

---

Subject: [PATCH v6 07/10] ipc: add new SEM\_SET command for sys\_semctl() call  
Posted by [Stanislav Kinsbursky](#) on Mon, 15 Oct 2012 16:00:12 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>

```
---  
include/uapi/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/uapi/linux/sem.h b/include/uapi/linux/sem.h
```

```
index 541fce0..b6ae374 100644
```

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

```
+++ b/include/uapi/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 9c70f9a..84d8efd 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 10e9085..3eac885 100644
```

```
--- a/ipc/sem.c
```

```
+++ b/ipc/sem.c
```

```
@@ -1085,12 +1085,13 @@ static int semctl_down(struct ipc_namespace *ns, int semid,  
    struct sem64_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) ? cmd : IPC_SET,
    &semid64.sem_perm, 0);
    if (IS_ERR(ipcp))
        return PTR_ERR(ipcp);
@@ -1105,6 +1106,10 @@ 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:
        err = ipc_update_perm(&semid64.sem_perm, ipcp);
        if (err)
@@ -1152,6 +1157,7 @@ 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 78b77ac..02b037d 100644
--- a/security/selinux/hooks.c
+++ b/security/selinux/hooks.c
@@ -5133,6 +5133,7 @@ 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 d51a8da..b4135ed 100644

```

```
--- a/security/smack/smack_lsm.c
+++ b/security/smack/smack_lsm.c
@@ -2253,6 +2253,7 @@ 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:
```

---

Subject: Re: [PATCH v6 07/10] ipc: add new SEM\_SET command for sys\_semctl() call

Posted by [Serge E. Hallyn](#) on Tue, 23 Oct 2012 16:34:04 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Stanislav Kinsbursky (skinsbursky@parallels.com):

> 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>

Acked-by: Serge E. Hallyn <serge.hallyn@ubuntu.com>

(again, modulo requested /\* fallthrough \*/ comment )

> ---

```
> include/uapi/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/uapi/linux/sem.h b/include/uapi/linux/sem.h

> index 541fce0..b6ae374 100644

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

> +++ b/include/uapi/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 9c70f9a..84d8efd 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_semctl64_ds(&s64, compat_ptr(pad));
> } else {
> diff --git a/ipc/sem.c b/ipc/sem.c
> index 10e9085..3eac885 100644
> --- a/ipc/sem.c
> +++ b/ipc/sem.c
> @@ -1085,12 +1085,13 @@ 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_semctl_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) ? cmd : IPC_SET,
> &semid64.sem_perm, 0);
> if (IS_ERR(ipcp))
> return PTR_ERR(ipcp);
> @@ -1105,6 +1106,10 @@ 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:
> err = ipc_update_perm(&semid64.sem_perm, ipcp);
> if (err)
> @@ -1152,6 +1157,7 @@ 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 78b77ac..02b037d 100644
> --- a/security/selinux/hooks.c
> +++ b/security/selinux/hooks.c
> @@ -5133,6 +5133,7 @@ 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 d51a8da..b4135ed 100644
> --- a/security/smack/smack_lsm.c
> +++ b/security/smack/smack_lsm.c
> @@ -2253,6 +2253,7 @@ 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:
>
> --
> To unsubscribe from this list: send the line "unsubscribe linux-security-module" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html
```

---