NAME

auth_attr − authorization description database

SYNOPSIS

/etc/security/auth_attr

DESCRIPTION

/etc/security/auth_attr is a local source for authorization names and descriptions. The **auth_attr** file can be used with other authorization sources, including the **auth_attr** NIS map and NIS+ table. Programs use the **getauthattr**(3) routines to access this information.

The search order for multiple authorization sources is specified in the **/etc/nsswitch.conf** file, as described in the **nsswitch.conf**(4) man page.

An authorization is a right assigned to users, which is checked by certain privileged programs to determine whether users can execute restricted functionality. Each entry in the **auth_attr** database consists of one line of text containing six fields separated by colons (:). Line continuations using the backslash (\) character are permitted. The format of each entry is:

```
name:res1:res2:short_desc:long_desc:attr
```

name

The name of the authorization. Authorization names are unique strings. Construct authorization names using the following convention:

prefix.

or

prefix.suffix

prefix is everything in the name field up to the final dot (.). Authorizations from Sun Microsystems, Inc. use **solaris** as a *prefix.* To avoid name conflicts, all other authorizations should use a *prefix* that begins with the reverse−order Internet domain name of the organization that creates the authorization (for example, **com.xyzcompany**). Prefixes can have additional arbitrary components chosen by the authorization's developer, with components separated by dots.

suffix is the final component in the name field and specifies what is being authorized.

When there is no suffix, the name is defined as a heading. Headings are not assigned to users but are constructed for use by applications in their GUIs.

When a name ends with the word **grant**, the entry defines a grant authorization. Grant authorizations are used to support fine-grained delegation. Users with appropriate grant authorizations cay delegate some of their authorizations to others. To assign an authorization, the user needs to have both the authorization itself and the appropriate grant authorization.

res1

Reserved for future use.

res2 Reserved for future use.

short_desc A short description or terse name for the authorization. This name should be suitable for displaying in user interfaces, such as in a scrolling list in a GUI.

long_desc A long description. This field can explain the precise purpose of the authorization, the applications in which it is used, and the type of user that would be interested in using it. The long description can be displayed in the help text of an application.

attr An optional list of semicolon-separated (;) key-value pairs that describe the attributes of an authorization. Zero or more keys may be specified. The keyword **help** identifies a help file in HTML. Help files can be read by a web browser using the URL: *file:/usr/lib/help/auths/locale/C/index.html*.

EXAMPLES

In the following name, a prefix (**solaris.**) is followed by a suffix (**printer**):

solaris.printer

Because the name field ends with a dot, the following entry defines a heading.

solaris.hostmgr.:::Computers & Networks::help=HostMgrHeader.html

In the next example, a heading entry is followed by other associated authorization entries. The entries below the heading provide separate authorizations for setting user attributes. The attr field for each entry, including the heading entry, assigns a help file. The application that uses the **help** key requires the value to equal the name of a file ending in **.htm** or **.html**.

solaris.usermgr.:::Users, Groups & Email Aliases::help=UserMgrHeader.html
solaris.usermgr.pswd:::Change User Passwords::help=UserMgrPswd.html
solaris.usermgr.write:::Add, Modify & Delete::help=UserMgrWrite.html

This example assigns to an administrator the following authorizations:

solaris.printmgr.grant
solaris.printmgr.admin
solaris.printmgr.nobanner
solaris.login.enable

With the above authorizations, the administrator can assign to others the **solaris.printmgr.admin** and **solaris.printmgr.nobanner** authorizations, but not the **solaris.login.enable** authorization. If the administrator has both the grant authorization, **solaris.printmgr.grant** and the wildcard authorization, **solaris.printmgr.***, the administrator can grant to others any of the printer authorizations. See the **user_attr** man page for more about how wildcards can be used to assign multiple authorizations whose names begin with the same components.

The following entry defines an authorization that grants the ability to assign any authorization created with a **com.sun** prefix, when the administrator also has either the specific authorization being granted or a matching wildcard entry.

solaris.!!grant:::Grant All Rights::help=PriAdmin.html

With the following entry from **/etc/nsswitch.conf**, the local **auth_attr** file is consulted before the NIS table.

auth_attr: files nisplus

FILES

/etc/nsswitch.conf

/etc/user_attr

/etc/security/auth_attr

CAVEATS

When deciding which authorization source to use (see **DESCRIPTION**), keep in mind that NIS+ provides stronger authentication than NIS.

Because the list of legal keys is likely to expand, any code that parses this database must be written to ignore unknown key-value pairs without error. When any new keywords are created, the names should be prefixed with a unique string, such as the company's stock symbol, to avoid potential naming conflicts.

Each application has its own requirements for whether the help value must be a relative pathname ending with a filename or the name of a file. The only known requirement is for the name of a file.

The following characters are used in describing the database format and must be escaped with a backslash if used as data: colon (:), semicolon (;), equals (=), and backslash (\).

SEE ALSO

exec_attr(4), **getauthattr**(3), **getexecattr**(3), **getprofattr**(3), **getuserattr**(3), **nsswitch.conf**(4) and **user_attr**(4)