

---

Subject: [PATCH v4 4/9] ipc: add new SHM\_SET command for sys\_shmctl() call  
Posted by [Stanislav Kinsbursky](#) on Mon, 13 Aug 2012 12:31:54 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

New SHM\_SET command will be interpreted exactly as IPC\_SET, but also will update key, cuid and cgid values. IOW, it allows to change existent key value. The fact, that key is not used is checked before update. Otherwise -EEXIST is returned.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>  
Signed-off-by: Cyrill Gorcunov <gorcunov@openvz.org>

```
---  
include/linux/shm.h      | 1 +  
ipc/compat.c            | 1 +  
ipc/shm.c                | 13 ++++++++  
security/selinux/hooks.c | 1 +  
security/smack/smack_lsm.c | 1 +  
5 files changed, 15 insertions(+), 2 deletions(-)
```

```
diff --git a/include/linux/shm.h b/include/linux/shm.h  
index 92808b8..6fc5447 100644
```

```
--- a/include/linux/shm.h  
+++ b/include/linux/shm.h  
@@ -63,6 +63,7 @@ struct shmctl {  
/* ipcctl commands */  
#define SHM_STAT 13  
#define SHM_INFO 14  
+#define SHM_SET 15
```

```
/* Obsolete, used only for backwards compatibility */  
struct shminfo {
```

```
diff --git a/ipc/compat.c b/ipc/compat.c  
index aa88515..e7d4af7 100644
```

```
--- a/ipc/compat.c  
+++ b/ipc/compat.c  
@@ -688,6 +688,7 @@ long compat_sys_shmctl(int first, int second, void __user *uptr)
```

```
case IPC_SET:  
+ case SHM_SET:  
    if (version == IPC_64) {  
        err = get_compat_shmid64_ds(&s64, uptr);  
    } else {
```

```
diff --git a/ipc/shm.c b/ipc/shm.c  
index d4c0a9e..677b9f6 100644
```

```
--- a/ipc/shm.c  
+++ b/ipc/shm.c  
@@ -635,6 +635,9 @@ copy_shmid_from_user(struct shmctl64_ds *out, void __user *buf, int
```

```

version)
    out->shm_perm.uid = tbuf_old.shm_perm.uid;
    out->shm_perm.gid = tbuf_old.shm_perm.gid;
    out->shm_perm.mode = tbuf_old.shm_perm.mode;
+ out->shm_perm.cuid = tbuf_old.shm_perm.cuid;
+ out->shm_perm.cgid = tbuf_old.shm_perm.cgid;
+ out->shm_perm.key = tbuf_old.shm_perm.key;

    return 0;
}
@@ -739,12 +742,13 @@ static int shmctl_down(struct ipc_namespace *ns, int shmid, int cmd,
    struct shmctl_kernel *shp;
    int err;

- if (cmd == IPC_SET) {
+ if (cmd == IPC_SET || cmd == SHM_SET) {
    if (copy_shmid_from_user(&shmid64, buf, version))
        return -EFAULT;
}

- ipc = ipcctl_pre_down(ns, &shm_ids(ns), shmid, cmd,
+ ipc = ipcctl_pre_down(ns, &shm_ids(ns), shmid,
+ (cmd != SHM_SET) ? : IPC_SET,
    &shmctl64.shm_perm, 0);
    if (IS_ERR(ipc))
        return PTR_ERR(ipc);
@@ -758,6 +762,10 @@ static int shmctl_down(struct ipc_namespace *ns, int shmid, int cmd,
    case IPC_RMID:
        do_shm_rmid(ns, ipc);
        goto out_up;
+ case SHM_SET:
+ err = ipc_update_key(&shm_ids(ns), &shmctl64.shm_perm, ipc);
+ if (err)
+ break;
    case IPC_SET:
        ipc_update_perm(&shmctl64.shm_perm, ipc);
        shp->shm_ctim = get_seconds();
@@ -935,6 +943,7 @@ SYSCALL_DEFINE3(shmctl, int, shmid, int, cmd, struct shmctl_kernel
*, buf)
}
    case IPC_RMID:
    case IPC_SET:
+ case SHM_SET:
    err = shmctl_down(ns, shmid, cmd, buf, version);
    return err;
    default:
diff --git a/security/selinux/hooks.c b/security/selinux/hooks.c
index 6c56ebf..9e07e22 100644

```

```
--- a/security/selinux/hooks.c
+++ b/security/selinux/hooks.c
@@ -5054,6 +5054,7 @@ static int selinux_shm_shmctl(struct shmid_kernel *shp, int cmd)
    perms = SHM__GETATTR | SHM__ASSOCIATE;
    break;
    case IPC_SET:
+ case SHM_SET:
    perms = SHM__SETATTR;
    break;
    case SHM_LOCK:
diff --git a/security/smack/smack_lsm.c b/security/smack/smack_lsm.c
index ee0bb57..a25dfd1 100644
--- a/security/smack/smack_lsm.c
+++ b/security/smack/smack_lsm.c
@@ -2145,6 +2145,7 @@ static int smack_shm_shmctl(struct shmid_kernel *shp, int cmd)
    may = MAY_READ;
    break;
    case IPC_SET:
+ case SHM_SET:
    case SHM_LOCK:
    case SHM_UNLOCK:
    case IPC_RMID:
```

---