
Subject: [PATCH v6 05/10] ipc: add new MSG_SET command for sys_msgctl() call
Posted by [Stanislav Kinsbursky](#) on Mon, 15 Oct 2012 10:30:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

New MSG_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>

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

diff --git a/include/uapi/linux/msg.h b/include/uapi/linux/msg.h

index 78dbd2f..76999c9 100644

--- a/include/uapi/linux/msg.h

+++ b/include/uapi/linux/msg.h

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

/* ipcctl commands */

#define MSG_STAT 11

#define MSG_INFO 12

+#define MSG_SET 13

/* msgrcv options */

#define MSG_NOERROR 010000 /* no error if message is too big */

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

index 35c750d..9c70f9a 100644

--- a/ipc/compat.c

+++ b/ipc/compat.c

@ @ -483,6 +483,7 @ @ long compat_sys_msgctl(int first, int second, void __user *uptr)
break;

case IPC_SET:

+ case MSG_SET:

if (version == IPC_64) {
err = get_compat_msqid64(&m64, uptr);
} else {

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

index 2f44946..68515dc 100644

--- a/ipc/msg.c

+++ b/ipc/msg.c

@ @ -392,6 +392,9 @ @ copy_msqid_from_user(struct msqid64_ds *out, void __user *buf, int
version)

```

    out->msg_perm.uid      = tbuf_old.msg_perm.uid;
    out->msg_perm.gid      = tbuf_old.msg_perm.gid;
    out->msg_perm.mode     = tbuf_old.msg_perm.mode;
+ out->msg_perm.cuid = tbuf_old.msg_perm.cuid;
+ out->msg_perm.cgid = tbuf_old.msg_perm.cgid;
+ out->msg_perm.key = tbuf_old.msg_perm.key;

    if (tbuf_old.msg_qbytes == 0)
        out->msg_qbytes = tbuf_old.msg_lqbytes;
@@ -418,12 +421,13 @@ static int msgctl_down(struct ipc_namespace *ns, int msqid, int cmd,
    struct msg_queue *msq;
    int err;

- if (cmd == IPC_SET) {
+ if (cmd == IPC_SET || cmd == MSG_SET) {
    if (copy_msqid_from_user(&msqid64, buf, version))
        return -EFAULT;
    }

- ipcp = ipcctl_pre_down(ns, &msg_ids(ns), msqid, cmd,
+ ipcp = ipcctl_pre_down(ns, &msg_ids(ns), msqid,
+ (cmd != MSG_SET) ? cmd : IPC_SET,
+ &msqid64.msg_perm, msqid64.msg_qbytes);
    if (IS_ERR(ipcp))
        return PTR_ERR(ipcp);
@@ -439,6 +443,7 @@ static int msgctl_down(struct ipc_namespace *ns, int msqid, int cmd,
    freeque(ns, ipcp);
    goto out_up;
    case IPC_SET:
+ case MSG_SET:
    if (msqid64.msg_qbytes > ns->msg_ctlmnbs &&
        !capable(CAP_SYS_RESOURCE)) {
        err = -EPERM;
@@ -451,6 +456,9 @@ static int msgctl_down(struct ipc_namespace *ns, int msqid, int cmd,

    msq->q_qbytes = msqid64.msg_qbytes;

+ if (cmd == MSG_SET)
+ ipc_update_key(&msg_ids(ns), &msqid64.msg_perm, ipcp);
+
    msq->q_ctime = get_seconds();
    /* sleeping receivers might be excluded by
     * stricter permissions.
@@ -569,6 +577,7 @@ SYSCALL_DEFINE3(msgctl, int, msqid, int, cmd, struct msqid_ds __user
    *, buf)
    }
    case IPC_SET:
    case IPC_RMID:

```

```

+ case MSG_SET:
    err = msgctl_down(ns, msqid, cmd, buf, version);
    return err;
default:
diff --git a/security/selinux/hooks.c b/security/selinux/hooks.c
index 62b2447..78b77ac 100644
--- a/security/selinux/hooks.c
+++ b/security/selinux/hooks.c
@@ -4885,6 +4885,7 @@ static int selinux_msg_queue_msgctl(struct msg_queue *msq, int cmd)
    perms = MSGQ__GETATTR | MSGQ__ASSOCIATE;
    break;
case IPC_SET:
+ case MSG_SET:
    perms = MSGQ__SETATTR;
    break;
case IPC_RMID:
diff --git a/security/smack/smack_lsm.c b/security/smack/smack_lsm.c
index c7eabc9..d51a8da 100644
--- a/security/smack/smack_lsm.c
+++ b/security/smack/smack_lsm.c
@@ -2374,6 +2374,7 @@ static int smack_msg_queue_msgctl(struct msg_queue *msq, int cmd)
    may = MAY_READ;
    break;
case IPC_SET:
+ case MSG_SET:
case IPC_RMID:
    may = MAY_READWRITE;
    break;

```
