

---

Subject: [PATCH -mm 2/2] user namespace : add unshare  
Posted by [Cedric Le Goater](#) on Fri, 08 Jun 2007 15:14:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: Serge E. Hallyn <serue@us.ibm.com>

Changelog: Fix !CONFIG\_USER\_NS clone with CLONE\_NEWUSER so it returns -EINVAL rather than 0, so that userspace knows they didn't get a new user namespace.

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

Signed-off-by: Andrew Morton <akpm@osdl.org>

Acked-by: Pavel Emelianov <xemul@openvz.org>

Cc: Herbert Poetzl <herbert@13thfloor.at>

Cc: Kirill Korotaev <dev@sw.ru>

Cc: Eric W. Biederman <ebiederm@xmission.com>

---

```
include/linux/sched.h      | 1
include/linux/user_namespace.h | 4 +++
kernel/fork.c              | 2 -
kernel/nsproxy.c           | 5 ++--
kernel/user_namespace.c     | 46 ++++++-----
5 files changed, 54 insertions(+), 4 deletions(-)
```

Index: 2.6.22-rc4-mm2/include/linux/sched.h

```
=====
--- 2.6.22-rc4-mm2.orig/include/linux/sched.h
+++ 2.6.22-rc4-mm2/include/linux/sched.h
@@ -25,6 +25,7 @@
#define CLONE_STOPPED 0x02000000 /* Start in stopped state */
#define CLONE_NEWUTS 0x04000000 /* New utsname group? */
#define CLONE_NEWIPC 0x08000000 /* New ipc */
+#define CLONE_NEWUSER 0x10000000 /* New user namespace */

/*
 * Scheduling policies
```

Index: 2.6.22-rc4-mm2/include/linux/user\_namespace.h

```
=====
--- 2.6.22-rc4-mm2.orig/include/linux/user_namespace.h
+++ 2.6.22-rc4-mm2/include/linux/user_namespace.h
@@ -4,6 +4,7 @@
#include <linux/kref.h>
#include <linux/nsproxy.h>
#include <linux/sched.h>
+#include <linux/err.h>
```

```

#define UIDHASH_BITS (CONFIG_BASE_SMALL ? 3 : 8)
#define UIDHASH_SZ (1 << UIDHASH_BITS)
@@ -45,6 +46,9 @@ static inline struct user_namespace *get
static inline struct user_namespace *copy_user_ns(int flags,
    struct user_namespace *old_ns)
{
+ if (flags & CLONE_NEWUSER)
+ return ERR_PTR(-EINVAL);
+
return NULL;
}

```

Index: 2.6.22-rc4-mm2/kernel/fork.c

```

=====
--- 2.6.22-rc4-mm2.orig/kernel/fork.c
+++ 2.6.22-rc4-mm2/kernel/fork.c
@@ -1616,7 +1616,7 @@ asmlinkage long sys_unshare(unsigned lon
err = -EINVAL;
if (unshare_flags & ~(CLONE_THREAD|CLONE_FS|CLONE_NEWNS|CLONE_SIGHAND|
    CLONE_VM|CLONE_FILES|CLONE_SYSVSEM|
- CLONE_NEWUTS|CLONE_NEWIPC))
+ CLONE_NEWUTS|CLONE_NEWIPC|CLONE_NEWUSER))
goto bad_unshare_out;

if ((err = unshare_thread(unshare_flags)))

```

Index: 2.6.22-rc4-mm2/kernel/nsproxy.c

```

=====
--- 2.6.22-rc4-mm2.orig/kernel/nsproxy.c
+++ 2.6.22-rc4-mm2/kernel/nsproxy.c
@@ -117,7 +117,7 @@ int copy_namespaces(int flags, struct ta

get_nsproxy(old_ns);

- if (!(flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC)))
+ if (!(flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC | CLONE_NEWUSER)))
return 0;

if (!capable(CAP_SYS_ADMIN)) {
@@ -163,7 +163,8 @@ int unshare_nsproxy_namespaces(unsigned
struct nsproxy *old_ns = current->nsproxy;
int err = 0;

- if (!(unshare_flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC)))
+ if (!(unshare_flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC |
+ CLONE_NEWUSER)))
return 0;

if (!capable(CAP_SYS_ADMIN))

```

Index: 2.6.22-rc4-mm2/kernel/user\_namespace.c

=====

--- 2.6.22-rc4-mm2.orig/kernel/user\_namespace.c

+++ 2.6.22-rc4-mm2/kernel/user\_namespace.c

@@ -21,6 +21,45 @@ EXPORT\_SYMBOL\_GPL(init\_user\_ns);

```
#ifdef CONFIG_USER_NS
```

```
  /*
```

```
  * Clone a new ns copying an original user ns, setting refcount to 1
```

```
  * @old_ns: namespace to clone
```

```
  * Return NULL on error (failure to kcalloc), new ns otherwise
```

```
  */
```

```
+static struct user_namespace *clone_user_ns(struct user_namespace *old_ns)
```

```
+{
```

```
+ struct user_namespace *ns;
```

```
+ struct user_struct *new_user;
```

```
+ int n;
```

```
+
```

```
+ ns = kcalloc(sizeof(struct user_namespace), GFP_KERNEL);
```

```
+ if (!ns)
```

```
+ return NULL;
```

```
+
```

```
+ kref_init(&ns->kref);
```

```
+
```

```
+ for (n = 0; n < UIDHASH_SZ; ++n)
```

```
+ INIT_LIST_HEAD(ns->uidhash_table + n);
```

```
+
```

```
+ /* Insert new root user. */
```

```
+ ns->root_user = alloc_uid(ns, 0);
```

```
+ if (!ns->root_user) {
```

```
+ kfree(ns);
```

```
+ return NULL;
```

```
+ }
```

```
+
```

```
+ /* Reset current->user with a new one */
```

```
+ new_user = alloc_uid(ns, current->uid);
```

```
+ if (!new_user) {
```

```
+ free_uid(ns->root_user);
```

```
+ kfree(ns);
```

```
+ return NULL;
```

```
+ }
```

```
+
```

```
+ switch_uid(new_user);
```

```
+ return ns;
```

```
+
```

```
+
```

```
 struct user_namespace * copy_user_ns(int flags, struct user_namespace *old_ns)
```

```
{
    struct user_namespace *new_ns;
@@ -28,7 +67,12 @@ struct user_namespace * copy_user_ns(int
    BUG_ON(!old_ns);
    get_user_ns(old_ns);

- new_ns = old_ns;
+ if (!(flags & CLONE_NEWUSER))
+     return old_ns;
+
+ new_ns = clone_user_ns(old_ns);
+
+ put_user_ns(old_ns);
    return new_ns;
}
```

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---