Sun Microsystems Systems Architecture Committee	
Subject	Least Privilege for Solaris
Submitted by	Casper Dik
File	PSARC/2002/188/opinion.html
Date	January 15th, 2003
Committee	Andrew Tucker, Ralph Campbell, James Carlson, Joseph Kowalski, Richard McDougall, Terrence Miller, Glenn Skinner.
Product Approval Committee	Solaris PAC

1. Summary

This project adds support for fine-grained process privileges to Solaris. This allows processes to perform certain privileged operations, such as binding to a reserved port, without having full root privileges. It also allows the powers of root processes to be restricted to a subset of those normally available.

2. Decision & Precedence Information

The project is approved as specified in reference [1], but as modified by the required technical changes listed in Appendix A. The project may be delivered in a minor release of Solaris.

3. Interfaces

Interfaces Exported		
Interface Name	**Classification**	**Comment**
/etc/security/extra_privs	Project Private	Additional privileges
/etc/security/device_policy	Project Private	Device privilege file
/etc/security/priv_names {LC_MESSAGES dir}/priv_names	Stable	Support files for priv_gettext
/usr/include/priv.h	Stable	Location
/usr/include/ucred.h	Stable	Location
/usr/include/sys/priv.h	Stable	Location
/usr/include/sys/priv_names.h	Stable	Location
/usr/include/sys/policy.h	Stable	Location
/usr/bin/ppriv	Invocation: Evolving Output: Unstable	Process privilege utility

/usr/sbin/getdevpolicy	Invocation: Evolving Output: Unstable	Device policy inspection utility
pfexec	Evolving	New -P option
add_drv	Evolving	New -p and -P options
update_drv	Evolving	New -p and -P options
setppriv getppriv	Evolving	System calls to set/get privilege set
setpflags getpflags	Evolving	System calls to set/get process flags
priv_str_to_set priv_set_to_str priv_getbyname priv_getbynum priv_getsetbyname priv_getsetbynum priv_gettext	Evolving	Privilege name functions
priv_set priv_ineffect	Evolving	Functions to set/check privileges
priv_allocset priv_freeset priv_emptyset priv_fillset priv_isemptyset priv_isfullset priv_isequalset priv_issubset priv_intersect priv_union priv_inverse priv_addset priv_copyset priv_delset priv_ismember	Evolving	Privilege set manipulation functions
ucred_get ucred_free ucred_geteuid ucred_getruid ucred_getsuid ucred_getegid ucred_getrgid ucred_getsgid ucred_getgroups ucred_getprivset ucred_getpid ucred_getpflags	Evolving	Credential access functions
door_ucred	Evolving	Returns door client credential
Privilege constants	Stable	See [1] for list

MODSETDEVPOLICY MODGETDEVPOLICY MODALLOCPRIV MODGETDEVTDEVPLCY	Project Private	modctl subcommands
AUE_SETPPRIV AUE_MODDEVPLCY AUE_MODALLOCPRIV	Stable	New audit events
NT_PRPRIV NT_PRPRIVINFO	Stable	New ELF notes
AT_SUN_AUXFLAGS AT_SUN_SETUGID	Consolidation Private	New aux vector attributes
SYS_privsys	Consolidation Private	New system call entry point
PCSPRIV prpriv_t pr_errpriv	Stable	/proc privilege information
prof_attr	Evolving	New "privs" attribute
user_attr	Evolving	New "defaultpriv" and "limitpriv" attributes
DB_CRED DB_CREDDEF	Consolidation Private	STREAMS macros
allocb_tmpl	Evolving	Kernel function to allocate mblk from template
allocb_cred allocb_cred_wait	Consolidation Private	Kernel function to allocate mblk with cred
crgetuid crgetruid crgetsuid crgetgid crgetrgid crgetsgid crgetgroups crgetngroups	Evolving	Kernel functions to access cred fields
crgetref	Consolidation Private	Kernel function to get cred ref count
crsetuid crsetruid crsetsuid crsetgid crsetrgid crsetsgid crsetgroups crsetngroups	Consolidation Private	Kernel functions to set cred fields
mblk_setcred	Consolidation Private	Kernel function to set cred in mblk
priv_getbyname	Evolving	Kernel function to lookup privilege names

priv_policy priv_policy_only priv_policy_choice	Evolving	Kernel functions to report privileges
secpolicy functions	Consolidation Private	Kernel functions to check security policies
priv_debug	Unstable	/etc/system variable

4. Opinion

4.1 Device Privileges

A substantial amount of time was spent discussing how the introduction of privileges should affect the access model for devices. Previously, access to privileged operations through devices was restricted by the permission bits on the device node. The introduction of privileges makes this approach seem inadequate, since it continues to associate the ability to perform privileged operations with specific user and group ids rather than with a process-specific privileges.

The project team proposed adding options to add_drv and update_drv to specify device policies, allowing administrators to define the privileges required to access a particular device. This restriction is in addition to the device node permissions. The committee had some concerns about the lack of visibility of the privilege restrictions (particularly the fact that they are not reflected in the device node itself), the completeness of such an approach, the inability to assign distinct privileges to separate device instances with the same minor number, and the lack of clear direction regarding when to use privileges instead of (or in addition to) file permissions. Although this issue prompted some extended discussion during and after the review, in the end the committee accepted the proposal with the addition of a "getdevpolicy" utility to retrieve such information on a per-device (as well as system-wide) basis. See also the steering committee advice in Section 6 regarding future work in this area.

The committee was also concerned about the possibility that the change in the permissions model (the imposition of additional restrictions beyond those indicated by the file permissions) would surprise and confuse administrators. This resulted in the TCR listed in Appendix A, requiring the project team to discuss this change in the Solaris release notes as well as other appropriate documentation such as the Writing Device Drivers guide. The documentation should include advice to driver developers regarding when the use of privileges is appropriate.

4.2 Privilege Escalation Prevention

The project proposes measures to prevent "privilege escalation", where a process with a subset of privileges is able to gain additional privileges by exercising its privileges. While the project team identified and addressed certain scenarios that allow such escalation, some committee members were concerned that this might not be a comprehensive list. The project team responded that they had attempted to discover all such cases, but that due to the complex interactions in the Solaris kernel others might be possible. As with other security issues, these problems would be fixed when identified. The committee accepted this rationale.

4.3 Distinction between Privileges and Authorizations

The similarities between privileges and the authorizations introduced by the Role Based Access Control project [2] prompted some discussion. The project team explained that privileges represent basic system privileges enforced by the kernel, while authorizations represent higher-level capabilities defined by applications.

4.4 NFS Support for Privileges

This project introduces several file access control privileges, which allow non-root processes to override file ownership and mode settings. Although these privileges will work for files in local file systems, they will not work for files mounted over NFS since NFS has no mechanism for passing privilege information over the wire. This is particularly a concern for diskless client systems, where all file systems are mounted over NFS, and privileged access is often required. Since adding such support would require extensions to the NFS protocol to support passing privilege information, the committee agreed that it was outside the scope of this project, but suggested that the issue be considered in the context of future enhancements to NFS. It was noted that the lack of support for fine grained privileges in NFS does not represent a regression from the previous situation, but is a potentially useful area to explore in the context of NFS security. Section 6 contains advice regarding this issue.

5. Minority Opinion(s)

None.

6. Advisory Information

The Solaris PAC is advised to fund a project to improve the visibility of privileges required for device access, possibly through the use of extended attributes [3].

The Solaris PAC (or members therein with engineering management responsibilities) is advised to charter a group to investigate how to address the use of fine-grained privileges within NFS.

Appendices

Appendix A: Technical Changes Required

1. Add notice of the change to the device permissions model to the Solaris release notes.

Appendix B: Technical Changes Advised

None.

Appendix C: Reference Material

Unless otherwise stated, path names are relative to the case directory (PSARC/2002/188).

1. final.materials/priv.pdf
   Design document
2. ../../1997/332/opinion.ps
   Execution Profiles for Restricted Environments
3. ../../1999/209/opinion.ps
   Extended File Attributes