Tech Info Library

ABS Tech Note: AWS16 File Access Control on the AWS 95 (1/94)

Revised: 1/11/94
Security: Everyone

ABS Tech Note: AWS16 File Access Control on the AWS 95 (1/94)

==========

Article Created: 23 December 1993

TOPIC

The purpose of this tech note is to differentiate Macintosh (AppleShare) file privileges and A/UX file permissions. This document will discuss the concepts involved with each form of file access control and also discuss how the two concepts were integrated with each other on the Apple Workgroup Server (AWS) 95.

DISCUSSION

AppleShare (Macintosh) File Access Privileges

The Macintosh was originally designed as a single-user machine, where all the files on the machine were only accessible by one user. As a result there is no support for file access control built into the file system. With the advent of AppleShare it was necessary to come up with a scheme to control access to files which were shared on a network. On a Macintosh which runs as an AppleShare file server there exists a special file which stores all the necessary information to determines the accessibility of files to network users (See the AppleShare Pro Server 1.0 Administrator's Guide for a detailed description of AppleShare file privileges).

A/UX File Permissions

UNIX was originally designed as a multi-user operating system, thus a file access control mechanism was designed into the file system from the start. On each UNIX system there is a list of users which can log on to the system, each user has a unique name and user ID. In order to provide more control and flexibility when setting up file accessibility UNIX also maintains a list of groups, each group has a unique name and group ID. A group is defined by a list of 0 or more users. Each user must be a member of at least one group, known as the user's primary group, but may also be a member of any of the other group.

For each file in a UNIX file system, in addition to the data content of the UNIX file there is also information stored which determines file access. Each UNIX file has associated with it the following data: An owner (usually the user which created the file); a group (usually the primary group of the user which created the file); and a set of 9 permission bits.

The 9 permission bits associated with a file can be broken down in 3 parts, leaving 3 bits for each part. Each set of 3 bits defines the level of access for the file depending on which user is accessing the file. The first 3 bits (reading from left to right) determines the level of accessibility for the owner of the file. The second 3 bits apply to users who are a member of the group which is assigned to the file. The third 3 bits apply to any user who does not fall in either of the first two categories. These 3 user types are termed "User, Group and Other", respectively.

Now for the interpretation of the 3 bits. The first bit (read from left to right) determines whether the user has read access on the file. The second bit determines if the user has write access on the file. The third bit determines if the user has execute permission on the file.

Read and write access on a file is pretty much self-explanatory, if a user has read permission on a file then they may read the contents of the file, if a user has write permission on a file then they may modify the contents of a file. Execute permission determines if a user can execute a file, in the case where the file is an executable program. In the case where the file in question is a directory the execute bit determines if a user may make the directory their current working directory ("cd" into it).

Integration of both File Access System on the AWS 95

On an Apple Workgroup Server 95, somehow both file access systems apply to some of the files and how this is done can sometimes be confusing. There exists 2 separate list of users and each file has it's UNIX permissions as well as it's AppleShare permissions. In order to explain how this integration is achieved the concept of the "Super User" or "root" must be understood. On every UNIX machine there exists a special user, "root", to which file access control does not apply. In effect the "root" user has read, write and execute permission on every file in the file system.

On the AWS 95 the AppleShare program (the program which provides file sharing service to network users) runs as a "root" process. That is, all file accesses made by the AppleShare process are made from the "root:" user, which means that the AppleShare process has unrestricted access to all UNIX files. This provides AppleShare users with the illusion that an AppleShare volume on an AWS 95 is the same as a volume on a regular Macintosh.

This integration can often confuse administrators and users since one would expect a file to be owned by the AppleShare user, but if we go look at it from the UNIX side it is owned by the "root" user.

Tech Info Library Article Number:14324