1

How the Mainframe Provides Security

Any farmer will tell you, only a fool lets a fox guard the henhouse door.
—Proverb

One way to implement mainframe security is to let all applications running on the system manage their own security. However, that would be akin to allowing foxes to guard the henhouse. Instead,the mainframe operating system is entrusted with providing security for all users and applications sharing the computer. Being an independent entity, the operating system has no vested interest in compromising the data.

A key integrity feature of the z/OS mainframe operating system is that all programs doing work are kept apart from each other. In other words, one program cannot see what the other is doing. This segregation is implemented via a feature called *address spaces*, whereby each entity in the mainframe is allocated an address space and cannot look into other address spaces.

Thus, the very foundation of the operating system provides data integrity, as shown in Figure 1.1.

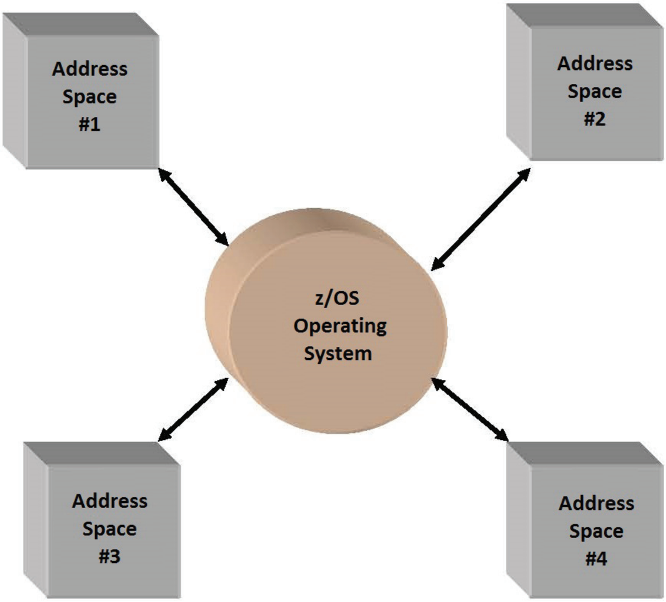

Figure 1.1: Programs are kept within their own boundaries by the operating system in the middle.

While the operating system provides basic integrity, it uses an "external" security product to do all other security checking.

When we talk about an external security product, we mean it is external to the "core" operating system. However, the security product is still part of the operating system. The security checking has been externalized from the core operating system to enable competing security products to provide mainframe security.

There are three main mainframe security products: IBM's Resource Access Control Facility, or RACF, and ACF2 and Top Secret, both from CA Technologies (formerly Computer Associates International). We will use RACF throughout this book.

The operating system intercepts all authentication and validation requests. It then passes along these requests to RACF, which in turn makes its decision based on information in its security database. In this sense, the operating system is strictly a gatekeeper or go-between; it does not actively make decisions to allow or fail the security requests.

One can think of the operating system as having subcontracted all installation-specific security checking to RACF.

How RACF Does Access Checking

When RACF receives a request for access checking, it decides to grant or deny the request based on information residing in the RACF database. RACF checking for an access request is quite involved. There are of course the "access lists" in RACF profiles that specify who has access, but that's not all. Several other factors influence RACF's decision-making process. In addition to access lists, following are the main factors RACF considers before deciding whether to grant or deny access:

1. *Universal access*—The "universal access" (UACC) specified in the profile is above and beyond what is in the access lists. For example, if the value is READ and a user ID is not in the access list, then the user ID gets at least READ access.

2. *General access*—If the profile has an entry of * (an asterisk) in its access list, all user IDs have access that is specified for *. There is a subtle difference between this general access and universal access. This is covered in chapter 11, "Security Administration: Beyond the Basics."

3. *Operations privilege*—If a user has the OPERATIONS privilege, the user might get access because of that fact. This is discussed in detail in chapter 2, "RACF Special Privileges."

4. *Global Access Checking (GAC) table*—The GAC table can grant access before the pertaining profile is even checked. This is discussed in detail in chapter 5, "The Global Access Checking (GAC) Table."

5. *RACF exits*—RACF exits can override all access definitions in the RACF database. This is discussed in detail in chapter 16, "RACF Exits."

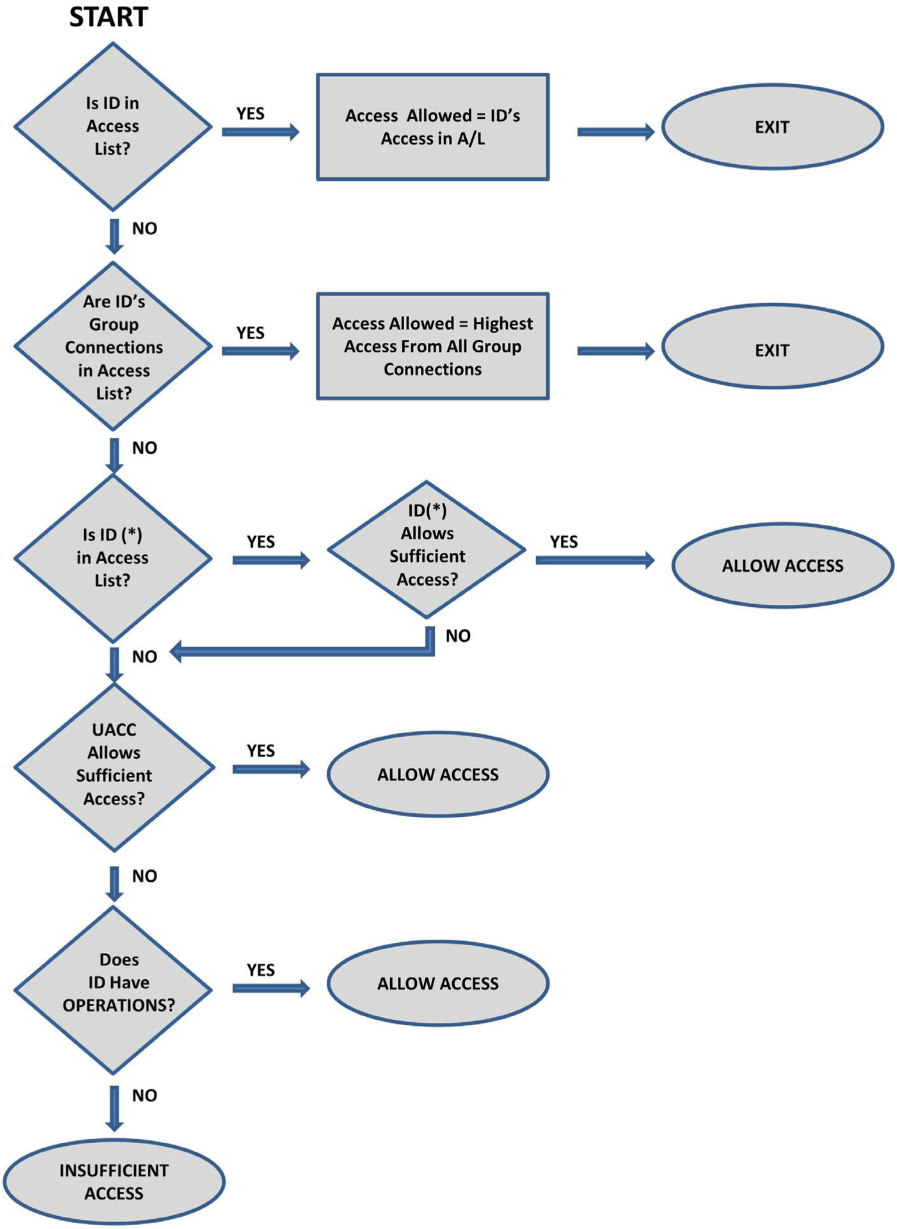
START
Is ID in Access List?
YES
Access Allowed = ID's Access in A/L
EXIT
NO
Are ID's Group Connections in Access List?
YES
Access Allowed = Highest Access From All Group Connections
EXIT
NO
Is ID (*) in Access List?
YES
ID(*) Allows Sufficient Access?
YES
ALLOW ACCESS
NO
NO
UACC Allows Sufficient Access?
YES
ALLOW ACCESS
NO
Does ID Have OPERATIONS?
YES
ALLOW ACCESS
NO
INSUFFICIENT ACCESS

Figure 1.2: A simplified version of RACF access checking.

The RACF Access Checking Diagram

The diagram in Figure 1.2 is a simplified version of RACF authorization checking. It covers the main areas, but it does not go into the details of seldom-used cases. Let's use the diagram to understand how RACF works, by taking Quiz 1.1.

Quiz 1.1

1. The user ID JOHN is connected to two groups, GROUP01 and GROUP02. GROUP01 has READ access to the profile PROD.DATA.**, and GROUP02 has UPDATE access to the same profile. What access does JOHN have to this profile?

   Answer: JOHN will have UPDATE access, since RACF looks at accesses of all groups the user ID is connected to and grants the highest among them.

2. User ID CHAN10 is explicitly specified with READ access in the access list of the profile PROD.DATA.**, but the user ID is also connected to the group ACCT1, and ACCT1 has UPDATE access to this profile. What access does CHAN10 get, READ or UPDATE?

   Answer: In this case, the user ID will get READ access. If the user ID is explicitly mentioned in an access list, then the access specified for that user ID is always what the user ID gets, regardless of other group connections. This RACF feature comes in handy when the group is large and a few individuals in the group require less (or more) access than the others.

3. If the universal access (UACC) of a profile is UPDATE and the user ID SMITH10 is explicitly specified as having READ access, what access does the user ID get?

   Answer: SMITH10 only gets READ access. If, however, the UACC value is READ and the access list specifies UPDATE for SMITH10, then SMITH10 gets UPDATE access. In other words, the specific access overrides the universal access. This feature of RACF comes in handy when there are a few exceptions from the general access required by all users.

4. The user ID PETER6 has the powerful OPERATIONS privilege that allows full access to all of the installation's data. How can you prevent PETER6 from accessing the payroll master file?

   Answer: Based on the flowchart in Figure 1.2, if a user ID is specifically mentioned in the access list, then the access specified in the access list is all the user ID gets. This is one instance where the special privilege OPERATIONS is overridden. This allows you to reduce the powers of the OPERATIONS privilege. So simply specify that user ID PETER6 is not to have any access in the access list of the profile for the payroll master file:

   ```
   PERMIT 'PROFILE' ID(PETER6) ACCESS(NONE) GENERIC
   ```