Subject: [PATCH 1/4] userns: let clone_uts_ns() handle setting uts->user_ns
Posted by serge on Mon, 21 Feb 2011 04:01:57 GMT
View Forum Message <> Reply to Message

To do so we need to pass in the task_struct who'll get the utsname,
so we can get its user_ns.

Signed-off-by: Serge E. Hallyn <serge.hallyn@canonical.com>
---
 include/linux/utsname.h | 10 ++++++----
 kernel/nsproxy.c | 7 +------
 kernel/utsname.c | 12 +++++++-----
 3 files changed, 14 insertions(+), 15 deletions(-)

diff --git a/include/linux/utsname.h b/include/linux/utsname.h
index 85171be..165b17b 100644
--- a/include/linux/utsname.h
+++ b/include/linux/utsname.h
@@ -52,8 +52,9 @@ static inline void get_uts_ns(struct uts_namespace *ns)
  kref_get(&ns->kref);
 }

-extern struct uts_namespace *copy_utsname(unsigned long flags,
-     struct uts_namespace *ns);
+extern struct uts_namespace *copy_utsname(struct task_struct *tsk,
+   unsigned long flags,
+   struct uts_namespace *ns);
 extern void free_uts_ns(struct kref *kref);

 static inline void put_uts_ns(struct uts_namespace *ns)
@@ -69,8 +70,9 @@ static inline void put_uts_ns(struct uts_namespace *ns)
 {
 }

-static inline struct uts_namespace *copy_utsname(unsigned long flags,
-     struct uts_namespace *ns)
+static inline struct uts_namespace *copy_utsname(struct task_struct *tsk,
+   unsigned long flags,
+   struct uts_namespace *ns)
 {
  if (flags & CLONE_NEWUTS)
   return ERR_PTR(-EINVAL);
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index b6dbff2..ffa6b67 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -69,16 +69,11 @@ static struct nsproxy *create_new_namespaces(unsigned long flags,
   goto out_ns;

```
  }

- new_nsp->uts_ns = copy_utsname(flags, tsk->nsproxy->uts_ns);
+ new_nsp->uts_ns = copy_utsname(tsk, flags, tsk->nsproxy->uts_ns);
  if (IS_ERR(new_nsp->uts_ns)) {
   err = PTR_ERR(new_nsp->uts_ns);
   goto out_uts;
  }
- if (new_nsp->uts_ns != tsk->nsproxy->uts_ns) {
-  put_user_ns(new_nsp->uts_ns->user_ns);
-  new_nsp->uts_ns->user_ns = task_cred_xxx(tsk, user)->user_ns;
-  get_user_ns(new_nsp->uts_ns->user_ns);
- }

  new_nsp->ipc_ns = copy_ipcs(flags, tsk->nsproxy->ipc_ns);
  if (IS_ERR(new_nsp->ipc_ns)) {
diff --git a/kernel/utsname.c b/kernel/utsname.c
index a7b3a8d..9462580 100644
--- a/kernel/utsname.c
+++ b/kernel/utsname.c
@@ -31,7 +31,8 @@ static struct uts_namespace *create_uts_ns(void)
  * @old_ns: namespace to clone
  * Return NULL on error (failure to kmalloc), new ns otherwise
  */
-static struct uts_namespace *clone_uts_ns(struct uts_namespace *old_ns)
+static struct uts_namespace *clone_uts_ns(struct task_struct *tsk,
+       struct uts_namespace *old_ns)
 {
  struct uts_namespace *ns;

@@ -41,8 +42,7 @@ static struct uts_namespace *clone_uts_ns(struct uts_namespace *old_ns)

  down_read(&uts_sem);
  memcpy(&ns->name, &old_ns->name, sizeof(ns->name));
- ns->user_ns = old_ns->user_ns;
- get_user_ns(ns->user_ns);
+ ns->user_ns = get_user_ns(task_cred_xxx(tsk, user)->user_ns);
  up_read(&uts_sem);
  return ns;
 }
@@ -53,7 +53,9 @@ static struct uts_namespace *clone_uts_ns(struct uts_namespace *old_ns)
  * utsname of this process won't be seen by parent, and vice
  * versa.
  */
-struct uts_namespace *copy_utsname(unsigned long flags, struct uts_namespace *old_ns)
+struct uts_namespace *copy_utsname(struct task_struct *tsk,
+       unsigned long flags,
+       struct uts_namespace *old_ns)
```

```
 {
  struct uts_namespace *new_ns;

@@ -63,7 +65,7 @@ struct uts_namespace *copy_utsname(unsigned long flags, struct
uts_namespace *ol
  if (!(flags & CLONE_NEWUTS))
   return old_ns;

- new_ns = clone_uts_ns(old_ns);
+ new_ns = clone_uts_ns(tsk, old_ns);

  put_uts_ns(old_ns);
  return new_ns;
--
1.7.0.4
```

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs