Global Information Assurance Certification Paper

Enhance mainframe system security while standardizing it

A step by step approach

Yves Depoorter

GSEC Practical Assignment

Version 1.4b – option 2

April 9, 2004

Abstract

The purpose of this paper is to propose a step by step approach to standardize and enhance system security in an OS/390 environment running RACF as security software[1]. What follows is essentially common sense added to the long experience accumulated by the security members of the cross-country team I've been (and still am) working for. I personally had the opportunity to use this approach several times and to contribute to it to some extent. I have always been convinced of its added value.

As mentioned in the SANS Security Essentials Course[2], the principle of defense in-depth requires a system to implement a multi-layer protection. The scope of this paper addresses the "host" layer, i.e. the system security[3].

I will first explain the context triggering a system security standardization project. Then, detail each of the 10 steps, specifying what has to be done, why it needs to be done, and how it can be done[4]. I will also discuss the potential impact on the running environment. In the summary, I will confirm the fact that the completion of these steps will enhance the system security, increase the level of control and optimize the security administration.

[1] RACF (Resource Access Control Facility) is the IBM security software running on mainframe systems. The approach is also valid for other security software, but technical aspects will vary.
[2] SANS Institute, SANS Security Essentials with CISSP CBK Version 2.1, Volume 1, pages 292-294.
[3] the four defense in-dept layers are: the Network, the Host, the Application and the Information layers.
[4] if you don't have RACF skills, it will not compromise your understanding of the general idea developed in this paper.

TABLE OF CONTENT

Introduction

Many companies have had to deal (or will sooner or later) with a consolidation of IT systems support teams due to reorganization, company merge, or outsourcing of IT services.

In this situation, the short term objective for the new system support team s often consists in taking over the system as is and transferring skills in order to keep service disruptions to a minimum. However, very quickly, optimization (i.e. standardize and enhance) of security as for any other IT management discipline, should become a priority.

In the security domain, the most frequent reasons to standardize are:

- managing a number of heterogeneous systems is more demanding in terms of resources
- corporate security policies and standards are not respected in all environments
- system security has been neglected for years because the efforts were mainly focused on application security. Following the principle of the weakest link in the chain, the overall environment is potentially exposed
- based on the same principle, the same risk applies where systems with different security levels communicate with each other
- the principle of separation of duties cannot be guaranteed. Therefore, a clear split between system resources and application resources is needed

My own experience is based on standardization of mainframe systems in an outsourcing environment. However, the approach proposed in this paper can be applied to enhance the security of any mainframe system.

In the next chapter, I will give an overview of the 10 step approach and discuss each of them in broader detail.

The 10 step approach

If you want to efficiently standardize the system security of your environment, you will have to:

- collect security related data
- store the data in a central database or tool
- analyze the data in order to evaluate what needs to be changed
- implement the changes
- check if you correctly implemented the changes

To help you achieve this, I propose to apply a security template consisting of the 10 following steps:

- Step 1: Collect and store security data
- Step 2: Implement basic auditing
- Step 3: Implement userid ownership
- Step 4: Adapt RACF group architecture and define system job roles
- Step 5: Review system-wide security settings
- Step 6: Standardize Operating System Resources [5]
- Step 7: Activate new Operating System Resources protection
- Step 8: Perform further security actions
- Step 9: Validate the new system security status
- Step 10: Validate the new environment

Consider the template as a "steps to follow" checklist. Depending on specific situations, the steps may be re-sequenced, merged or skipped.

By following the template, you will:

- ensure you won't forget to consider all aspects
- be able to implement a common but flexible structure
- apply an identical solution to identical cases

Each step contains [6]:

- a "What" section describing actions to be taken
- a "Why" section describing reasons for taking them
- a "How" section describing practical and technical tasks to be performed in a RACF environment. This assumes the reader has basic knowledge of the RACF product
- a "potential impact" section describing the "risk" in terms of system availability

[5] Operating System Resources = resources of the operating system, e.g. operating system files, programs, commands, etc in opposition with user and application resources. Later in this paper, I will also use the acronym "OSR".

[6] I didn't cover the workload aspect in this paper. Consider that the system size, the architecture in place, the number and the homogeneity of software are key elements for realistic workload estimation.

Step 1: Collect and store security data

What

In this step, you will:

- collect security related data on the system
- store the data in a central database or tool

Why

This input is a prerequisite to run the standardization project. The database or tool will allow you to query the current situation and/or generate commands to be executed on the system.

How

Proceed as follows:

- prepare a JCL job (or use one provided by a security tool or software you may have installed on the system) which will extract, in a well defined and structured format [7], all possible security related data from your system :
  - the RACF database (using IRRDBU00 utility)
  - the Data Security Monitor file
  - the APF and LINKLIST libraries
  - the SYS1.PARMLIB
  - the system procedures libraries
  - the catalogs
  - the output of MVS display commands
  - subsystems information : TSO, VTAM, CICS, DB2, HSM, etc
  - etc
- define the extraction job in your job scheduler software to ensure its regular execution (e.g. daily run)
- if you already manage other systems which you previously standardized, you will most probably have your central database running on another system (the focal point). In that case, you will also have to transfer the data to the focal point after each extraction.
- upload the data in your central database, which can be a security tool (e.g. Consul RACF or Vanguard) or "simply" tables in a classic relational database (e.g. DB2 tables).

Impact of the running environment

This step has no impact on your environment as you only collect and store information. Nevertheless, ensure that the collected data has the right level of protection.

[7] the format must match the layout of the tables to be uploaded .

Step 2: Implement basic auditing

What

In this step, you will activate the system-wide audit parameters. The audit parameters at the resource level will be addressed in a subsequent step.

Why

You must ensure that security related events are logged and kept during a minimum period of time [8]. This allows you to generate security reports and perform security investigations, either on a regular basis or to satisfy ad-hoc requests.

In addition, making audit records available early in the standardization process will help you take the right decisions in the following steps.

How

Proceed as follows:

- activate the RACF system-wide audit settings, i.e. SAUDIT, OPERAUDIT and CMDVIOL
- activate the audit settings for the key RACF classes. These key classes are environment dependant but in general consider the USER, GROUP and DATASET classes, and all Operation System Resources (OSR) classes.
- activate the LOGOPTIONS ALWAYS parameter for classes requiring unconditional auditing, e.g. the SURROGAT and DASDVOL classes.
- put in place the collection and the retention of the generated SMF records.

Impact on the running environment

Nowadays, disk storage space itself is less critical than in the past, but make sure that the additional SMF space required by the logging has been accounted for.

[8] defined in the corporate security policy.

Step 3: Implement userid ownership

What

In this step, you will assign an ownership to all userids.

There are various criteria you can use and combine to define ownership. For instance, you could build ownership information based on the sequence:

- the ownership keyword
- the employee last name
- the employee first name
- the employee number

Whatever the approach chosen, the idea is to uniquely identify the individual owning the userid. That is the reason why using the employee number provided by the Human Resources department is often recommended.

Example of ownership information following the above suggested syntax:

```
**OWN**Hood/Robin/01234/
**OWN**Bond/James/00007/
**OWN**Max/Mad/02096/
**OWN**Simpson/Bart/01988/
**OWN**HelpDesk//hpdsk/
**OWN**Started Task//systm/
```

Now you will need the right coding to "decrypt" the ownership information: find the ownership keyword "**OWN**", then use the "/" as separator to find last name, first name and employee number. For non personal userids, pseudo employee number can be used[9] but must refer to a real owner in a separate table.

Why

You must ensure an effective userids management, which begins with userid ownership.

How

Proceed as follows:

- make an inventory of all userids on the system
- check if the ownership information syntax is correct and fix deviations
- check the ownership information relevance against reference data, e.g. the Human Resources database
- delete non required userids

In a RACF environment, it is recommended to store ownership information in the INSTALLATION-DATA field. Even when already used for other purposes[10], it is long

[9] "*hpdsk*" and "*systm*" in the example.
[10] e.g. for application access purposes

enough to store ownership information further in the field and use the ownership keyword to detect the ownership information[11].

In the example hereafter, application related information [12] can coexist with ownership information :

```
**OWN**Hood/Robin/01234/
**OWN**Bond/James/00007/
APPL A-APPL B**OWN**Max/Mad/02096/
APPL B**OWN**Simpson/Bart/01988/
**OWN**HelpDesk//hpdsk/
**OWN**Started Task//systm/
```

Impact on the running environment

This step has no impact on your running environment. Nevertheless, you will have to review the user registration process and tool to ensure that ownership data be taken into account.

[11] in addition, a user cannot modify his/her own INSTALLATION-DATA field unless specifically authorized.
[12] "*APPL A-APPL B*" and "*APPL B*"

Step 4: Adapt RACF group architecture and define system job roles

What

In this step, you will define three main "branches": the system branch, the application branch and the end-users branch [13]. Groups linked to the system job roles are also created.

Why

You must ensure that the management of system resources, application resources and end-users resources is clearly separated. The usage of groups will clarify the resources protection and simplify the access management.

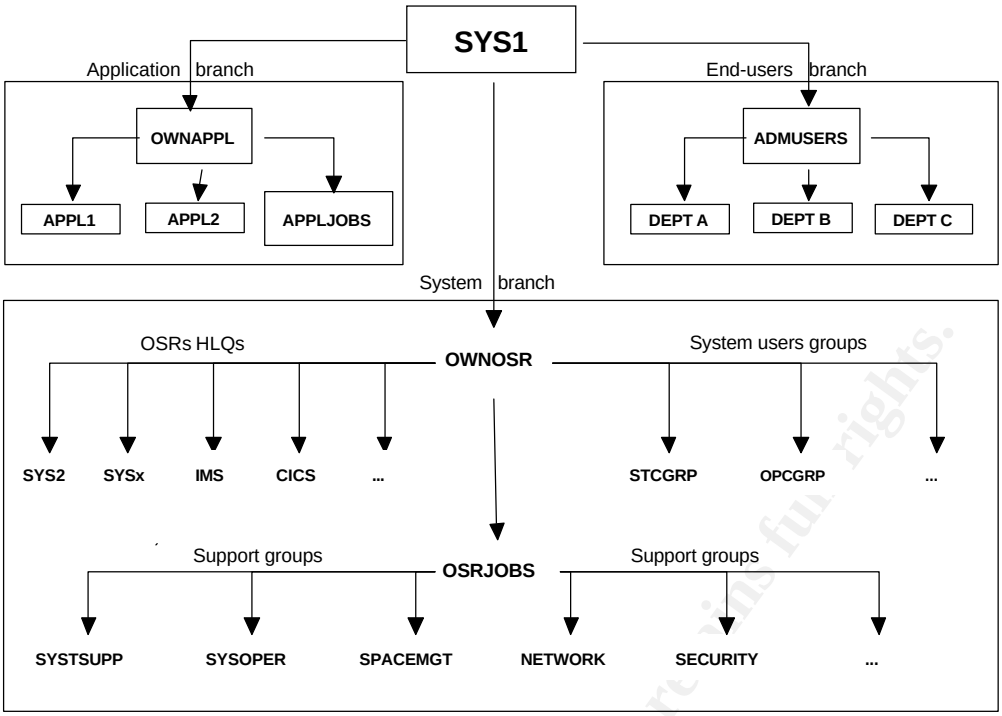
How

Proceed as follows [14]:

- define the OWNOSR group under SYS1. It is the group owning Operating System Resources (OSR), i.e. the system branch.
- assign OWNOSR as owning and superior group of all RACF groups covering OSR dataset high level qualifiers (HLQs): SYS2, IMS, CICS, etc.
- assign OWNOSR as owning and superior group of other OSR related groups: started task users group, job scheduler users group, etc.
- assign OWNOSR as owning and superior group of the OSRJOBS group. This group will be the superior group of all functional groups related to system support activities: system support, system operations, space management, network support, security, etc.
- connect the support people to their respective functional group(s).
- define the OWNAPPL group under SYS1. It is the group owning the application data structure, i.e. the application branch.
- define the ADMUSERS group under SYS1. ADMUSERS is the group at the top of the user's group structure, i.e. the end-users branch.
- depending on your decentralization level, define a more or less granular group structure.

The following figure gives an idea of what a basic structure should look like.

[13] the application and end-users parts are not in the scope of this paper but I mentioned them for more consistency.
[14] choose the group names that fit your naming convention.



Impact on the running environment

This step has a potential impact on the running environment as the RACF tree structure is modified , but it should be rather limited :

- if you carefully take into account the changes due to the “insertion” of the OWNOSR group in the existing structure
- as the new system support groups are added next to (and not instead of) existing groups

Step 5: Review system-wide security settings

What

System-wide security settings will determine the basic system protection

Why

In this step, you must ensure that the system-wide security settings are correctly defined. These are very important as they impact the integrity of the whole system.

How

In a RACF environment, you should at least evaluate the following settings:

- the password rules quality: e.g. minimum length, change frequency, history of previous passwords, maximum number of tries before userid revocation, etc
- the PROTECT-ALL parameter: must be on FAILURES to require all files to have a RACF profile defined
- the Global Access Table: may not contain system sensitive data
- the RVARY passwords: may not be the default value
- the WHEN(PROGRAM) option: activates access control to load modules and program access to data sets
- JES(BATCHALLRACF): requires batch jobs to run with a RACF-defined user
- inactive RACF resources classes: may not imply the lack of resources protection

Impact on the running environment

As these parameters are system-wide, the potential impact can be significant, unless the changes are carefully planned and prepared.

For instance:

- modifying the password rules will require informing and educating end-users and Helpdesk teams
- modifying the default protection parameters could impact users or jobs accessing resources for which the default access has changed

Step 6: Standardize the Operating Systems Resources

What

In this step, you will identify, categorize [15], audit and protect the Operating System Resources according to their sensitivity and to the new group architecture.

Why

- identify: to distinguish them from the other resources of the system
- categorize: because they require different levels of protection depending on their sensitivity
- audit: as you will need to collect SMF data about OSR usage
- protect: because you will allow access to the new functional groups depending on their needs

In short, standardizing the Operating System Resources will allow to better control them.

How

Proceed as follows:

- identify the OSR and assign them the OWNOSR group as their owning group. Several techniques exist and you probably will have to combine them to "isolate" all OSR:
  - based on the naming conventions
  - based on information from basic system files like the PARMLIB
  - based on product information
  - based on available security tools
  - based on people's knowledge
  - etc

- categorize the OSR by assigning a "sensitivity label" to each resource. In a RACF environment, a good solution consists in using the LEVEL field. It has the advantage to be rarely used on installations and to be ignored by the RACF authorization algorithm. For instance:
  - assign LEVEL(10) to resources requiring UACC [16]=NONE
  - assign LEVEL(11) to resources requiring at the most UACC=READ
  - assign LEVEL(12) to resources requiring UACC=UPDATE

- assign the audit parameters according to the LEVEL value. The principle being generally accepted consists in auditing all access attempts one level above the public access. In practice, this means:
  - for LEVEL(10) resources, set AUDIT(ALL(READ))
  - for LEVEL(11) resources, set AUDIT(ALL(UPDATE))
  - for LEVEL(12) resources, set AUDIT(ALL(CONTROL))

[15] the term « categorize » used here is not related to the RACF CATEGORY field.
[16] UACC stands for universal access or access by default.

- authorize the functional groups defined under the OWNJOBS structure [17] to the OSR according their role and your level of decentralization. Often, the security section of software technical brochures contains guidance about the access rights to be granted to each support team .

Impact on the running environment

This step has a potential impact on the running environment as you will modify ownership but it should be rather limited. At the access level, only additions have been made , nothing has been removed.

[17] see step 4.

Step 7: Activate new Operating Systems Resources protection

What

Now the new OSR protection structure is in place, it has to be activated.
This also means that you will delete references to the (sometimes very old and complex) previous structure that was in place. This is the most challenging and often most labor intensive part of the standardization project.

The security data collected at the beginning [18] were already useful when executing the previous steps. As we now have to perform in-depth and iterative preparatory analysis to avoid disruptions, their importance is emphasized. The results of queries on these data will help you make the correct decisions.

If they are available, you should seek assistance from people having a good knowledge of the system and its "history".

Why

You need better protection for and control over the OSR. Therefore, you must implement an access structure you know and remove the references to the old structure.

How

Proceed as follows:

- where the public access is higher than the required level, lower the UACC value and adapt the access list if needed.
  Example: the SYS1.PARMLIB dataset is public (UACC=READ). As it is a LEVEL(10) OSR, you have to set the UACC to NONE. It means that users or groups that accessed that library thanks to the public access won't have access anymore. You will have to use the different sources of information to determine who really needs access [19].
- remove the old groups from the access lists and challenge the access need for all others, for instance, users groups without system oriented job description.
- remove the userids defined in the datasets HLQ groups. For instance, often, system support users are connected to the SYS1 group. This gives them authority on many resources without being specified in their access list. It should be avoided and replaced by connections in the support groups.
- review the need for system privileges, among others the RACF attributes[20] like SPECIAL, OPERATIONS [21] and AUDITOR

[18] in steps 1 and 2.
[19] in case you have problems to assess who needs access to a resource, consider using the WARNING parameter, but be sure you turn it off as quick as possible.
[20] system wide and at group level.
[21] removing the OPERATIONS attribute also require special attention, especially if it was assigned to application userids or started tasks userids.

Impact on the running environment

As already mentioned, t his is the step with the highest potential impact on the running environment as access rights are “redesigned” in depth. But again, a good preparation will dramatically reduce the risks of disruption .

Step 8: Perform further security actions

What

In this step, you will implement additional security settings, install additional tools and/or integrate subsystem protection into the main security software.

Why

You may be required to further enhance and/or monitor your security in order to comply with your corporate security policy.

You may also decide to centralize the security of subsystems in order to optimize the access management and the control of these subsystems.

How

Each system being different, I will only provide some examples :

- install additional security exits to enforce more severe password rules, or to limit public access definitions, etc
- install additional tools for privileged users administration, for intrusion detection, etc
- implement the full RACF protection on subsystems, e.g. Netview, DB2, SDSF

Impact on the running environment

The potential impact of this step on the running environment will vary in function of the scope and depth of the changes. Installing a tool for privileged users administration will probably have no impact while a password check exit surely will.

Step 9: Validate the new system security status

What

In this step, you will run a security health check to identify remaining security problems in your environment and correct them.

Why

You must ensure that you have effectively reached the security level you targeted.

How

Proceed as follows:

- if you have a health checking tool [22], customize the settings in order to assess your security parameters against the rules in place
- run a health check, via your tool or manually
- correct the findings
- if necessary, fine-tune the parameters of the health checking tool, rerun it and correct again
- once that "one shot" health check has taken place, you may now integrate it in your recurrent activities and schedule a health check run at the frequency required by your corporate security policies

Impact on the running environment

The impact of this step should be minimal. It is dependent on the corrections that need to be made to resolve the health check findings.

[22] it can be the tool you used to standardize the system

Step 10: Validate the new environment

What

As you changed the security environment to some extent, the new situation will have to be validated. This will take place at three levels:

- validate the business need for accessing the system. This is a management responsibility
- validate the business need for OSR access and system privileges. This is a management responsibility
- optionally, validate the new system security status as in step 9. This must be performed by another security specialist, preferably working outside your direct department or from an external company

Why

Before switching to the "daily management" mode, you must ensure that users on the system have a business need and, optionally, have an independent review of the new environment.

How

Proceed as follows:

- perform a userid validation. Send a list of userids to the responsible managers and ask them to validate the need for their employees to access the system. Take corrective actions where needed

- perform an OSR access and system privileges validation. This is a much more detailed report as it goes further than just validating the access to the system. It requires time to review and assumes that the people who validate understand what OSR access and system privileges mean, which should be the case as this concerns managers from the IT department. Take corrective actions where needed

  The report lists should at least contain:
  - users connected to system groups and their access to OSR
  - users with non expiring passwords
  - users with system-wide RACF privileged attributes (i.e. SPECIAL, OPERATIONS, AUDITOR)
  - users with group-wide RACF privileged attributes (i.e. Group Special, Group Operations, Group JOIN, etc)
  - users with RACF Class authorities (i.e. CLAUTH(USER), CLASS(APPL), etc)
  - users with subsystems privileges (i.e. HSM CNTL or USER, DB2 admin, etc)

- request a system security review. Who better than other IT security specialists will be able to evaluate the job you just performed? Therefore, request an

independent team to assess the system security [23] and, if needed, take corrective actions

Impact on the running environment

As step 9, the potential impact of this step on the running environment will mostly depend on the completeness of the previous steps. If your system is at the required level, you won't have to make any changes.

[23] ensure you provide the security rules you have to comply with.

Summary

In this paper, I explained in which context a security enhancement and standardization project generally takes place .

Then, I proposed a “steps to follow” checklist.
In the first 2 steps, you will collect a lot of security related information. It allows you to better perform the next steps. It also increases your level of system auditing and control if you implement data collection in a recurrent mode .

In the third and fourth steps, you will assign userid ownership and define a new role-based group structure. It will enhance your userid and access management.

In the fifth step, you will adapt the system-wide security parameters. It will improve the overall integrity of your system.

In steps six to eight, you will standardize and apply a new access model to the Operating System Resources. It will allow a better protection for and control over the OSR.

In the last two steps, you will check and validate your new environment. You will then be sure that you reached the required level of security before closing the project and moving to the daily management mode.

In summary, these 10 steps complement each other. They all contribute, differently but consistently, to a higher security level.

Your challenge for the future will consist in maintaining the security of your system at the best possible level, while dealing with a constantly evolving environment.

References

SANS Institute, SANS Security Essentials with CISSP CBK Version 2.1, Volume 1, pages 292-294.

IBM Corporation, OS/390: SecureWay Security Server RACF Security Administrator's Guide, Document number: SC28-1915-08

IBM Corporation, OS/390: SecureWay Security Server RACF Auditor's Guide, Document number: SC28-1916-07

Maria Caballero, GSEC Practical Assignment, How to improve security by contracting security outsourcing
http://www.giac.org/practical/GSEC/Maria_Caballero_GSEC.pdf

Rob van Hoboken, Consul Risk Management, Mainframe's midlife crisis: Security
http://www.computerworld.com/securitytopics/security/story/0,10801,89302,00.html?nas=SEC2-89302

Stu Henderson, Interpreting Output from the RACF DSMON Utility
http://www.stuhenderson.com/XDSMNTXT.HTM