PART 6-D

With MVS/ESA Security Labels towards B1

“Why are manufacturers always announcing improved releases when all the customer is interested in is a stable and optimally functioning product?”
(The Version-X Release-Y Level-Z Question)

“You will never become a real security officer, you think too much.”
(M.F. van Hekken)

Abstract:

IBM’s OS/390 operating system supplemented by RACF has intentionally been designed to function as part of a Trusted Computer Base with B1 security. Among the enhancements there is full support for security labels for both subjects and objects, allowing to implement a mandatory access control policy and to use OS/390 systems as Multi Level Security systems. The evolution of the MVS and OS/390 security mechanisms ultimately resulting in the introduction of security labels will be discussed from the viewpoint of the technical EDP-auditor.
Introducing security in commercial electronic data processing is a process of decades, which turns out to become effective during the most recent years. In this paper we will show the move from the initial systems in the sixties and seventies, with none or minimal security, to our modern reliable and secure systems at the start of the nineties. Although the technical mechanisms and the knowledge are available to protect vital data from any unauthorized disclosure, modification and destruction, this does not imply that all vital data is now well protected nor that all information systems are now tamperproof. It is obvious that the effectiveness of security principally depends on the security attitude of the computing center and on how this center uses the mechanisms provided by the Trusted Computer Base (TCB). One should note that technical measures never can conceal or compensate organizational weaknesses. Both management of the company and computing center management have to define a security strategy and security objectives for the information systems, and have to implement and enforce technical and organizational measures for security and internal control. Moreover, active verification via the self assessment, peer review and formal audit are indispensable to guarantee that security will remain at the desired level.

The current paper presents the evolution of the security mechanisms in IBM’s Multiple Virtual Storage (MVS) systems in historical perspective, showing that the move to B1 security, as defined by USA Department of Defense (DoD), is a logical step in this evolution process. A review of both history and RACF 1.9 shows that there are no discontinuities, and that the functionality of RACF has steadily been extended until it matched the B1 picture as drafted by DoD. From the viewpoint of the technical EDP-auditor, we will concisely discuss this evolution, focusing on the new security labels for subjects and objects.

1 Resource Protection in Historical Perspective

IBM’s MVS operating system for the System/370 mainframes was a successor of the System/360 operating systems, and was principally designed to support commercial batch processing in a closed environment. Although the first version in 1976 already contained the Time Sharing Option (TSO) software, this interactive user interface still was dedicated to preparing batch jobs, submitting them and inspecting the results of their execution. In fact, TSO was subordinate to batch in those days. Due to the ‘closed shop’ characteristic and the emphasis on batch production, access control was not one of the major topics during the design of these early non-RACF systems. The security mechanisms initially integrated in System/370 and MVS were:

* Architectural support: both the hardware and the operating system allowed to make a distinction between authorized system software and unauthorized user programs (supervisor versus problem-program mode, key in storage, virtual versus real storage, address spaces, etc.). This architectural isolation still forms the foundation of modern access control.

* Logon to TSO: the end user had to provide a userid and a password. Both were stored in clear text in the system library SYS1.UADS, together with some administrative data such as logon procedure, account number, user profile, etc. Although it was possible to protect this data set against unauthorized access, many centers did not pay any attention to it. As a
consequence, their users could browse all members of this library reading all passwords, and even could change them via edit commands.

- Data set protection: the system data set PASSWORD could contain read and/or write passwords for data sets. Although this data set was protected by MVS, some installations were sloppy with their maintenance tools, sometimes allowing users to execute powerful, old-fashioned disk dump programs. For a malicious user it was then relatively easy to define the data set PASSWORD as input to this disk dump program and to create an unprotected copy, so that he could read all passwords.

- VSAM passwords: for data sets controlled by the Virtual Storage Access Method (VSAM) it was possible to define passwords which were stored in the catalogs. Their vulnerability was similar to that of the PASSWORD data set.

- Expiration date: one may define an expiration date for a data set. If a user attempts to update such a data set before the specified date, the system sends a message to the operator and asks him to grant or reject the access request for this data set. This mechanism was often used to protect system libraries against unauthorized updates. However, in centers with weak organizational controls this mechanism also turned out to be vulnerable to malicious users calling the operator or sending him a message asking to grant access to a system library when they wanted to modify it, and sometimes the operator did so.

Although there were some security mechanisms allowing to have at least some control over the users, many computing centers did not really use them due to lack of interest in security, or used them in such a sloppy way that they rendered ineffective. Again, these were the days of the ‘closed shop’ and the majority of interactive TSO users were well-known to computing center staff, so that there was still the aspect of social control which compensated the lack of effective technical measures and procedures in many centers.

1.1 Data Sets

Initially MVS already contained an interface for a security package, which was later filled in by Resource Access Control Facility (RACF). The first versions of RACF were designed to allow a gradual move from an unprotected to a protected environment, and only controlled those subjects (persons, users) and objects (resources, data sets) explicitly defined. For each subject and each object one had to create a RACF profile describing the authorities of the subjects, the access requirements for the objects, the relations, etc. In those days security was sometimes characterized as ‘nothing is protected unless explicitly specified’, especially because many centers did not bring all users and resources under RACF control. They often were satisfied when the most important users and resources were protected, and allowed the remaining users to work as non-RACF users and to access unprotected data sets.

Initially the data sets were described by discrete profiles, which means that for each data set one had to define a unique profile describing the access requirements. Moreover, it was necessary to set a bit in the Data Set Control Block (DSCB) in a disk’s Volume Table of Contents (VTOC) to indicate that a data set was RACF controlled. By activating Automatic Data Set Protection (ADSP) per user, one instructed the system to automatically generate a profile when this user created a new data set and to set the so-called VTOC bit. During OPEN
processing, the system tested this VTOC bit and invoked RACF if the bit was on, while the bit’s off status indicated that there was no need to call RACF. This mechanism was crippled by some shortcomings of the organization: sometimes it was relatively easy to switch off the VTOC bit intentionally or accidentally via maintenance tools, so allowing users to bypass the security checking for these data sets.

The increasing need for more security and, moreover, for reliable security less vulnerable to operational errors was understood by the designers of access control software, and resulted in developing new techniques and in the introduction of new mechanisms. For instance, in 1983 IBM announced RACF 1.5 and Data Facility Product (DFP) version 1.2 with the *Always Call* facility. Now RACF was *always* invoked during OPEN processing, regardless of the content of the VTOC bit. Moreover, the concept of discrete profiles, i.e. one RACF profile per data set, was extended with the introduction of *generic profiles* each describing a group of data sets. For instance, the generic name SYS1.* covers some hundreds of system data sets all having names starting with the high-level qualifier SYS1.

Although constituting a major improvement, Always Call and the concept of generic profiles did not provide the ultimate solution, because it was still possible to create unprotected data sets, i.e. new data sets with names which were not covered by the RACF profiles already existing in the system. In 1986 this was solved by introducing RACF 1.7 with the *Protect All* facility, controlled by the SET RACF OPTIONS (SETROPTS) command. The system administrator could issue the SETROPTS command with one of the following three parameters:

- **SETROPTS NOPROTECTALL**: the creation and usage of unprotected data sets is allowed and everybody may access them.
- **SETROPTS PROTECTALL(WARNING)**: if a user creates or accesses a data set for which there is no matching RACF profile, RACF sends a message to both the user and the console indicating that there is an unprotected data set. Still allowing creation and access, this is an intermediate step towards the option FAILURES, so supporting users to recognize unprotected data sets and recommending to protect them.
- **SETROPTS PROTECTALL(FAILURES)**: both creation of and access to unprotected data sets is inhibited by RACF. If one needs access to an ‘old’ unprotected data set a system or group administrator first has to define an appropriate profile describing the data set.

In modern systems with a full implementation of Protect All, all resources are protected by default and nobody can create unprotected resources. We so moved from the initial situation of ‘nothing is protected unless explicitly specified’ to a new situation, characterized by ‘everything is protected’.

### 1.2 Security Levels and Categories

Each company and each government agency use a classification to indicate the value and confidentiality of data, often ranging from ‘unclassified’ or ‘for internal use only’ to ‘secret’ or even ‘top secret’. This classification uses numbers or symbolic names representing *levels* ordered in a hierarchical way, which means that you may compare two levels to determine whether one is higher than, equal to, or lower than the other. In addition to assigning levels to
indicate the amount of trust, there is often a desire to limit particular data to certain groups of persons. For instance, marketing staff may view announcement letters and pricing information, but are not allowed to see the salary of a general manager. The persons within a particular group and the data sets exclusively assigned to this group form together a category. In order to define relations between subjects and objects in accordance with these levels and categories, one of RACF 1.7’s enhancements was the introduction of a security classification. It was based on:

**Security Level**: the administrator could select hierarchical levels at a scale from 1 to 254, and assign symbolic names to these levels. The higher the number, the higher the security level. For instance, INTERNALUSE is 10, CONFIDENTIAL is 100, SECRET is 150 and TOPSECRET is 200. This allows RACF to compare a user’s level with a data set’s level. E.g., a user authorized for INTERNALUSE is not allowed to access any data set classified as CONFIDENTIAL.

**Security Category**: the administrator may define non-hierarchical categories of users and resources, and assign symbolic names to them. This allows to specify relations between groups of users and groups of resources. E.g., the marketing department’s data sets marked with the category OFFICE may only be accessed by users with the category OFFICE specified in their user profile.

Both levels and categories are specified as members of the profiles SECLEVEL and CATEGORY in the RACF class SECDATA. These members can be maintained via the RESOURCE ALTER (RALTER) command by the RACF administrator. An example of such a structure is:

```
Class SECDATA
   - Profile SECLEVEL
     - Member INTERNALUSE /10
     - Member CONFIDENTIAL /100
     - Member SECRET /150
     - Member TOPSECRET /200
   - Profile CATEGORY
     - Member OFFICE
     - Member FACTORY
     - Member WAREHOUSE
```

After the RACF administrator has defined these levels and categories, the users may assign them to data sets by issuing, e.g.

ADDSD ‘PETER.DATA.SET’ SECLEVEL(SECRET) CATEGORY(OFFICE) etc.

ALTDSD ‘PETER.DATA.SET’ ADDCATEGORY(FACTORY) :

In this example, the ADD DATA SET DESCRIPTION (ADDSD) command is issued to create a profile for the data set PETER.DATA.SET and to assign the level SECRET and the category OFFICE to it. In a non-B1 situation, this means that the user who wants to access this data set must have at least the level SECRET and the category OFFICE in his RACF user profile. Via
the second command in this example, the ALTER DATA SET DESCRIPTION (ALTDSD) command, the number of categories is extended to two by adding FACTORY. Now the data set can only be accessed by a user at level SECRET with both the categories OFFICE and FACTORY in his user profile.

The reader will notice that security levels and categories are prerequisites for the security labels, introduced by RACF 1.9 in order to fulfill the B1 requirements (see section 6).

1.3 Erase On Scratch

A data set on a disk consists of a number of records, which can be found via a ‘pointer’, i.e. the Data Set Control Block (DSCB) in the disk’s Volume Table of Contents (VTOC). In this VTOC, one will find the administration of all free space on the disk, and the DSCBs of all existing data sets. Initially deleting a data set only meant removing this pointer - the DSCB - and adding the data set’s space to the disk’s free space. While removing the pointer hides the data, it does not actually remove this data. The free space of a disk still contains all data, and each user allocating free space to create a new data set has the opportunity to read the residual data still residing on his part of the disk. This habit of the system is well known to the developers of information systems, and they are aware that data sets containing sensitive data first must be overwritten with either binary zeros or random data before the space may be released when deleting the data set.

In order to assist the users, one of the enhancements introduced by RACF 1.7 was the Erase On Scratch mechanism. It can be influenced both by the RACF administrator and by the users, and is intended to overwrite specified data sets with binary zeros before the space is released, so preventing sensitive data from unintentional disclosure. For the user, the following parameters were introduced:

```
  ADDSD profile-name  | ERASE  | SECLEVEL(seclvel name) |
  ALTDSD profile-name | ERASE  | SECLEVEL(seclvel name) |
                   NOERASE | NOSECLEVEL         |
```

In this way, the user authorized to add or alter a profile could decide to activate the Erase On Scratch mechanism for a data set by setting the ERASE indicator in the data set profile. The effect of this indicator depends on the system-wide options selected by the RACF administrator, who may issue the SET RACF OPTIONS (SETROPTS) command with one of the parameters:

```
  SETROPTS  | ERASE( ALL ) | ERASE(SECLEVEL(seclvel name)) | ERASE(NOSECLEVEL) | NOERASE
```

With the option ALL, he indicates that each data set has to be overwritten by binary zeros at delete time, regardless of what the user specifies in the data set profile. By specifying a security level he forces RACF to overwrite all data sets at the specified level and higher; this implies that the data sets at lower levels are not overwritten. When specifying NOSECLEVEL, he allows the user to control the Erase On Scratch mechanism; only in this case RACF will use the ERASE indicator in the data set profile to decide whether the data must be overwritten during delete processing. With the NOERASE option the RACF administrator overrules the
ERASE indicator and does not allow RACF to overwrite any data set: all data of each deleted data set will then stay in the disk’s free space and can be read by everybody allocating this free space.

With the Erase On Scratch mechanism, a computing center has the opportunity to control in detail how the system deals with residual data. Obviously, the EDP-auditor has to pay due attention to how the user handles the ERASE indicators and how the RACF administrator has set the system-wide options to judge whether a computing center is effectively using this mechanism.

1.4 Notify Function

If a subject (user) attempts to access an object (resource, data set) without having a sufficient authority, RACF will reject access and terminates the subject’s current activity with an Abnormal End (ABEND) code 913. A message with the userid, name of the subject, name of the object, the allowed level of access (e.g., READ) and the attempted level of access (e.g., UPDATE) are both displayed to this subject committing the violation and on the system console. Since RACF 1.7, a similar message can also be routed to the owner of the data set or to another user by issuing commands with the NOTIFY parameter, such as:

ADDSD ‘PETER.DATA.SET’ NOTIFY(PETER) etc.

ALTDS ‘PETER.DATA.SET’ NOTIFY(GEORGE)

In this way the owner of the data set PETER.DATA.SET, or a person having ALTER access to this discrete profile, or a RACF administrator can instruct RACF to inform a user in case of an access violation. After issuing the first command the receiver will be PETER, which is changed via the second command into GEORGE. An example of such a message to GEORGE is:

ICH70004I USER(SYLVia) GROUP(ABC) NAME(SYLVia JOHNSON) CN(00)
ICH70004I ATTEMPTED ‘UPDATE’ ACCESS OF ENTITY   CN(00)
ICH70004I ‘PETER.DATA.SET’ IN CLASS ‘DATASET’    CN(00)
ICH70004I AT 15:33:35 ON OCTOBER 2, 1990     CN(00)

If the userid GEORGE is not logged on, the message is stored into the system data set SYS1.BRODCAST and is displayed at the next logon of GEORGE. With this RACF 1.7 enhancement the user has the option to be informed if somebody is making errors and whether there is an attack to his data sets.

2 Password Management in Historical Perspective

Also in the area of passwords history shows improvements of a similar magnitude as those introduced for the objects. While the password was first in clear text in the system library SYS1.UADS, with RACF it was moved to the RACF data base and was scrambled via a masking algorithm. Although this provided more security, the password still remained in clear text in the Terminal Status Block (TSB) in virtual storage legible to many users, and, moreover, the masking algorithm was easy to compromise. Hence the design of TSO was
improved to remove the clear text password from the TSB, and RACF was extended with Data Encryption Standard (DES) for passwords. This was RACF 1.6 in 1984, introducing a one-way encryption via DES and so storing encrypted passwords which could not be decrypted.

Moreover the SETROPTS command was extended to allow the system administrator to specify rules for password syntax and usage. He now may issue the SETROPTS command with the following parameters:

**SETROPTS PASSWORD(HISTORY(number-previous-passwords))**: a list of previous passwords is maintained and the user is not allowed to select one of them when specifying a new password. So rotation of passwords is inhibited. Usually the list is 24 passwords.

**SETROPTS PASSWORD(INTERVAL(number-days))**: after the specified number of days a password is marked ‘expired’ and has to be changed by the user during his next logon. This forces the users to select periodically new passwords, for instance every 60 days.

**SETROPTS PASSWORD(REVOKE(number-invalid-passwords))**: if a user forgets his password and attempts to guess it, RACF will revoke (freeze) his userid after he exceeds the specified threshold. This is a protection against hackers attempting to guess userrids and password. In many installations the threshold is between 3 and 5 attempts, which substantially reduces the likelihood of a hacker guessing a valid password.

**SETROPTS PASSWORD(RULEn(LENGTH(m)content(position))**: up to eight syntax rules can be specified for new passwords. For each position, one may indicate whether an alphabetic, alphanumeric, numeric, vowel, nonvowel and consonant character are allowed. In order to inhibit the users to select trivial passwords, one may, e.g., force them to insert at least one numeric character in the new password. This can be specified as:

```
SETROPTS PASSWORD( RULE1(LENGTH(6:8) NUMERIC(1)) +
RULE2(LENGTH(6:8) NUMERIC(2)) RULE3(LENGTH(8) NUMERIC(8)))
```

So if the password contains between six and eight characters of which at least one is a numeric, one of the eight rules will apply and the password is accepted. RACF now will accept GRANNY76, but will reject the password MICHAEL.

In some centers there is a tendency to implement a password checker. Its code is included in a RACF New Password Exit routine (ICHPWX01), and it may verify whether the password contains adjacent keys on the keyboard, an indication of month or year, repeating characters, a part of the userid, etc. It turns out that such password checkers are userunfriendly, because the user becomes confused when new passwords are rejected while he does not understand why, and even increase the vulnerability of passwords. If the user cannot specify passwords which are easy to remember to him, he will write the passwords in his agenda or on paper, so making it vulnerable for disclosure to unauthorized persons.

It is the author’s conviction that RACF at the level 1.7 and higher provides sufficient support to force the users to use passwords in a responsible and secure way, and that there is no need for additional checks of the new passwords in an exit routine. One must keep in mind that a trivial password such as a userid or the user’s Christian name can only be selected once, and thereafter has to be followed by at least twenty three other passwords. Moreover, forcing the user to insert at least one numeric in the password inhibits to use names of persons and brands of cars. And, after all, the hacker has only 3 to 5 changes to guess the correct combination of userid and password. Elementary statistics shows that the probability of a hit is negligible in
such an environment as long as there are reliable procedures for initial passwords and password resets.

Above we outlined how MVS migrated from a ‘closed shop’ with none or minimal security to a mature and secure environment for processing vital data. With modern RACF and a security-minded attitude by those in charge of the system, hackers have no realistic change to breach the security.

3 Logging in Historical Perspective

One of the major design objectives of MVS as an operating system supporting commercial batch jobs was to log all usage of resources for accounting and billing purposes. One defined System Management Facilities (SMF) to collect data from the operating system, its subsystems and the major supporting packages, and to store it in a formalized format into the SMF data sets. As soon as such a data set is full, SMF switches to another predefined data set and asks the operator to copy the content of the full data set to disk, tape or cartridge, and to empty this data set by resetting its pointers.

RACF also employed SMF for logging relevant events, violations and status information. With RACF 1.5 in 1983, both the owner of a resource and the RACF (group) administrators and auditors were allowed to specify the logging options by access level, such as READ, UPDATE, CONTROL and ALTER. They could instruct RACF to write logging records in case of a successful and/or failing access to a data set at or above the specified access level. With RACF 1.6 in 1984, it became possible for the (group) auditor to inspect all RACF profiles within the scope of his authority, and to set the associated logging options. Many of these options only can be set and inspected by auditors, and are hidden from other users and administrators, so that nobody but the auditor can see what is logged.

With RACF 1.7 in 1986 the logging was extended to all activities by the users with the (group) OPERATIONS attribute, and with RACF 1.8 in 1988 it became possible to log activities by security level. Currently the auditor can use the following RACF commands and facilities:

**ALTER and LIST DATA SET DESCRIPTION (ALTDSD and LISTDSD):** with the GLOBALAUDIT parameter the auditor may instruct RACF to log all or selected access events per data set or per group of data sets. A similar mechanism is available to the owner, to a person with ALTER authority for a discrete profile, and to a RACF administrator: they can set the AUDIT parameter. The net effect is an OR-relation of the AUDIT and GLOBALAUDIT setting: if at least one of them asks RACF to log an event it will be logged as an SMF record. The reason to make a distinction between AUDIT and GLOBALAUDIT is to allow the auditor to set the logging options as desired by him, without the owner knowing what RACF will log on request of this auditor.

**RESOURCE ALTER and LIST (RALTER and RLIST):** provide a similar facility as the ALTDSD and LISTDSD commands, but now for the non-data set resources.

**LIST GROUP (LISTGRP):** all members of a group are listed, together with their authorities within this group. The listing shows which users can act as group administrator, can add
other users to this group or can define sub-groups, have a create authority or maintenance responsibility for group resources, etc.

**ALTER and LIST USER (ALTUSER and LISTUSER):** setting the UAUDIT parameter means that every invocation of RACF on behalf of this user will be logged (unless access to a resource is approved by RACF’s Global Access Checking).

**SET RACF OPTIONS (SETROPTS):** the auditor may activate logging mechanisms by specifying the parameters AUDIT for a class name, CMDVIOL for command violations, OPERAUDIT, SAUDIT for all activities of the RACF administrator, and SECLEVELAUDIT.

**RACF REPORT WRITER (RACFRW):** select log records from SMF data bases, format the data and create reports about the logged events. Usually reports are periodically printed about logon violations revealing for which userids invalid passwords have been issued or invalid groups have been specified, and about resource violations showing which accesses have been rejected by RACF.

**DATA SECURITY MONITOR (DSMON):** create a report about the current status of the system’s security mechanisms, consisting of:

- System report: identification of system, level and status of RACF,
- Group Tree report: names of superior and sub groups, and userid of group owner (note: for a list of the group members one has to use the LISTGRP command),
- Program Properties Table (PPT) report: names of programs allowed to obtain Supervisor Mode and/or a key-in-storage reserved for operating system functions,
- RACF Authorized Caller Table report: names of programs allowed to call RACF to verify a user’s identity,
- RACF Class Descriptor Table report: names and status of all RACF classes,
- RACF Global Access Checking (GAC) Table report: global access rules applying to all subjects and all objects (see also Figure 12 on page 15),
- RACF Started Procedures Table report: names and authorities of programs which can be started by the system operator as so-called Started Tasks,
- Selected User Attribute report: all userids with the RACF system or group authority SPECIAL, OPERATIONS and AUDITOR, or being revoked,
- Selected Data Sets report: all data sets vital to a secure and reliable functioning of the system and their current security status.

**CROSS REFERENCE LISTER (ICHUT100):** provide a cross reference listing of the occurrences of certain strings in the RACF data base. For instance, show all authorities of the user PAANS and list all resource profiles in which this name is mentioned.

For both the security officer and the technical EDP-auditor, this set of facilities provides a solid foundation for their reviews and allows them to detect, track and report all relevant events. Authors such as Murphey, Rao Vallabhaneni and Weber base their audit strategies for MVS systems primarily upon these logging and auditing facilities.
USA Department of Defense (DoD) has published the book ‘Trusted Computer System Evaluation Criteria’, also called the Orange Book, in 1983. These criteria provide a basis to evaluate the effectiveness of security controls integrated into the Trusted Computer Base (TCB) of electronic data processing systems, and supported three objectives of DoD:

1. to provide users with a yardstick to assess the degree of trust that can be placed in computer systems for the secure processing of classified or other sensitive information,
2. to provide guidance to manufacturers as to what to build into their new, widely-available trusted commercial products in order to satisfy trust requirements for sensitive applications,
3. to provide a basis to specifying security requirements in acquisition specifications.

Using these criteria, one has grouped computer systems into four major divisions, i.e. D with minimal protection, C with discretionary protection, B with mandatory protection and A with verified protection. The two protections relevant to our current discussion are:

**Discretionary Protection:** the term ‘discretion’ means ‘liberty of deciding as one thinks fit’, which means in RACF terms that the owner, the person with the ALTER authority and the RACF administrator have the liberty to specify who may access the resources under their control. For instance, a database owner may limit access to the database to a single user only allowing him to read, or may allow all users to update or even delete this database. In this environment the amount of protection of vital data depends on how a relatively large number of owners and administrators define the profiles and access lists for resources.

**Mandatory Protection:** the term ‘mandatory’ refers to receiving a command from a superior authority. In the context of computer security it means that the owner of a mainframe defines how all data in his system is classified and protected, and nobody is allowed to deviate from these general rules. Even the owner of a database lacks the freedom to allow another user to access his database, unless this access is approved by the rules applicable to the entire mainframe. So with mandatory protection we have two layers of access control: first RACF will deal with the general - mandatory - rules and, second, it handles the rules per resource as specified by the resource owner.

Currently MVS with RACF up to 1.8 has been classified as a C2 system with controlled access protection. It is IBM’s intention to let MVS 3.1.3 with RACF 1.9 be evaluated as a B1 system with labeled security protection. Let us first view the Orange Book and summarize the requirements for such a move from C2 to B1:

**B1 requires all features of C2:** the following list will only mention the additional requirements relative to C2.

**Labels.** The TCB shall maintain security labels associated with each subject and object, and shall use these labels as the basis for mandatory access control decisions. At import, an authorized user has to specify the security level, and at export the TCB must use a communication channel or storage medium cleared for this level. Such a channel or device may be cleared for a single level, e.g., a dial-in line or a diskette may only be used for
INTERNALUSE data. One also may introduce multi-level channels or devices, but then the protocol used shall provide for unambiguous labeling of the data. If printed, the TCB shall mark the top and bottom of each page with human-readable labels.

**Mandatory Access Control.** A security label is a combination of a hierarchical level and one or more non-hierarchical categories. A user can read data if his label dominates that of the data, i.e. the subject’s security level is *higher or equal* to the object’s level, and the subject’s categories contain all the object’s categories. In case of write access, the subject’s level must be *less or equal* to the object’s level and the subject’s categories must be contained in the object’s categories. This means: "read down and write up", so preventing users from copying data to lower levels, but allowing them to provide input to higher levels. Obviously, update access is only allowed if the subject’s level is *equal* to the data’s level. These access rules can be described by the Bell-LaPadula Model (first published in 1973) introducing two properties:

*Simple Security Property:* a subject has read access to an object only if the security level of the subject dominates the security level of the object,

*-*property (Star Property): a subject has write access to an object only if the security level of the subject is *dominated by* the security level of the object.

The purpose of the *-*property is allowing subjects to move data to higher security levels, preventing, at the same time, any subject from declassifying data, i.e. writing data down to lower security levels. This implies ‘one way traffic’ for data, it can go up, but it never can go downwards.

**Accountability:** The process of identification and authentication of users shall be extended by determining the users’ clearance. The logging also has to be extended: the log records must describe the user’s security level and the auditor must have the capability to specify logging by security level.

**Life-Cycle Assurance:** An independent review team shall investigate and test the TCB to uncover all design and implementation flaws that would permit a subject to read, change or delete data, while such an action would normally be rejected by the TCB.

The major differences between C2 and B1 are:

- the security labels, which are combinations of RACF’s SECLEVEL and CATEGORY,
- the rule not to down-write data to lower security levels,
- printing the security level on all output pages,
- recording the security level in the log records,
- only verified users are allowed to use the system (in ‘old’ systems it was allowed to define a part of the user base to RACF and to have non-RACF users; with B1 all users must be RACF defined and even console operators must logon).

The reader should not underestimate the impact of adopting the *-*property. This does not only affect copying data sets, but also deals with sending files via Network Job Entry (NJE) and exchanging messages via the TSO SEND and LIST BROADCAST (LISTBC) commands. All these mechanisms have to be adapted to obey to the "read down and write up" rule.
In addition to the security levels and categories, which are part of the RACF class SECDATA (see section 2), one has introduced the class SECLABEL allowing the RACF system administrator to relate symbolic names of security labels to levels and categories. In general, one may say that the security label of subject X dominates the security label of object Y if, and only if:

- LEVEL OF X is greater or equal to LEVEL OF Y, and
- CATEGORIES OF X include all CATEGORIES OF Y

Note that X may have more categories than Y, but X should have at least those of Y.

Let us quote but one example to explain these relations: assume that the structure of Figure 11 has been defined via the RESOURCE DEFINE (RDEFINE) commands. It is based on four levels and three categories, i.e. OFFICE, WAREHOUSE and FACTORY. If there is a top manager, allowed to see all office data, but not directly involved in the transfer of goods in the warehouse and factory, one may define the label DIRECTOR containing level TOPSECRET and the category OFFICE. In theory, this label should be sufficient to see all data sets in the category OFFICE. However, since we also defined the labels PROD-MGR for a production manager and GROUP-MGR for a group manager, one will notice that these two labels contain categories beyond OFFICE, i.e. WAREHOUSE and FACTORY. So the label DIRECTOR does not dominate the labels PROD-MGR and GROUP-MGR, and he is hence not allowed to see the contents of data sets labeled with PROD-MGR and GROUP-MGR. The only label dominated by DIRECTOR is CLERK, both located within the category OFFICE. Obviously, if this top manager would be allowed to see everything, one has to extend the security label DIRECTOR by including the categories WAREHOUSE and FACTORY.

This example shows that the first step towards a B1 system is to make an exhaustive inventory of all subjects, objects and their relations. This inventory has to be translated into a set of levels and categories supporting a consistent set of security labels. Thereafter labels can be assigned to subjects by, it may surprise a novice reader, permitting the user READ access to the label. To define the labels DIRECTOR and PROD-MGR of Figure 11 and to assign them to users, one has to issue the following commands:
SETROPTS SECLABELCONTROL

RDEFINE SECLABEL DIRECTOR SECLEVEL(TOPSECRET) ADDCATEGORY(OFFICE)
RDEFINE SECLABEL PROD-MGR SECLEVEL(SECRET) ADDCATEGORY(OFFICE FACTORY)

PERMIT DIRECTOR CLASS(SECLABEL) ACCESS(READ) ID(GEORGE,JOHN,Sylvia)
PERMIT PROD-MGR CLASS(SECLABEL) ACCESS(READ) ID(GEORGE,Peter)

Using the PERMIT operation with READ, one allows the users to use the label. When establishing a session with TSO, the new TSO LOGON panel contains one additional field, named SECLABEL, in addition to the conventional fields such as USERID, PASSWORD, NEW PASSWORD, etc. Here the user may specify one of the SECLABELs to which he has been authorized. In this example, user GEORGE may select either the label DIRECTOR or PROD-MGR, while PETER may only use PROD-MGR. Of course, GEORGE may logoff after a session with DIRECTOR and re-logon with PROD-MGR, but his scope then is limited to what is allowed by the label PROD-MGR. At any time, only one security label applies to a TSO session or to a batch job. For the latter, one may specify the SECLABEL in the JOB card, e.g.

//GEORGE1 JOB (123,C321),’George York’,USER=GEORGE,SECLABEL=DIRECTOR

After having specified the SECLABEL during logon or in the JOB card, this label is used to test the TSO user’s or batch job’s authority before any access to a resource is allowed (unless overruled by a global access rule).

With the introduction of security labels, the authorization checking process has not been changed. If a user requests access to a data set by issuing the OPEN SuperVisor Call, RACF is invoked and will perform a series of tests as shown in Figure 11. The first test is done by Global Access Checking (GAC): if the global access rules as specified in the GAC table allow access there will be no log record, regardless of the activated logging options. If access is not approved by GAC, RACF will search for the most specific generic or discrete RACF profile for this resource in its buffers in virtual storage or will fetch it from the RACF data base. Thereafter the security label of the resource is compared with that of the requesting user. If the user’s security level is too low or he misses one or more categories, the access request is immediately denied. This test can never approve access: if the level and categories are correct RACF will continue with the standard test sequence on userid, standard access list, universal access, OPERATIONS attribute, and the conditional access list for using specified programs to access the resource, until there is an approval or a denial. If there is no matching condition, access is obviously denied.
7. MULTI LEVEL SECURITY

Characteristic of a Multi Level Security (MLS) system is the existence of resources at distinct levels of security, each labeled in accordance with its classification. The so-called *-property, indicating that write-down is not allowed, may be seen as an extension of the label concept. In RACF 1.9 one has introduced four mechanisms to control such an MLS environment, i.e.

1. allow the use of security labels (SECLABELCONTROL),
2. enforce the use of these labels for all subjects and objects (MLACTIVE),
3. enforce the *-property (MLS),
4. allow controlled maintenance (MLSTABLE and MLQUIET).

The first mechanisms is the class SECLABEL. One has to activate this class by issuing the command SETROPTS SECLABELCONTROL and by establishing links between security labels, levels and categories. This has already been discussed in section 6.

The second mechanism enforces that all subjects must use these security labels to establish TSO sessions, to start batch jobs and to access objects. This prerequisite for MLS is controlled by the ML parameter, and the desired option can be selected by issuing one of the commands:
By temporarily using the WARNING mode a computing center allows the administrators to implement security labels gradually; after full implementation one may switch to FAILURE mode which means that working without security labels is inhibited.

Now we have defined a set of security labels and have enforced everybody to use these labels, the switch to B1 security has been prepared. The third and last step to B1 is enforcing the *-property (no write-down). This mechanism is controlled by issuing one of the commands:

```
SETROPTS NOMLS
SETROPTS MLS(WARNING)
SETROPTS MLS(FAILURES)
```

After a full implementation of the security labels and activation of the MLS(FAILURES) option, the following rules apply:

**READ ONLY**: If user X’s label dominates the label of resource Y, user X may read the content of Y.

**WRITE ONLY**: Resource Y’s label must dominate the label of user X, and the resource profile must allow user X to write to Y, and the RACHECK function must be invoked with the STATUS=WRITEONLY option. Only if these three requirements have been fulfilled, user X may write to the resource Y (but may not read the content of Y).

**UPDATE**: Only if user X’s label is equal to the label of resource Y, the user may read and write.

Only after activating both SECLABEL, MLACTIVE and MLS, all users are enforced to use security labels and the system adheres to the *-property by inhibiting declassification of data to lower security levels. Now we have a system with B1 security.

A fourth mechanism has been introduced to maintain the set of security labels. It is obvious that an administrator is not allowed to change labels in a running system, because this impacts the overall security and causes an undefined (and so an insecure) situation. In order to let administrators change the security labels in a controlled way without disrupting active sessions, one may issue the SETROPTS command with the parameters MLSTABLE and MLQUIET to enforce tranquillity. In a tranquil system, the regular users are not allowed to logon nor to start batch jobs. Only the administrators, i.e. the users with the RACF SPECIAL attribute, are allowed to logon in order to adapt the security definitions. Moreover, the tranquil state allows operators to start MVS started tasks which also can be used for administrative purposes.
TSO users may exchange notes by issuing the TSO SEND command, which has the following syntax:

```
SEND 'text' USERID (userid list) NOW LOGON
OPERATOR (route code) SAVE
CN (console id)
```

The NOW parameter allows the user to send the text immediately to another user without storing it, which implies that if the recipient is not logged on the message is deleted. When specifying the LOGON parameter, the message is displayed if the recipient is logged on, and is otherwise stored in the system data set SYS1.BRODCAST until the recipient reads his mail. In case of SAVE, the message is only stored in SYS1.BRODCAST. As an alternative, the user may send the message to an operator by specifying an MVS route code, or to a console by specifying its identification.

If the message is stored in SYS1.BRODCAST, the recipient may view it by issuing the LIST BROADCAST (LISTBC) command. The system then inspects the data set and displays all messages directed to this user.

It is obvious that the conventional mechanism of SEND and LISTBC might be (ab)used in a Multi Level Security environment to copy data to lower levels, and so to bypass the "read down and write up" rule. To ensure that users cannot declassify data via messages, MVS 3.1.3’s TSO has been extended with mandatory access control to verify that the recipient is authorized to view the messages. This extension deals with two questions:

1. Who may sent a message to whom?
2. If there is a message, may the recipient receive it?

The first question requires that both the sending and receiving userids are made known to RACF as legitimate participants in the communication process. This is done by defining the receiving userid as a RACF resource in the new RACF class SMESSAGE. E.g., the administrator may issue the following commands:

```
SETROPTS CLASSACT(SMESSAGE)
RDEFINE SMESSAGE GEORGE UACC(READ)
RDEFINE SMESSAGE JOHN UACC(NONE)
PERMIT JOHN CLASS(SMESSAGE) ID(SYLVI A PETER) ACCESS(READ)
```

For this type of resources, assigning the READ authority to a userid means, in fact, authorizing this userid to send messages. In this example, the universal access READ for the resource GEORGE in the class SMESSAGE implies that any and all users may send messages to the userid GEORGE. The second definition applies to the userid JOHN, which got a universal access NONE so that nobody can send messages to it, unless explicitly authorized via the
PERMIT command. We did so for the userids SYLVIA and PETER by permitting them READ access to this resource.

At the receiving side, TSO also has been extended to check whether the security label of the recipient dominates the security label of the sending user. If the Multi Level Security option has been activated via the SETROPTS MLS(FAILURES) command, the message will be suppressed if the recipient’s label does not dominate, simply because the recipient then lacks sufficient authority to view this message.

With these new mechanisms introduced by RACF 1.9 the administrator may control who may send messages and to whom, and the system will prevent data from being declassified via SEND and LISTBC communication.

7 Conclusions

The technical EDP-auditor is principally interested in how the company has defined its security strategy, how it has formulated the security objectives for each of its information systems, and whether the implemented measures and procedures for security and internal control are effective to support the security objectives. In this scope the technical EDP-auditor is interested in the new mechanisms provided by MVS 3.1.3 and RACF 1.9, which can be used by the computing centers as tools to protect the data and the information systems. It is his responsibility to judge whether the computing center uses the right tools for the right purposes, and whether there is a balance between the value and confidentiality of the data, and the measures implemented.

If a center implements a system with B1 security, the technical EDP-auditor should be capable to judge whether the proper RACF definitions are in place and whether the center has assigned the right authorities to the right persons. In order to form an independent and correct opinion about a particular installation, he must understood what B1 means, which tools are provided by the TCB, and how these tools must be used. Although limited to only some aspects of B1 security as provided by MVS/ESA and RACF, this paper attempts to show that the new mechanisms are a logical step in the evolution process of the IBM enterprise systems, the so-called mainframes. The introduction of B1 security does not imply a shock effect, but, nevertheless, it may require additional education for both the security staff in the computing centers and the technical EDP-auditor to obtain a maximum benefit of the new mechanisms.

8 Note

MVS 3.1.3 and RACF 1.9 provide a large number of enhancements allowing a center to control the flow of batch jobs and the activities of users in much more detail than in the previous versions of this TCB. However, due to the limited space in this paper we decided to limit our discussion to security labels and to present these labels in the light of the evolution process of MVS security. The reader is referred to the IBM manuals on RACF 1.9 and B1 Security for information about the other enhancements.