Subject: [PATCH] namespace: ensure clone_flags are always stored in an unsigned long
Posted by ebiederm on Fri, 22 Jun 2007 06:46:36 GMT
View Forum Message <> Reply to Message

While working on unshare support for the network namespace I noticed
we were putting clone flags in an int. Which is weird because the
syscall uses unsigned long and we at least need an unsigned to
properly hold all of the unshare flags.

So to make the code consistent, this patch updates the code to use
unsigned long instead of int for the clone flags in those places
where we get it wrong today.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
fs/namespace.c | 2 +-
include/linux/mnt_namespace.h | 2 +-
include/linux/nsproxy.h | 2 +-
include/linux/pid_namespace.h | 2 +-
include/linux/utsname.h | 4 ++--
kernel/nsproxy.c | 6 +++---
kernel/pid.c | 2 +-
kernel/utsname.c | 2 +-
8 files changed, 11 insertions(+), 11 deletions(-)

diff --git a/fs/namespace.c b/fs/namespace.c
index 82d4fd2..56d6e8d 100644
--- a/fs/namespace.c
+++ b/fs/namespace.c
@@ -1657,7 +1657,7 @@ static struct mnt_namespace *dup_mnt_ns(struct mnt_namespace
*mnt_ns,
return new_ns;
}

-struct mnt_namespace *copy_mnt_ns(int flags, struct mnt_namespace *ns,
+struct mnt_namespace *copy_mnt_ns(unsigned long flags, struct mnt_namespace *ns,
struct fs_struct *new_fs)
{
struct mnt_namespace *new_ns;
diff --git a/include/linux/mnt_namespace.h b/include/linux/mnt_namespace.h
index 1fa4d98..8eed44f 100644
--- a/include/linux/mnt_namespace.h
+++ b/include/linux/mnt_namespace.h
@@ -14,7 +14,7 @@ struct mnt_namespace {
int event;
};

```
-extern struct mnt_namespace *copy_mnt_ns(int, struct mnt_namespace *,
+extern struct mnt_namespace *copy_mnt_ns(unsigned long, struct mnt_namespace *,
   struct fs_struct *);
 extern void __put_mnt_ns(struct mnt_namespace *ns);

diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
index 189e0dc..4cc8a6f 100644
--- a/include/linux/nsproxy.h
+++ b/include/linux/nsproxy.h
@@ -31,7 +31,7 @@ struct nsproxy {
 };
 extern struct nsproxy init_nsproxy;

-int copy_namespaces(int flags, struct task_struct *tsk);
+int copy_namespaces(unsigned long flags, struct task_struct *tsk);
 void get_task_namespaces(struct task_struct *tsk);
 void free_nsproxy(struct nsproxy *ns);
 int unshare_nsproxy_namespaces(unsigned long, struct nsproxy **,
diff --git a/include/linux/pid_namespace.h b/include/linux/pid_namespace.h
index 169c6c2..b9a17e0 100644
--- a/include/linux/pid_namespace.h
+++ b/include/linux/pid_namespace.h
@@ -29,7 +29,7 @@ static inline void get_pid_ns(struct pid_namespace *ns)
  kref_get(&ns->kref);
 }

-extern struct pid_namespace *copy_pid_ns(int flags, struct pid_namespace *ns);
+extern struct pid_namespace *copy_pid_ns(unsigned long flags, struct pid_namespace *ns);
 extern void free_pid_ns(struct kref *kref);

 static inline void put_pid_ns(struct pid_namespace *ns)
diff --git a/include/linux/utsname.h b/include/linux/utsname.h
index f8d3b32..7dc303c 100644
--- a/include/linux/utsname.h
+++ b/include/linux/utsname.h
@@ -49,7 +49,7 @@ static inline void get_uts_ns(struct uts_namespace *ns)
 }

 #ifdef CONFIG_UTS_NS
-extern struct uts_namespace *copy_utsname(int flags, struct uts_namespace *ns);
+extern struct uts_namespace *copy_utsname(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)
@@ -57,7 +57,7 @@ static inline void put_uts_ns(struct uts_namespace *ns)
  kref_put(&ns->kref, free_uts_ns);
 }
 #else
```

```
-static inline struct uts_namespace *copy_utsname(int flags,
+static inline struct uts_namespace *copy_utsname(unsigned long flags,
       struct uts_namespace *ns)
 {
  return ns;
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index 1bc4b55..8737c64 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -54,8 +54,8 @@ static inline struct nsproxy *clone_nsproxy(struct nsproxy *orig)
  * Return the newly created nsproxy.  Do not attach this to the task,
  * leave it to the caller to do proper locking and attach it to task.
  */
-static struct nsproxy *create_new_namespaces(int flags, struct task_struct *tsk,
-   struct fs_struct *new_fs)
+static struct nsproxy *create_new_namespaces(unsigned long flags,
+   struct task_struct *tsk, struct fs_struct *new_fs)
 {
  struct nsproxy *new_nsp;

@@ -99,7 +99,7 @@ out_ns:
  * called from clone.  This now handles copy for nsproxy and all
  * namespaces therein.
  */
-int copy_namespaces(int flags, struct task_struct *tsk)
+int copy_namespaces(unsigned long flags, struct task_struct *tsk)
 {
  struct nsproxy *old_ns = tsk->nsproxy;
  struct nsproxy *new_ns;
diff --git a/kernel/pid.c b/kernel/pid.c
index eb66bd2..c6e3f9f 100644
--- a/kernel/pid.c
+++ b/kernel/pid.c
@@ -365,7 +365,7 @@ struct pid *find_ge_pid(int nr)
 }
 EXPORT_SYMBOL_GPL(find_get_pid);

-struct pid_namespace *copy_pid_ns(int flags, struct pid_namespace *old_ns)
+struct pid_namespace *copy_pid_ns(unsigned long flags, struct pid_namespace *old_ns)
 {
  BUG_ON(!old_ns);
  get_pid_ns(old_ns);
diff --git a/kernel/utsname.c b/kernel/utsname.c
index 160c8c5..8e5a8fe 100644
--- a/kernel/utsname.c
+++ b/kernel/utsname.c
@@ -37,7 +37,7 @@ 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(int flags, struct uts_namespace *old_ns)
+struct uts_namespace *copy_utsname(unsigned long flags, struct uts_namespace *old_ns)
 {
  struct uts_namespace *new_ns;

--
1.5.1.1.181.g2de0
```

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