Subject: Execution Profiles for Restricted Environments

Submitted by: Glenn Faden, Gary Winiger

File: PSARC/1997/332/opinion.ms

Date: April 21, 1999

Committee: Brian Wong, Ralph Campbell, Joseph Kowalski, Tim Marsland, David Robinson, Andy Rudoff, Steve Zucker

Steering Committee:
Solaris Operating Environment Steering Committee (soesc@sac.eng)

1. Summary

This project provides the ability to assign arbitrary sets of attributes to users and roles. Attributes specify which accounts are roles, which roles are granted to users, and the specific authorizations and execution profiles which have been granted to an account. In addition, new attributes can be defined without any changes to the API.

A role is a type of shared account which can only be accessed through a secondary login mechanism such as su(1M), and is only available to specified users.

An authorization is a right granted to a user or role to perform a function which is not generally allowed. Trusted applications check that a user (or role) is authorized before performing restricted operations.

A profile is a bundling mechanism that contains the list of commands and authorizations required to perform a particular function. For each command, the profile may specify the effective user id and/or the effective group id to be used when executing the command. Authorizations and commands from multiple profiles are additive. Assigning a profile to a role allows users who can assume that role to perform specific privileged operations without giving users access to all superuser powers. For example, a user might be able to administer the printer subsystem, but not add users or change the password of other users.

This project provides a CLI for assigning roles, authorizations, and profiles to users and a ''C'' API, the data structures and database formats, and conventions for using authorizations, roles, and profiles.

2. Decision & Precedence Information

This project is approved as specified in reference [1].

This project may be delivered in a minor release of Solaris.

This project and the following project are required to coordinate to ensure that the databases introduced by this project are available through LDAP.

PSARC/1998/361 Native LDAP

3. Interfaces

The project exports the following interfaces.

Interfaces Exported		
Interface	Classification	Comments
auths(1)	evolving	
pfcsh(1)	evolving	
pfexec(1)	evolving	
pfksh(1)	evolving	

Interfaces Exported		
Interface	Classification	Comments
pfsh(1)	evolving	
profiles(1)	Stable	
roles(1)	Stable	
roleadd(1M)	Stable	based on useradd(1M)
roledel(1M)	Stable	based on userdel(1M)
rolemod(1M)	Stable	based on usermod(1M)
useradd(1M)	Stable	New options for authorizations and profiles
userdel(1M)	Stable	Updates additional files
usermod(1M)	Stable	New options for authorizations and profiles
kva_match(3)	Stable	
getauthattr(3)	Stable	
getauthnam(3)	Stable	
setauthattr(3)	Stable	
endauthattr(3)	Stable	
free_authattr(3)	Stable	
chkauthattr(3)	Stable	
getexecattr(3)	Stable	
getexecuser(3)	Stable	
getexecprof(3)	Stable	
setexecattr(3)	Stable	
endexecattr(3)	Stable	
match_execattr(3)	Stable	
getprofattr(3)	Stable	
getprofnam(3)	Stable	
setprofattr(3)	Stable	
endprofattr(3)	Stable	
free_profattr(3)	Stable	
getuserattr(3)	Stable	
getuserid(3)	Stable	
getusernam(3)	Stable	
setuserattr(3)	Stable	
enduserattr(3)	Stable	
free_userattr(3)	Stable	
getauusernam(3)	Stable	Existing interface updated to use nsswitch
getauusernam_r(3)	Stable	getauusernam(3)
pam_role_auth.so.1	Evolving	pam_role_auths(5)
libsecdb.so.1	Stable	kva_match(3)

Interfaces Exported		
Interface	Classification	Comments
auth_attr(4)	Evolving	Extensible without breaking backward compatibility
exec_attr(4)	Evolving	Extensible without breaking backward compatibility
prof_attr(4)	Evolving	Extensible without breaking backward compatibility
user_attr(4)	Evolving	Extensible without breaking backward compatibility
audit_user(4)	Evolving	No change to database, now supports nsswitch
policy.conf(4)	Evolving	

The project imports the following interfaces.

Interfaces Imported		
Interface	Classification	Comments
_nss_XbyY_fgets	consolation private	libc promoted from not visible to Sun Private

4. Opinion

There were a number of issues raised at the commitment review. These were responded to in [2]. The primary issue was that no bundled administrative interfaces were being supplied. The project team felt that the unbundled Seabreeze project (LSARC/1998/476) which also uses and manages these databases should be the administrative interface. The committee felt that it was not sufficient to depend on an unbundled product as the only administrative interface. The project team responded by modifying existing bundled CLIs and adding new CLIs.

A secondary issue was the interdependence of this case with PSARC/1997/331 (Process Privilege Mechanisms) or PSARC/1997/334 (Security Policy Hook Architecture). The project team assured the committee that there was none beyond a shared configuration file that was to be presented with PSARC/1997/334. That file (policy.conf(4)) is now included with this case.

4.1. Authorization Names

There was discussion about the syntax of authorization names. The project team has proposed ''com.sun.<subsystem>.<function>''. Members of the committee thought the ''com.sun.'' prefix was perhaps too long, too internal sounding, or potentially objectionable to Solaris OEMs. Alternative proposals of ''solaris.'', ''sol.'' or ''sunw.'' were made. The project team said they were open to a required technical change. However, since the Seabreeze project (LSARC/1998/476) was already in beta test with the ''com.sun.'' prefix, the project team did not feel it was appropriate to change the prefix without a required change from the committee.

Any change required must be coordinated with an LSARC required change for Seabreeze.

Tim Marsland contacted the ''OEM ready'' team on behalf of this case. Based on the response from the ''OEM ready'' team, the project team has changed the ''com.sun'' prefix to the recommended ''solaris'' prefix and coordinated the change with the Seabreeze project.

4.2. Authorization and Keyword Registry

The naming convention for authorization names is adequate for external customers to manage their own name space. Within Sun and Solaris the naming convention requires a registry for authorization names. There being no other established registry agent, the project team volunteered to act as the registry. A mail alias will be established for this purpose.

Unlike authorization names which are hierarchical and have a partitioned name space, keywords in the databases reside in a flat name space. The project team advised that keywords for groups outside Sun should use a unique prefix. The project team volunteered to act as the registry for Sun. A mail alias will be established for this purpose.

4.3. LDAP

The databases as presented do not support LDAP because there is no native LDAP support in the current gate. The project team plans to assist the PSARC/1998/361 project team to include these databases in the native LDAP support.

4.4. Increased Use of setuid Programs

Finally, there was some discussion of the fact that more programs would be setuid-enabled. These programs are characterized as those that interpret authorizations and are not presently setuid-enabled. For programs which interpret authorizations, the project team intends to implement the ''Principle of Least Privilege'' by turning off the effective ''root'' uid at the beginning of the program and only turning it back on for the required privilege operations. For programs which do not interpret authorization, but need to be run as ''root,'' the role mechanism and the use of execution profiles to enable ''root privilege'' for select users of those programs provides an advantage over running a ''root shell''. The committee concluded that this arrangement is more secure from a technical standpoint, but a minority expressed some skepticism over the notion that more (perhaps, in the future, many more) setuid programs would be viewed negatively by the market. The steering committee is advised to ensure that proper marketing precedes this product's launch and implementation in order to address such concerns.

5. Minority Opinion(s)

None.

6. Advisory Information

6.1. Bundled Administrative Tools

The committee felt that the presently bundled tools for administering Solaris databases are inadequate. The team had expected to rely on the unbundled Seabreeze product (LSARC/1998/476) for administration. When advised by the committee that unbundled administration was not acceptable, the project team responded by modifying the existing useradd/del/mod(1M) CLIs and adding new roleadd/del/mod(1M) CLIs. While such CLIs are acceptable for this case, the project team points out that they would have preferred to follow a more encompassing administrative framework and would have done so if it were bundled. The committee agreed, observing that the issue might have been moot had Seabreeze been a part of Foundation Solaris. The project team is working with the Seabreeze team to provide an encompassing administration framework for user databases.

Both the Seabreeze framework and the Viper framework (the follow-on to Solaris Management Console LSARC/1999/313) make use of this project's user attribute and authorization databases and mechanisms.

6.2. Future Projects

As part of mail correspondence subsequent to the review meeting, the question of the use of this case in future projects was discussed. In order to make full use of this case for administration and control in Solaris the committee recommends that future projects to formalize roles and the use of authorizations in Solaris be created.

7. Appendices

7.1. Appendix A: Technical Changes Required

None.

7.2. Appendix B: Technical Changes Advised

None.

7.3. Appendix C: Reference Material

1. Final specification. File: PSARC/1997/332/commitment.materials/*
2. Mail. File: PSARC/1997/332/mail

