Subject: [PATCH 2.6.24-rc8-mm1 07/15] IPC: introduce ipc_update_perm()
Posted by Pierre Peiffer on Tue, 29 Jan 2008 16:02:36 GMT
View Forum Message <> Reply to Message

From: Pierre Peiffer <pierre.peiffer@bull.net>

The IPC_SET command performs the same permission setting for all IPCs.
This patch introduces a common ipc_update_perm() function to update these
permissions and makes use of it for all IPCs.

Signed-off-by: Pierre Peiffer <pierre.peiffer@bull.net>
Acked-by: Serge Hallyn <serue@us.ibm.com>

```
 ipc/msg.c  |    5 +----
 ipc/sem.c  |    5 +----
 ipc/shm.c  |    5 +----
 ipc/util.c |   13 ++++++++++++
 ipc/util.h |    1 +
 5 files changed, 17 insertions(+), 12 deletions(-)

Index: b/ipc/msg.c
===================================================================
--- a/ipc/msg.c
+++ b/ipc/msg.c
@@ -447,10 +447,7 @@ static int msgctl_down(struct ipc_namesp

   msq->q_qbytes = msqid64.msg_qbytes;

-  ipcp->uid = msqid64.msg_perm.uid;
-  ipcp->gid = msqid64.msg_perm.gid;
-  ipcp->mode = (ipcp->mode & ~S_IRWXUGO) |
-       (S_IRWXUGO & msqid64.msg_perm.mode);
+  ipc_update_perm(&msqid64.msg_perm, ipcp);
   msq->q_ctime = get_seconds();
   /* sleeping receivers might be excluded by
    * stricter permissions.
Index: b/ipc/sem.c
===================================================================
--- a/ipc/sem.c
+++ b/ipc/sem.c
@@ -915,10 +915,7 @@ static int semctl_down(struct ipc_namesp
   freeary(ns, ipcp);
   goto out_up;
  case IPC_SET:
-  ipcp->uid = semid64.sem_perm.uid;
-  ipcp->gid = semid64.sem_perm.gid;
-  ipcp->mode = (ipcp->mode & ~S_IRWXUGO)
```

```
-    | (semid64.sem_perm.mode & S_IRWXUGO);
+  ipc_update_perm(&semid64.sem_perm, ipcp);
   sma->sem_ctime = get_seconds();
   break;
  default:
Index: b/ipc/shm.c
===================================================================
--- a/ipc/shm.c
+++ b/ipc/shm.c
@@ -665,10 +665,7 @@ static int shmctl_down(struct ipc_namesp
   do_shm_rmid(ns, ipcp);
   goto out_up;
  case IPC_SET:
-  ipcp->uid = shmid64.shm_perm.uid;
-  ipcp->gid = shmid64.shm_perm.gid;
-  ipcp->mode = (ipcp->mode & ~S_IRWXUGO)
-    | (shmid64.shm_perm.mode & S_IRWXUGO);
+  ipc_update_perm(&shmid64.shm_perm, ipcp);
   shp->shm_ctim = get_seconds();
   break;
  default:
Index: b/ipc/util.c
===================================================================
--- a/ipc/util.c
+++ b/ipc/util.c
@@ -761,6 +761,19 @@ int ipcget(struct ipc_namespace *ns, str
   return ipcget_public(ns, ids, ops, params);
 }

+/**
+ * ipc_update_perm - update the permissions of an IPC.
+ * @in:  the permission given as input.
+ * @out: the permission of the ipc to set.
+ */
+void ipc_update_perm(struct ipc64_perm *in, struct kern_ipc_perm *out)
+{
+ out->uid = in->uid;
+ out->gid = in->gid;
+ out->mode = (out->mode & ~S_IRWXUGO)
+   | (in->mode & S_IRWXUGO);
+}
+
 #ifdef __ARCH_WANT_IPC_PARSE_VERSION


Index: b/ipc/util.h
===================================================================
--- a/ipc/util.h
```

```
+++ b/ipc/util.h
@@ -112,6 +112,7 @@ struct kern_ipc_perm *ipc_lock(struct ip

 void kernel_to_ipc64_perm(struct kern_ipc_perm *in, struct ipc64_perm *out);
 void ipc64_perm_to_ipc_perm(struct ipc64_perm *in, struct ipc_perm *out);
+void ipc_update_perm(struct ipc64_perm *in, struct kern_ipc_perm *out);

 #if defined(__ia64__) || defined(__x86_64__) || defined(__hppa__) || defined(__XTENSA__)
   /* On IA-64, we always use the "64-bit version" of the IPC structures.  */
```

--
Pierre Peiffer

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers