of a DP user to the system with a user ID (Name or Personal-No).

**ID Token**. Checkcard-sized special calculator with a keypad to generate a dynamic password according to the Challenge/ Response method.

**Interface**. Hardware and/or software that links systems, programs, or devices.

**I/O**. Input/Output: pertaining to a device that performs input and/or output operations.

**IPL**. Initial Program Load; the initialization of a computer.

**ITSEC**. Information Technology Security Criteria. Criteria book issued by the EU, worked out by 4 member states - France, The Netherlands, UK and Germany. Evaluations to determine security are made on the basis of ITSEC. The products are awarded a certificate if they meet the requirements. F-C2, E2 means that the security functions of class F-C2 and the evaluation level E2 were reached. F-C2, E2 is the commercial standard.

**Image**. A still picture or one frame.

**Interlace**. The technique of using more than one vertical scan to reproduce a complete image. In television, a 2:1 interlace is used, giving two vertical scans per frame. One scan will be odd lines, the other will be even lines.

## K

**KB**. kilobyte: 1024 bytes.

**Kilohertz (kHz)**. Thousands of cycles per second.

**KEK**. Key Encrypting Key; a key used to encrypt, decrypt or authenticate keys for transmissions.

**Key**. In computer security, a sequence of symbols used with a cryptographic algorithm for encrypting or decrypting data.

**Key administration**. Administration program which administers the keys and restores them if they are destroyed.

**Key token**. In TSS, a data structure that can contain a cryptographic key, a control vector, and other information related to the key.

**KM**. Master Key: the top level key in a hierarchy of key encrypting keys.

**KMC**. Key Management Center; a department for managing cryptographic keys.

## L

**Local Area Network (LAN)**. A data network located on the user's premises in which serial transmission is used for direct data communication between workstations.

**Leading logon type**. The type of logon that must be done before other logons can be used.

**Link**. (1) A logical connection, (2) A physical connection, (3) An interconnection between data or programs.

**LPT**. Parallel interface to attach a printer, streamer, etc.

## M

**MAC**. Message Authentication Code. Cryptographic checksum, based on the DES-MAC. The encryption result, the last 8 encrypted bytes, is the checksum.

**Migrate**. (1) To move data from one storage media to another, (2) To change to a new operating environment.

**Module**. Program module which takes over a specific function. Example: logging in a linear file on the server, or logging in a local ring buffer file. Modules can be swapped by the system administrator.

**Multi-tasking**. A technique that allows several processes to appear to run simultaneously, even though the computer only has one CPU. This is achieved by sequentially switching the CPU between tasks.

**Multiplexer**. A device that interleaves the transmission of several input signals over a connection such that the input signals can be recovered.

## N

**Network**. (1) A network of devices and software connected for information interchange, (2) An arrangement of modes and connecting branches to interconnect computers, terminals and workstations.

**Node**. In a network , a point at which one or more units are connected. Each node has a network address.

## O

**Object**. (1) Resource of the DP system, such as files, interfaces, networks, etc. (2) A visual component of a user interface that a user can work with to perform a task.

**OEM**. Equipment sold by another manufacturer.

**Off-line encryption**. File encryption in a separate work process.

**On-line encryption**. Transparent encryption during the Read or Write process. On-line encryption in Safe Guard Professional can be both file and sector oriented.

**Orange Book**. Security standards from the US Department of Defense that specify different security levels.

## P

**Panel**. The set of information displayed on the screen of a display station.

**Password**. In computer security, a string of characters used to gain access to a computer file or system, during sign-on or at a later time. A PIN can be considered as a password.

**Path**. (1) In a network, any route between two nodes, (2) The route traversed by information exchanged between two network devices, (3) A

command in DOS related to the path through its (sub)directory structure to reach a file.

**Pause Function**. ″Logoff″ for a short work interruption. The screen is blanked, the keyboard can only be used for special entries and the work station is locked. To continue work, the user who triggered the pause, must log on again.

**PCF**. Programmed Cryptographic Facility: IBM program for enciphering and deciphering text and for key management.

**Pel**. Picture element. The smallest building block that a screen or bit-mapped image can display. Pel and pixel can be used interchangeably.

**Pixel**. A single point of an image, having a single pixel value.

**Pop-up**. A window which appears on the screen to display text, graphics, messages, or documents.

**PIN**. Personal Identification Number: The secret number that a user must remember to gain access to a service, can be used in conjunction with an IBM Personal Security Card.

**Plain text**. Non-encrypted data.

**Private key**. In computer security, a secret key used to encrypt data.

**Protocol**. Rules and agreements for communication between devices.

**PSC**. Personal Security Card: A standard smart card with a processor for executing DES-based cryptographic functions. It can hold more than 4000 bytes of data, including the characteristics of a signature for the purpose of verification. Incorporated. Currently supported by IBM.

**Public key**. In computer security, a widely known key used to encrypt data, the encrypted data must be decrypted with a related private key.

## R

**RAM**. Random Access Memory: Memory where data can be written and read directly.

**Reuse**. This is the recreation of the original status of a file, the main memory or the swap file after it has been deleted or after the user has logged off.

**Resolution**. The ability of an image reproducing system to reproduce fine detail.

**RGB**. A method of processing color images according to their red, green, and blue color content. Colors can also be measured on an HIS color scale. Contrast with composite, Y/C, and YUV.

**Role**. Role which the user plays, particularly with regard to rights administration. A role is assigned specific administrative rights, for example system administrator, auditor, accessory or simple user.

**ROM**. Read Only Memory: Memory to store programs or data permanently.

## S

**SAF**. System Authorization Facility: a program that provides access to RACF.

**SAPI**. Interface for application call via TSS hardware using special verbs for executing security information.

**Scanner**. A device which performs scanning.

**Schema**. The data-definition part of a database table.

**Sealing**. Sealing of data for purposes of the integrity check.

**Security**. The protection of data, system operations and devices from accidental or intentional ruin, damage or exposure.

**Security architecture**. IBM strategy and architecture for secure information systems.

**Security Enabling Services**. Add-on in OS/2, which acts as interface between ISV products and OS/2.

**Security server**. Server, which saves data important for the security of the security sub-system and carries out authentications.

**Security officer**. Role in rights administration. Executive responsible for DP security. Generally he is responsible for the initial installation or de-installation, as well as for key management.

**Server**. On a LAN, a station that provides services to other stations; for example, file server, print server, and security server.

**Session**. The period of time that a network connection lasts, including the establishment and release of the connection.

**Session key**. Key valid for only one session. A session is the time between logon and logoff.

**Signature verification**. In TSS an optional feature of the security interface unit for user verification through their signature, written with a signature verification pen and recognized by a signature verification module on the cryptographic adapter.

**Single SignOn (SSO)**. First logon to a network. The Safe Guard Professional SSO function enables the automatic logon to other computers in the network, after the user has authenticated himself once. Passwords are automatically synchronized.

**Software configuration utility**. One of the utilities distributed with the workstation security services program for configuring security servers and device drivers.

**Sub-directory**. A directory contained within another directory in the file system hierarchy.

**Subject**. A subject is a user or process which accesses objects (files etc.).

**Subsystem**. A secondary or subordinate system, usually capable of operating independently.

**System administrator**. User in a DP system who plays an important administrative role. He is responsible for the administration of rights.

# T

**Token**. Bit string (combination of bits) to enable the execution of a specific operation.

**Token-ring**. An IBM network with a ring topology that passes tokens from one attaching device to another.

**Trusted programs**. Programs, which either do not have the user ′s full access rights or whose rights go over and beyond these.

**Trusted workplace shell**. Workplace which corresponds to the individual rights profile of the user. It cannot be changed by the user.

**TSS**. Transaction Security System: A series of cryptographic products for providing a secure workstation.

# U

**User**. User in a DP system. In Safe Guard Professional a user generally does not have administration rights.

**User ID**. User identification, name for a user in the system.

## W

**Wildcards**. Placeholders for any number of other characters. An asterix (*) stands for a permitted set of any other characters. A question mark (?) stands for any other single character.

**Workstation**. A terminal or microcomputer that often is connected to a main frame or a network, at which the user can perform applications.

## X

**XGA**. Extended graphics array. A high resolution display with a display matrix (pels) of 1,024 x 768 at 256 colors. XGA can also provide more colors with reduced resolution (640 x 480 at 65,536 colors).

# List of Abbreviations

| | |
|---|---|
| **ACA** | Access Control Authority |
| **APA** | Agent Process Authority |
| **AIX** | Advanced Interactive Executive |
| **ANSI** | American National Standards Institute |
| **API** | Application Programming Interface |
| **APAR** | Authorized Program Analysis Report |
| **ASCII** | American National Standard Code for Information Interchange |
| **AT** | Advanced Technology |
| **BIOS** | Basic Input Output System |
| **BGA** | Business Graphics Adapter (8514/A card) |
| **BMP** | Bit-Mapped Graphics |
| **CKDS** | Cryptographic Key Data Set |
| **CLA** | Client Logon Authority |
| **CLD** | Client Logon Driver |
| **CMOS** | Complimentary Metal Oxide Semiconductor |
| **CPU** | Central Processor Unit |
| **CTCPEC** | Canadian Trusted Computer Product Evaluation Criteria |
| **CV** | Control Vector |
| **CVC** | Card Verification Code |
| **CVV** | Card Verification Value |
| **DAC** | Discretionary Access Control |
| **DBMS** | Database Management System |
| **DEA** | Data Encrypting Algorithm |
| **DES** | Data Encrypting Standard |
| **DIB** | Device Independent Bitmap |
| **DLL** | Dynamic Link Library |
| **DMA** | Direct Memory Access |
| **DOS** | Disk Operating System |
| **DoD** | Department of Defense (USA) |
| **EBCDIC** | Extended Binary Coded Decimal Interchange Code |
| **EC** | Engineering Chance |
| **EISA** | Extended Industry Standard Architecture |
| **ESS** | Establish Secure Session |
| **FAT** | File Allocation Table |
| **GUI** | Graphical User Interface |
| **HPFS** | High Performance File System |
| **IBM** | International Business Machines |
| **ICIS** | Installation, Configuration, Initialization Support |
| **ICRF** | Integrated Cryptographic Facility |
| **IMS** | Information Management System |
| **IPL** | Initial Program Load |
| **IRQ** | Interrupt Request |
| **ISA** | Industry Standard Architecture |
| **I&A** | Identification and Authentication |
| **ISO** | International Organization for Standardization |
| **ISS** | Installable Security Subsystem |
| **ISV** | Independent Software Vendors |
| **KB** | kilobyte |
| **Kbps** | kilobits per second |
| **KEK** | Key Encrypting Key |
| **KM** | Master Key |
| **KMC** | Master Key Center |
| **KPI** | Kernel Programming Interface |
| **LAN** | Local area network |
| **LSB** | Least Significant Bit |
| **LSS** | Logon Shell Services |

| | |
|---|---|
| **MAC** | Message Authentication Code |
| **MB** | megabyte (1,048,576 bytes) |
| **Mbps** | megabits per second |
| **MBps** | megabytes per second |
| **MC** | Micro Channel |
| **MCA** | Micro Channel Architecture |
| **MVS** | Multiple Virtual System |
| **NLS** | National Language Support |
| **OEM** | Other equipment manufacturer |
| **OS/2** | Operating System/2 |
| **PC** | Personal Computer |
| **PCF** | Programmed Cryptographic Facility |
| **PC AT** | Personal Computer Advanced Technology |
| **PC XT** | Personal Computer eXtended Technology |
| **PIN** | Personal identification number |
| **PM** | Presentation Manager (OS/2) |
| **POSIX** | Portable Operating System Interfaces for Computer Environments |
| **PSC** | Personal Security Card |
| **PS/1** | Personal System/1 |
| **PS/2** | Personal System/2 |
| **PVD** | Password Validation Driver |
| **RACF** | Resource Access Control Facility |
| **REXX** | Restructured EXtended eXecutor Language |
| **RISC** | Reduced Instruction Set Computer |
| **RLA** | Remote Logon Authority |
| **RPQ** | Request for Price Quotation |
| **RSCS** | Remote Spooling Communications Subsystem |
| **R/W** | read/write |
| **SAF** | System Authorization Facility |
| **SAPI** | Secured Application Programming Interface |
| **SCS** | Security Context Services |
| **SDK** | Software Developers Kit |
| **SES** | Security Enabling System |
| **SIU** | Security Interface Unit |
| **SKS** | Security Kernel Services |
| **SLA** | System Logon Authority |
| **SLD** | System Logon Driver |
| **SLE** | Session Level Encryption |
| **SNA** | System Network Architecture |
| **SPA** | Server Process Authority |
| **SRPI** | Server Requester Programming Interface |
| **SVGA** | Super Video Graphics Adapter |
| **SWF** | Security Workstation Feature |
| **TCB** | Trusted Computing Base |
| **TCSEC** | Trusted Computer System Evaluation Criteria (Orange book) |
| **TMK** | Terminal Master Key |
| **TSO** | Time Sharing Option |
| **TSR** | Terminate and Stay Resident |
| **TSS** | Transaction Security System |
| **UIA** | User Identification Authority |
| **UPM** | User Profile Management |
| **WAN** | wide area network |
| **VM** | virtual machine |
| **WOAM** | Work Place Shell Object Access |
| **WPS** | Workplace Shell |
| **XGA** | Extended Graphics Array |

# Index

## D



## E



## F



## G



## H

## I



## K



## L



## M

## N



## O



## P



## R

## S

## T



## U



## V

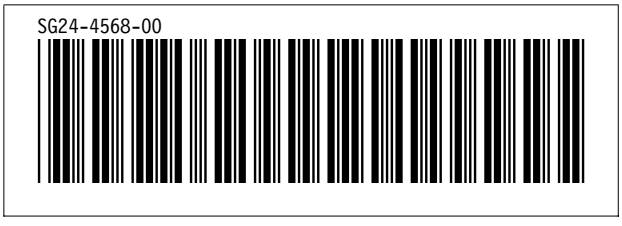


## W



## X

Printed in U.S.A.



## Artwork Definitions

| id | File | Page | References |
|---|---|---|---|
| ITSLOGO | 4568SU | i | i |

## Figures

| id | File | Page | References |
|---|---|---|---|
| SECDRV | 4568CH1 | 3 | 1 |
| CYCLE | 4568CH2 | 7 | 2 7 |
| BIGPIC | 4568CH30 | 26 | 3 26 |
| 8BPFW | 4568CH30 | 31 | 4 31 |
| 8BIGP2 | 4568CH40 | 34 | 5 33 |
| 8SKS1 | 4568CH40 | 36 | 6 35 |
| 8SCS1 | 4568CH40 | 38 | 7 |
| 8SCS2 | 4568CH40 | 39 | 8 |
| 8SCS3 | 4568CH40 | 40 | 9 |
| 8SCS4 | 4568CH40 | 41 | 10 |
| 8SCS5 | 4568CH40 | 43 | 11 |
| 8LSS1 | 4568CH40 | 51 | 12 |
| 8LSS2 | 4568CH40 | 58 | 13 57 |
| 8ISS1 | 4568CH50 | 69 | 14 69 |
| 8ISS2 | 4568CH50 | 71 | 15 70 |
| 8ISS3 | 4568CH50 | 77 | 16 |
| 8ISS4 | 4568CH50 | 79 | 17 |
| 8ISS6 | 4568CH50 | 80 | 18 |

**Headings**

## Index Entries

| id | File | Page | References |
|---|---|---|---|
| 8IOPCL | 4568CH30 | | |
| | | 22 | (1) Security architectures |
| | | | 22, 22 |

## Revisions

| id | File | Page | References |
|---|---|---|---|
| CLARIFY | ? | | |
| | | ? | ? |
| | | | 49, 50, 55, 57, 57, 57, 58, 58, 58, 59, 59, 60, 60, 63, 63, 63, 63, 64, 64, 66, 66, 66 |
| MODIFY | ? | | |
| | | ? | ? |
| | | | 57, 57, 59, 59, 63, 63, 63, 63 |

## Tables

| id | File | Page | References |
|---|---|---|---|
| 68CVO | 4568CH30 | | |
| | | 23 | 1 |
| | | | 23 |
| OS2VNT | 4568CH30 | | |
| | | 25 | 2 |
| | | | 25 |

## Processing Options

Runtime values:

| | |
|---|---|
| Document fileid | SG244568 SCRIPT |
| Document type | USERDOC |
| Document style | SDELIB |
| Profile | EDFPRF40 |
| Service Level | 0022 |
| SCRIPT/VS Release | 4.0.0 |
| Date | 95.10.05 |
| Time | 15:43:12 |
| Device | 3820A |
| Number of Passes | 4 |
| Index | YES |
| SYSVAR D | YES |
| SYSVAR G | INLINE |
| SYSVAR V | ITSCEVAL |
| SYSVAR X | YES |

Formatting values used:

| | |
|---|---|
| Annotation | NO |
| Cross reference listing | YES |
| Cross reference head prefix only | NO |
| Dialog | LABEL |

| | |
|---|---|
| Duplex | YES |
| DVCF conditions file | (none) |
| DVCF value 1 | (none) |
| DVCF value 2 | (none) |
| DVCF value 3 | (none) |
| DVCF value 4 | (none) |
| DVCF value 5 | (none) |
| DVCF value 6 | (none) |
| DVCF value 7 | (none) |
| DVCF value 8 | (none) |
| DVCF value 9 | (none) |
| Explode | NO |
| Figure list on new page | YES |
| Figure/table number separation | YES |
| Folio-by-chapter | NO |
| Head 0 body text | Part |
| Head 1 body text | Chapter |
| Head 1 appendix text | Appendix |
| Hyphenation | NO |
| Justification | NO |
| Language | ENGL |
| Keyboard | 395 |
| Layout | OFF |
| Leader dots | YES |
| Master index | (none) |
| Partial TOC (maximum level) | 4 |
| Partial TOC (new page after) | INLINE |
| Print example id's | NO |
| Print cross reference page numbers | YES |
| Process value | (none) |
| Punctuation move characters | ., |
| Read cross-reference file | (none) |
| Running heading/footing rule | NONE |
| Show index entries | NO |
| Table of Contents (maximum level) | 3 |
| Table list on new page | YES |
| Title page (draft) alignment | RIGHT |
| Write cross-reference file | (none) |

/XRL/5

**Imbed Trace**

| | |
|---|---|
| Page 0 | 4568SU |
| Page 0 | 4568VARS |
| Page 0 | 4568FM |
| Page i | 4568EDNO |
| Page ii | 4568ABST |
| Page xiii | 4568SPEC |
| Page xiii | 4568TMKS |
| Page xiv | 4568PREF |
| Page xviii | 4568ACKS |
| Page xx | 4568CH1 |
| Page 5 | 4568CH2 |
| Page 19 | 4568CH30 |
| Page 32 | 4568CH40 |
| Page 67 | 4568CH50 |
| Page 82 | 4568GLOS |
| Page 90 | 4568ABRV |
| Page 98 | 4568EVAL |