Global Information Assurance Certification Paper

GIAC Certification – Version 2.0
Technical Aspect of Implementing/Upgrading SAP Security 4.6
Mary E. Sims
November 15, 2001

Introduction

SAP is one of the most popular ERP systems. This system is made up of multiple modules that correlate to business processes. The modules are typically referred to by a two-letter abbreviation. They are as follows MM (materials management), SD (sales and distribution), CO (controlling – cost accounting), FI (finance), PM (plant maintenance), PS (project systems), PP (production planning), HR (human resources), and PS (project systems). Essentially SAP ~~is~~ can be effectively used as the only system an entity will need to conduct business. This creates a risk for controlling activities within the system. Information for each module is kept in tables and is shared between the modules; therefore incorrect information can create a snowball affect in many processes.

During an SAP implementation or upgrade the security process should be involved in the planning phase to coordinate the security between processes. According to SAP R/3 Upgrade Guide a common problem is security is often thought about towards the end of an implementation/upgrade that can be too late to properly secure the system. This is a detailed process and requires significant coordination between the business processes. This paper is designed to not discuss the coordination between processes but the actual technical aspect of securing the SAP environment and even more specific will discuss in detail controlling security for the SAP Release 4.0 and above.

Overview

Functions are performed through transaction codes defined as a combination of letters and numbers that are used to perform a task. For example, transaction MM01 is used to create a material that resides in a set of tables known as the material master. Prior to SAP release 4.0 the security concept relied on authorization objects when combined to make a profile that allowed ~~a~~ certain transactions to be performed. Users were assigned multiple profiles that were generally task based. The problem with this method was its difficulty in determining what transactions users had access to. Sometimes a combination of profiles would allow not only the transaction codes you wanted to assign to a user but inadvertently would also allow transactions that you did not want to assign.

As of release 4.0, SAP introduced a new concept that would allow restrictions based on specific transaction codes but also to further restrict transaction codes by the use of the authorization objects. The first level of restriction is always at a transaction code level that is controlled by the authorization object S_TCODE. Therefore if a user was not assigned the transaction no matter what authorization objects existed in a users account (buffer) they could not perform the transaction. Most security professionals believe that transactional-based security to be an improvement over the security in previous versions.

When performing an implementation or an upgrade, security leadership should always be involved from the very beginning. This is important because processes need to be compiling what transactions they will need and how these should fall in a user's role. A role is equal to a job such as receiving clerk or accounts receivable clerk. SAP recommends that all transaction codes necessary for a specific role should be combined into one activity group. With this methodology a user should not need more than two to three roles assigned to their user ID. Role based security is superior to task based because buffer problems can result from task based and it is easier to assign composite activity groups to users based on roles rather than based on a task.

An activity group is compilation of transaction codes and authorization objects. An activity group is created through Profile Generator that is transaction code PFCG. Activity groups can then be attached to a composite activity group that is nothing more than a storage place for one or more activity groups. SAP recommends that this should be close to a one to one relationship. The composite activity group is then assigned to the user ID. Keep in mind when using Profile Generator you should always enter the transaction code through the menu tab and never directly into the S_TCODE authorization. If entered into the S_TCODE authorization you will break the relationship between the menu tab and this authorization. When this occurs even if the transaction code is entered into the menu tab it will not be automatically added to the S_TCODE authorization and the users with this activity groups will not have access to this transaction code. The S_TCODE authorization is automatically updated when a transaction is entered in the menu tab.

Buffer Problems

Buffer problems will occur when a user is assigned multiple activity groups. This can occur through assignment of multiple composite activity groups or if the composite activity group has multiple activity groups. A security administrator can usually diagnose a buffer problem because it appears the user has been assigned access to the transaction that SAP is denying. This is because the user buffer is full and authorization objects begin to fall out of the user buffer or the whole buffer shuts down. Most of the time this can be fixed by saving the user master record. If this repeatedly occurs the user ID should be reviewed and activity groups need to be combined or access needs to be removed.

You can determine if you have users with buffer problems by reviewing table USR04 and looking for users who have approximately 300 profiles. The user buffer exceeds the maximum around 312 profiles or 3000 authorization objects. SAP automatically creates profiles based on the number of authorization objects. However this does not mean unique authorization objects it is simply the number of authorization objects listed in the activity group.

This problem is easily prevented through using role based activity groups rather than task based. This makes a case for ensuring that security leadership is brought in during the planning phase. This will prevent user frustration after the system is live. Also when creating an activity group in Profile Generator the utilities => merge authorizations function should be used to merge all like authorizations together. This will decrease the number of profiles created for each activity group.

Authorization Objects/Transaction SU24

Profile Generator uses the information in transaction SU24 to determine what authorization objects are needed for a transaction code. SU24 is made up of multiple tables that are brought together by the program SAPMS921. Sometimes this information is incorrect and will result in authorization objects incorrectly added or omitted from an activity group. This is why positive and negative testing is a must before the system is implemented. SAP allows alteration of the information in SU24, thereby changing the authorization objects that Profile Generator uses for specific transaction codes. This is useful because you can prevent Profile Generator from bringing in certain authorizations that are more powerful and should be used more sparingly than how they are delivered in SU24. The following are a list of these authorizations that should be deactivated from the SU24 and only added after testing has occurred and the reasons for its use is documented.

- **S_TABU_DIS** – This authorization allows table maintenance that typically should only be allowed in the development systems and transported forward into production. This prevents the systems from becoming out of sync. This authorization should only be assigned in production in very rare instances. It is acceptable to assign the display version of this authorization in production. SAP assigns tables to authorization groups and these authorization groups are used by S_TABU_DIS to assign tables for security access. The problem is that SAP may assign as many as 1500 tables to an authorization group. There are rare instances where table changes are appropriate in production. When this occurs a custom authorization should be created with only the tables appropriate for the production change.

- **S_TABU_CLI** – This allows cross client table maintenance. Most tables in SAP need to be transported to other clients and systems in order for the change to take affect. However there are a few tables that will automatically update other clients. These tables are called client dependent In these instances this authorization is needed and should be handed out sparingly.

- **S_DEVELOP** – This authorization allows users with developer keys to make changes to and develop new program code, transaction codes, and authorizations. This authorization should only be given in development. These types of changes should be transported into production in order to keep the systems in sync.

- **S_PROGRAM** – This is used to restrict use of programs. SAP delivers many programs that can be accessed through transaction SA38. These programs often have the authorization S_PROGRAM attached to them or the developers often use the authorization in custom-built programs. They will assign authorization groups that are used by this authorization object. If a user does not have the appropriate authorization group they will not have access to this program. As of version 4.6 programs can be assigned transaction codes that alleviates the need to assign the transaction SA38 to users.

This transaction is dangerous because if a sensitive program does not have security assigned to it the user will be able to run it with the SA38 access. If a user does not have SA38 then they will need to have the transaction code assigned to the program. It is important for security to ensure that the developers have implemented a procedure for assigning security to newly developed programs.

- **S_ADMI_FCD** – This authorization object in combination with the S_TCODE authorization can allow sensitive system altering access. There are several system functions that can be performed through this authorization object. For example, you can give a user transaction code SM04 through authorization object S_TCODE that allows a user to view all users currently active on SAP. However if you give the user the additional authorization S_ADMI_FCD along with the system function PADM they will be able to terminate another user's session.

- **S_BTCH_ADM** – If this authorization is given independent of S_BTCH_JOB and S_BTCH_NAM the user will be able to create, change, copy, delete, and start all background jobs. This authorization should only be given to a basis role.

- **S_BTCH_NAM** – Allows restrictions for creating and viewing batch jobs. You can restrict to a user's own jobs and/or other users' jobs. Users should not have access to all batch jobs because sensitive data could reside in these jobs or significant changes could be made to the system if a user runs a job they are unfamiliar with.

- **S_BDC_MONI** – This allows restrictions based on the job name. For example you may only want a user to create, change, or delete jobs beginning with certain letters. This is used very affectively if a good naming convention is in place for job names.

- **S_USER * -** All authorizations beginning with S_USER should only be given to security professionals. All these authorizations control the Profile Generator and user master functions.

Sometimes you will still want to change an authorization brought in by Profile Generator, but you do not want to change the SU24. For example, a transaction code can be changed from allowing change to display only by changing the authorization object from change to display activity. You may only want display in certain roles but change in others. The SU24 is based on a transaction and does not change based on the intent. This change must be made at the activity group level. In order to prevent the change authorization from reappearing if you make changes to the display activity group, the change authorization brought in by Profile Generator must be inactivated and then manually add the display authorization. This will ensure the integrity of your display roles.

Security Related Parameters

The following parameters should be set in order to increase the security aspect of your system.

You should review program RSUSR003 in order to quickly determine how the following parameters are configured.

- **Parameter login/failed_user_auto_unlock** controls the unlocking of users locked by logging on incorrectly. If the parameter is set to 1, the system will automatically unlock a user at midnight. If the parameter value is set at 0 the lock will not be removed. It is recommended to set this value at 0.

- **Parameter login/fails_to_session_end** controls the number of times a user can enter an incorrect password before the system terminates the logon session. SAP recommends three attempts. This does not mean that the user cannot attempt to try again. The parameter **login/fails_to_user_lock** determines the number of times a user can try to restart their session and try the password again. SAP recommends three but I would only recommend two. When this set to three a user can try nine different passwords. They may have to put up with the hassle of starting over three times but this will not deter a persistant hacker.

Passwords

Sap offers a few methods of contolling password selection by users. These methods are discussed below.

- The parameter Parameter login/min_password_lng determines the length of a password. SAP recommends at least 5 characters but I recommend 8. Parameter login/password_expiration_time allows you to determine how many days before a password expires. SAP recommends 45 to 60 days.

- Users should not use passwords such as company name, city name, and popular terms for example the word password and any variations of this word. This can be accomplished through adding these to the table USR40. For example, if you do not want a user to use any variation of the word password you could enter *pass* into this table. This will decrease the likely hood of users hacking into the system with someone else's ID.

SAP delivers the user IDs SAP*, DDIC, and SAPCPIC with full authorizations that will allow a user to perform any SAP function. All of these user IDs are delivered with a standard password 'Admin'. This password is widely publicized in SAP literature such as Authorizations Made Easy and System Administration Made Easy and should be changed immediately. If the IDs are not actively used they should also be locked. In particular SAPCPIC should be locked. SAPCPIC is used for communicating between processes/hosts. This is very dangerous because an outside hacker can use this ID to send unauthorized data into the SAP system.

Developing Roles for Power Users

The only two profiles that are still available for 4.0 or greater versions are SAP_ALL and SAP_NEW that allow total unrestricted access. These profiles are also dynamic, meaning that any new authorizations are automatically added to these profiles. This access should never be given to any user in any system other than the sandbox. Do not be fooled into believing that this access is needed because it is too difficult to develop the access that the user needs. The users should be able to keep track of what functions or processes are needed. Also a user can be traced by using transaction code ST01 to determine what authorization objects are necessary.

Power users such as Configurators, Developers, and Basis users are often the most troublesome at demanding all access. In development systems these users should be granted significantly more access than production however they still should not be given access that falls outside of their areas of expertise.

Often times these users will claim that they need access to all transaction codes. SAP delivers somewhere around 20,000 transaction codes but most companies only use around 2,000. SAP delivers a tool called Luminate that allows you to determine which transaction codes are used. If this is an implementation from a legacy system you should only assign the transaction codes that have been determined necessary by the processes. Most SAP systems have a sandbox and only in the sandbox should unrestricted access be allowed. The sandbox is usually a self contained SAP environment that does not allow transporting into other systems, therefore, it is safe to play without affecting your production environment.

In lieu of giving power users total unrestricted access in development systems the following should be developed. These methods will also easily allow the addition of transaction codes and will easily give all power users this access.

One activity group should be developed with all process transactions codes. A process transaction code is used to perform daily functions in SAP. These transactions can usually be found by using the SAP standard menu.

Activity groups should be built for each piece of the IMG that is located by using the transaction SPRO. The IMG is made up of transaction codes that are used to configure or customize SAP to your environment. Change access should only be given in the proper system and then transported to all other systems. This will keep all systems in sync. The IMG is broken down into processes so an activity group should be developed for each process. Due to the significant number of transactions in each process a special tool is needed called Project IMG. This is performed through the Profile Generator.

An activity group should be built for transactions that are considered system transactions. A system transaction affects the actual operational aspect of SAP. These transactions are usually located under tools on the SAP standard menu. These transactions should be given out sparingly. The best method is to conduct a meeting with the team leads of the developer, basis, and configurator groups and come to an agreement on which transactions should be assigned to which group of power users.

Often power users push to have batch IDs with SAP_ALL and SAP_NEW. This also is not necessary. A batch ID is used to run programs in the background or set up programs to run on regular intervals. It is often more difficult to find the security linked to programs but with persistence it is possible. All programs can be run in the foreground using transaction code SA38. When a program is run in foreground an authorization error message will occur when the program fails, at that point use transaction SU53 to troubleshoot the missing security. A trace can also be used to list the authorization objects used by the program. Also transaction code SE80 can be used to view the programming code. All authorization objects in a program are called authority-checks. The find feature can be used to list the authorization objects used by the program.

If all else fails SAP offers a service called SAPNet or OSS notes which allows users to ask questions or search for previously asked questions and their answers. SAPNet can be helpful in finding missing security that you are unable to troubleshoot.

System Reconciliation

Even though much of SAP is very sophisticated there are still some processes that must be performed somewhat manually on a regular basis. One of these processes is a report called User Master Data Reconciliation that can be located using the transaction PFUD. This report syncs up the user master record to the activity group. For example if you change the activity groups assigned to a composite activity group this will not automatically occur for the users. This report must be run to sync up the change in the user master. Also if you terminate a composite activity group or activity group in a user master the termination does not take affect until this report is run. It is recommended that this report is run nightly when very few users are on the system. This report does run for a very long time and uses significant system resources.

Security Audit

When you have finally implemented the transaction based security there is still a need to audit the security on a regular basis. The activity groups are very dynamic and are often in a state of change. A new transaction code will be needed or more/less authorization restriction is needed for a role. This will be ongoing for the life of your system. Auditing is also needed, in order to prevent what is referred to as "access creep" that is simply letting a user gain more access without reevaluating the current access. This could result in the user have greater access then intended. SAP's security philosophy is the least restrictive access assigned to the user record will prevail.

A tool that can be used to audit system changes is the Infosystems Authorization report which is located at transaction code SUIM. This report can be used to monitor changes made to user masters, and activity groups. There are however some deficiencies in this report. There are pieces of it that do not work properly. SAPnet offers several coding changes that can be applied to help with the accuracy of this report. This report should also be used to monitor the powerful authorizations mentioned above to ensure that they do not appear in activity groups. According to sapstuff.com there is a program called RSUSR005 that lists users with excessive authorizations.

Conclusion

This paper has outlined a few of the important aspects of upgrading or implementing transactional based security. These guidelines help ensure that users will have the access they need while preventing them from performing functions that do not fall within their role or allowing access to sensitive company data.

[1] SAP R/3 Upgrade Guide. SAP Labs, Inc. 2001.
http://wwwtech.saplabs.com/docs/sysadmin/upgrades.pdf

[2] SAP R/3 Security Tip Page.
http://www.sapstuff.com/sap/tipsecure.shtml

[3] Sachar, Paulus. "The New mySAP.com User Administration Concept". 2001.
http://www.sapinfo.net/goto/tech/5051/?session=f882082be125d70a37d71d9b37b40450

[4] Authorizations Made Easy 4.6A/B. SAP Labs, Inc. R/3 Simplification Group. 2000.

[5] System Administration Made Easy 4.6A/B. SAP Labs, Inc. R/3 Simplification Group. 2000.

[6] ASAP World Consultancy, Elkington, Blain. Special Edition Using SAP R/3: The Most Complete Reference. Que Publishing. 1999.