RBAC: Role Based Access Control in AIX

Contributed by Michael Felt

RBAC or Role Based Access Control has been available in AIX since 1997 starting with AIX 4.2.1. Prior to that, access control is AIX was the same as for any other UNIX or UNIX variant. Yes, access control (DAC, or discretionary access control), but no role based management of lists of authorizations or priviledges to execute sets of commands.

In short AIX has known three levels of RBAC:

1. No RBAC

2. Limited RBAC

3. Extended RBAC (starting in AIX6)

This article describes briefly the first two stages of RBAC in AIX and introduces the core concepts in AIX extended RBAC. Further articles will discuss the implementation and usage of extended RBAC.

No RBAC: Traditional UNIX SecurityAccess Control in UNIX has always been available. The mechanism is based on the principles of DAC - Discretionary Access Control. Systems based on DAC have a concepts of objects, owners, groups and others. Every object is owned by a single user, with additional access controlled via group membership (group permissions), or anyone else (others, i.e. neither user nor group member). The owner has the privlidge (discretion or right) to determine who has access to an object (i.e. a file): user only, group only, others only, or any possible mix. Also, the owner can modify object accessibility at any time (i.e. at their DISCRECTION). Simply stated, on UNIX with DAC mechanisms - if you are the owner only owner settings apply (except that you always retain the right to change those settings), if you are not the user and are in the group specified only the group settings apply; if you are neither the owner nor in the group, only the other settings apply.

Limited RBACIntroduced in AIX 4.2.1 in 1997 AIX introduced what it called "User Roles" as an enhancement on the AIX user and group security methods that previously had not known any role based controls, only the traditional UNIX user, group and other DAC permission settings. The system works by having front-end programs that are accessible via group or other permission bits. The first task of this role-based program is to verify that the user has the appropriate role to use the program. If this is true, the program executes a SETGID or SETUID (to root) depending on the permission level of the program performing the verification. Each program verifies the users roles (e.g., basic or advanced role) before performing any function.

Although easy to use and manage by a system administrator, it was very difficult to adopt to programs not specifically coded to use the AIX Role mechanism and has remained limited to common tasks: queue control, user administration, backups, and system shutdown. The great advantage is that these tasks could be performed by users who were neither system administrators (in the strict sense) nor did they ever gain root access (prompt). People who considered this approach too limited generally opted for the package sudo - and accepted both the additional risks and workload associated with it use and administration.

Extended RBAC and Legacy RBAC

Starting with AIX6, system administrators can choose between Legacy RBAC (the old system introduced in AIX 4.2.1) or enhanced RBAC. New installations will have extended RBAC activated by default. Legacy RBAC will still be supported but IBM is recommending that administrators switch to Enhanced RBAC. Legacy RBAC provides several pre-defined roles that can be setup administrative users that can perform specialised tasks without any need for root access. Legacy RBAC also provides a framework for extending the pre-defined roles but it is quite difficult to use.

Extended RBAC

Extended RBAC is granular. Using Extended RBAC an administrator can specify any command as a privledged command. Rather than being coded into (front-end) applications that rely on setuid/setgid file modes enhanced RBAC uses a security database that is located in local files, or optionally, in an LDAP server. The data is stored in "flat-file text" so no additional database management engine is needed to use enhanced RBAC.

There are five (5) components to the RBAC security database:

- Authorizations
- Roles
- Privileged Commands
- Privileged Devices
- Privileged Files

Basically, in enhanced RBAC we need to distinquish three concepts: Authorizations, Roles, and Privileges.

Authorizations: The basic question is: can a user perform a specific task. If he has access to an authorization(s) (similiar to a key to open an otherwise locked door(s)) the task can be performed. Otherwise the task or resource remains unaccessible.

Roles: A role is a list of all the authorizations needed to complete a task. Authorizations get assigned to one or more roles; roles get assigned to users. Note: authorizations do not get assigned to users!

Priviledges: A priviledge is an explicit access granted to a command, device, or file. Priviledges are assigned to users. Using priviledges the LPP (Least Priviledge Principle) can be implemented without any code change to existing programs and can lessen the dependency on setuid/setgid file permissions.

Pre-defined Roles in Extended RBAC

ISSO - Information System Security Officer

The ISSO role manages all other roles. This makes it the most powerful role on the system. Some of the ISSO tasks or responsibilities are:

- Establishing and maintaining security policy
- Setting passwords for users
- Network configuration
- Device administration

SA - Systems Administrator

The SA role provides authorizations for daily administration and includes:

- User administration (except password setting)
- File system administration
- Software installation update
- Network daemon management
- Device allocation

SO - System Operator

The SO role provides the authorizations for day to day operations and includes:

- System shutdown and reboot
- File system backup, restore and quotas
- System error logging, trace and statistics
- Workload administration