Subject: [PATCH][RFC] Cleanup in namespaces unsharing
Posted by Pavel Emelianov on Fri, 08 Jun 2007 09:09:16 GMT
View Forum Message <> Reply to Message

Currently we have two funtions to copy the namespaces:
copy_namespaces() and unshare_nsproxy_namespaces(). The
second one checks for unsupported functionality with

```
#ifndef CONFIG_IPC_NS
 if (unshare_flags & CLONE_NEWIPC)
  return -EINVAL;
#endif
```

-like constructions, while the first one does not. One
of the side effects of this is that clone() with the
CLONE_NEWXXX set will return 0 if the kernel doesn't
support XXX namespaces thus confusing the user-level.

The proposal is to make these calls clean from the ifdefs
and move these checks into each namespaces' stubs. This
will make the code cleaner and (!) return -EINVAL from
fork() in case the desired namespaces are not supported.

Did I miss something in the design or this patch worth merging?

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

```
diff --git a/include/linux/ipc.h b/include/linux/ipc.h
index 7c8c6d8..b5aed71 100644
--- a/include/linux/ipc.h
+++ b/include/linux/ipc.h
@@ -100,6 +100,9 @@ extern struct ipc_namespace *copy_ipcs(u
 static inline struct ipc_namespace *copy_ipcs(unsigned long flags,
       struct ipc_namespace *ns)
 {
+ if (flags & CLONE_NEWIPC)
+  ns = ERR_PTR(-EINVAL);
+
  return ns;
 }
 #endif
diff --git a/include/linux/utsname.h b/include/linux/utsname.h
index f8d3b32..230706e 100644
--- a/include/linux/utsname.h
+++ b/include/linux/utsname.h
@@ -60,6 +60,9 @@ static inline void put_uts_ns(struct uts
```

```
 static inline struct uts_namespace *copy_utsname(int flags,
       struct uts_namespace *ns)
 {
+ if (flags & CLONE_NEWUTS)
+  ns = ERR_PTR(-EINVAL);
+
  return ns;
 }

diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index 1bc4b55..ef26615 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -157,16 +157,6 @@ int unshare_nsproxy_namespaces(unsigned
  if (!(unshare_flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC)))
   return 0;

-#ifndef CONFIG_IPC_NS
- if (unshare_flags & CLONE_NEWIPC)
-  return -EINVAL;
-#endif
-
-#ifndef CONFIG_UTS_NS
- if (unshare_flags & CLONE_NEWUTS)
-  return -EINVAL;
-#endif
-
  if (!capable(CAP_SYS_ADMIN))
   return -EPERM;
```

Subject: Re: [PATCH][RFC] Cleanup in namespaces unsharing
Posted by Cedric Le Goater on Fri, 08 Jun 2007 09:35:57 GMT
View Forum Message <> Reply to Message

Pavel Emelianov wrote:
> Currently we have two funtions to copy the namespaces:
> copy_namespaces() and unshare_nsproxy_namespaces(). The
> second one checks for unsupported functionality with
>
> #ifndef CONFIG_IPC_NS
> if (unshare_flags & CLONE_NEWIPC)
>   return -EINVAL;
> #endif
>
> -like constructions, while the first one does not. One
> of the side effects of this is that clone() with the
> CLONE_NEWXXX set will return 0 if the kernel doesn't

> support XXX namespaces thus confusing the user-level.
>
> The proposal is to make these calls clean from the ifdefs
> and move these checks into each namespaces' stubs. This
> will make the code cleaner and (!) return -EINVAL from
> fork() in case the desired namespaces are not supported.
>
> Did I miss something in the design or this patch worth merging?

I've sent a more brutal patch in the past removing CONFIG_IPC_NS
and CONFIG_UTS_NS. Might be a better idea ?

Let me refresh it and resend.

C.

Subject: Re: [PATCH][RFC] Cleanup in namespaces unsharing
Posted by Pavel Emelianov on Fri, 08 Jun 2007 11:23:08 GMT
View Forum Message <> Reply to Message

Cedric Le Goater wrote:
> Pavel Emelianov wrote:
>> Currently we have two funtions to copy the namespaces:
>> copy_namespaces() and unshare_nsproxy_namespaces(). The
>> second one checks for unsupported functionality with
>>
>> #ifndef CONFIG_IPC_NS
>> if (unshare_flags & CLONE_NEWIPC)
>>   return -EINVAL;
>> #endif
>>
>> -like constructions, while the first one does not. One
>> of the side effects of this is that clone() with the
>> CLONE_NEWXXX set will return 0 if the kernel doesn't
>> support XXX namespaces thus confusing the user-level.
>>
>> The proposal is to make these calls clean from the ifdefs
>> and move these checks into each namespaces' stubs. This
>> will make the code cleaner and (!) return -EINVAL from
>> fork() in case the desired namespaces are not supported.
>>
>> Did I miss something in the design or this patch worth merging?
>
> I've sent a more brutal patch in the past removing CONFIG_IPC_NS
> and CONFIG_UTS_NS. Might be a better idea ?

In case namespaces do not produce performance loss - yes.

By that patch I also wanted to note that we'd better make
all the other namespaces check for flags themselves, not
putting this in the generic code.

> Let me refresh it and resend.
>
> C.
>

Subject: Re: [PATCH][RFC] Cleanup in namespaces unsharing
Posted by Cedric Le Goater on Fri, 08 Jun 2007 12:01:32 GMT
View Forum Message <> Reply to Message

Pavel Emelianov wrote:
> Cedric Le Goater wrote:
>> Pavel Emelianov wrote:
>>> Currently we have two funtions to copy the namespaces:
>>> copy_namespaces() and unshare_nsproxy_namespaces(). The
>>> second one checks for unsupported functionality with
>>>
>>> #ifndef CONFIG_IPC_NS
>>> if (unshare_flags & CLONE_NEWIPC)
>>>   return -EINVAL;
>>> #endif
>>>
>>> -like constructions, while the first one does not. One
>>> of the side effects of this is that clone() with the
>>> CLONE_NEWXXX set will return 0 if the kernel doesn't
>>> support XXX namespaces thus confusing the user-level.
>>>
>>> The proposal is to make these calls clean from the ifdefs
>>> and move these checks into each namespaces' stubs. This
>>> will make the code cleaner and (!) return -EINVAL from
>>> fork() in case the desired namespaces are not supported.
>>>
>>> Did I miss something in the design or this patch worth merging?
>> I've sent a more brutal patch in the past removing CONFIG_IPC_NS
>> and CONFIG_UTS_NS. Might be a better idea ?
>
> In case namespaces do not produce performance loss - yes.
>
> By that patch I also wanted to note that we'd better make
> all the other namespaces check for flags themselves, not
> putting this in the generic code.

yep. let's fix that in the coming ones if they have config option.

a similar issue is the following check done in
unshare_nsproxy_namespaces() and copy_namespaces() :

```
 if (!capable(CAP_SYS_ADMIN))
  return -EPERM;
```

it would be interesting to let the namespace handle the unshare
permissions. CAP_SYS_ADMIN shouldn't be required for all namespaces.
ipc is one example.

C.

Subject: Re: [PATCH][RFC] Cleanup in namespaces unsharing
Posted by Pavel Emelianov on Fri, 08 Jun 2007 13:01:28 GMT
View Forum Message <> Reply to Message

Cedric Le Goater wrote:
> Pavel Emelianov wrote:
>> Cedric Le Goater wrote:
>>> Pavel Emelianov wrote:

[snip]

>>>> Did I miss something in the design or this patch worth merging?
>>> I've sent a more brutal patch in the past removing CONFIG_IPC_NS
>>> and CONFIG_UTS_NS. Might be a better idea ?
>> In case namespaces do not produce performance loss - yes.
>>
>> By that patch I also wanted to note that we'd better make
>> all the other namespaces check for flags themselves, not
>> putting this in the generic code.
>
> yep. let's fix that in the coming ones if they have config option.
>
> a similar issue is the following check done in
> unshare_nsproxy_namespaces() and copy_namespaces() :
>
> if (!capable(CAP_SYS_ADMIN))
>   return -EPERM;
>
> it would be interesting to let the namespace handle the unshare
> permissions. CAP_SYS_ADMIN shouldn't be required for all namespaces.
> ipc is one example.

Frankly, I think that some capability *is* required for

cloning the namespaces.

>
> C.
>

Thanks,
Pavel

Subject: Re: [PATCH][RFC] Cleanup in namespaces unsharing
Posted by serue on Fri, 08 Jun 2007 14:07:58 GMT
View Forum Message <> Reply to Message

Quoting Pavel Emelianov (xemul@openvz.org):
> Cedric Le Goater wrote:
> > Pavel Emelianov wrote:
> >> Cedric Le Goater wrote:
> >>> Pavel Emelianov wrote:
>
> [snip]
>
> >>>> Did I miss something in the design or this patch worth merging?
> >>> I've sent a more brutal patch in the past removing CONFIG_IPC_NS
> >>> and CONFIG_UTS_NS. Might be a better idea ?
> >> In case namespaces do not produce performance loss - yes.
> >>
> >> By that patch I also wanted to note that we'd better make
> >> all the other namespaces check for flags themselves, not
> >> putting this in the generic code.
> >
> > yep. let's fix that in the coming ones if they have config option.
> >
> > a similar issue is the following check done in
> > unshare_nsproxy_namespaces() and copy_namespaces() :
> >
> > if (!capable(CAP_SYS_ADMIN))
> >   return -EPERM;
> >
> > it would be interesting to let the namespace handle the unshare
> > permissions. CAP_SYS_ADMIN shouldn't be required for all namespaces.
> > ipc is one example.
>
> Frankly, I think that some capability *is* required for
> cloning the namespaces.

We can
 1. start a long per-namespace discussion on which namespaces really

need it
2. add a new CAP_SYS_UNSHARE capability so at least we're not
   using CAP_SYS_ADMIN for this
3. leave it as is

3 is really not that bad, though, since unshare activity can AFAICT
always be consolidated in small setuid helpers (or helpers with file
capabilities set :). Starting a vserver, starting a c-r job, and
unsharing mounts namespace on login using pam, can all be easily done
with privilege.

2 is unfortuntely a hassle since we have (last i looked) 1 free cap. Or
are we down to none?

I think had sent an email months ago starting a per-ns discussion on the
safety of not requiring a capability, but finding that coudl be a pain.
Off the bat, certain CLONE_NEWPID seems safe, right? CLONE_NEWNS could
be safe if we automatically made all the vfsmounts in the new ns slaves
of the original. CLONE_NEWNET would be pretty worthless since
presumably you'll always need CAP_NET_ADMIN to actually set up your
virtual net devices. CLONE_NEWIPC does seem safe. CLONE_NEWPTS should
be safe if we implement it the way Herbert suggested, with
/dev/pts/0 in a child ptsns showing up in /dev/pts/child_xyz/0 for the
parent.

thanks,
-serge