

---

Subject: [PATCH v4 6/9] ipc: add new SEM\_SET command for sys\_semctl() call  
Posted by [Stanislav Kinsbursky](#) on Mon, 13 Aug 2012 12:32:05 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

New SEM\_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](mailto:skinsbursky@parallels.com)>

Signed-off-by: Cyrill Gorcunov <[gorcunov@openvz.org](mailto:gorcunov@openvz.org)>

---

```
include/linux/sem.h      | 1 +
ipc/compat.c             | 1 +
ipc/sem.c                | 10 ++++++++--
security/selinux/hooks.c | 1 +
security/smack/smack_lsm.c | 1 +
5 files changed, 12 insertions(+), 2 deletions(-)
```

diff --git a/include/linux/sem.h b/include/linux/sem.h

index 10d6b22..c74b9b5 100644

--- a/include/linux/sem.h

+++ b/include/linux/sem.h

@@ -18,6 +18,7 @@

/\* ipcctl cmds \*/

#define SEM\_STAT 18

#define SEM\_INFO 19

+#define SEM\_SET 20

/\* Obsolete, used only for backwards compatibility and libc5 compiles \*/

struct semid\_ds {

diff --git a/ipc/compat.c b/ipc/compat.c

index da4f5c6..7cfdb5b 100644

--- a/ipc/compat.c

+++ b/ipc/compat.c

@@ -290,6 +290,7 @@ static long do\_compat\_semctl(int first, int second, int third, u32 pad)  
break;

case IPC\_SET:

+ case SEM\_SET:

if (version == IPC\_64) {  
err = get\_compat\_sem64\_ds(&s64, compat\_ptr(pad));  
} else {

diff --git a/ipc/sem.c b/ipc/sem.c

index 845c912..d9024d5 100644

--- a/ipc/sem.c

+++ b/ipc/sem.c

@@ -1084,12 +1084,13 @@ static int semctl\_down(struct ipc\_namespace \*ns, int semid,

```

struct semid64_ds semid64;
struct kern_ipc_perm *ipcp;

- if(cmd == IPC_SET) {
+ if (cmd == IPC_SET || cmd == SEM_SET) {
    if (copy_semids_from_user(&semid64, arg.buf, version))
        return -EFAULT;
    }

- ipcp = ipcctl_pre_down(ns, &sem_ids(ns), semid, cmd,
+ ipcp = ipcctl_pre_down(ns, &sem_ids(ns), semid,
+ (cmd != SEM_SET) ? : IPC_SET,
    &semid64.sem_perm, 0);
    if (IS_ERR(ipcp))
        return PTR_ERR(ipcp);
@@ -1104,6 +1105,10 @@ static int semctl_down(struct ipc_namespace *ns, int semid,
    case IPC_RMID:
        freeary(ns, ipcp);
        goto out_up;
+ case SEM_SET:
+ err = ipc_update_key(&sem_ids(ns), &semid64.sem_perm, ipcp);
+ if (err)
+ break;
    case IPC_SET:
        ipc_update_perm(&semid64.sem_perm, ipcp);
        sma->sem_ctime = get_seconds();
@@ -1149,6 +1154,7 @@ SYSCALL_DEFINE(semctl)(int semid, int semnum, int cmd, union
semun arg)
    return err;
    case IPC_RMID:
    case IPC_SET:
+ case SEM_SET:
    err = semctl_down(ns, semid, cmd, version, arg);
    return err;
    default:
diff --git a/security/selinux/hooks.c b/security/selinux/hooks.c
index 622e020..704391b 100644
--- a/security/selinux/hooks.c
+++ b/security/selinux/hooks.c
@@ -5160,6 +5160,7 @@ static int selinux_sem_semctl(struct sem_array *sma, int cmd)
    perms = SEM_DESTROY;
    break;
    case IPC_SET:
+ case SEM_SET:
    perms = SEM_SETATTR;
    break;
    case IPC_STAT:
diff --git a/security/smack/smack_lsm.c b/security/smack/smack_lsm.c

```

```
index 62826b1..1116076 100644
--- a/security/smack/smack_lsm.c
+++ b/security/smack/smack_lsm.c
@@ -2277,6 +2277,7 @@ static int smack_sem_semctl(struct sem_array *sma, int cmd)
     case SETALL:
     case IPC_RMID:
     case IPC_SET:
+ case SEM_SET:
     may = MAY_READWRITE;
     break;
     case IPC_INFO:
```

---