Subject: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Fri, 07 Apr 2006 18:36:00 GMT
View Forum Message <> Reply to Message

Sysctl uts patch. This clearly will need to be done another way, but
since sysctl itself needs to be container aware, 'the right thing' is
a separate patchset.

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

```
---
 kernel/sysctl.c |   38 ++++++++++++++++++++++++++++----------
 1 files changed, 28 insertions(+), 10 deletions(-)

40f7e1320c82efb6e875fc3bf44408cdfd093f21
diff --git a/kernel/sysctl.c b/kernel/sysctl.c
index e82726f..c2b18ef 100644
--- a/kernel/sysctl.c
+++ b/kernel/sysctl.c
@@ -233,8 +233,8 @@ static ctl_table kern_table[] = {
  {
   .ctl_name = KERN_OSTYPE,
   .procname = "ostype",
-  .data  = system_utsname.sysname,
-  .maxlen  = sizeof(system_utsname.sysname),
+  .data  = init_uts_ns.name.sysname,
+  .maxlen  = sizeof(init_uts_ns.name.sysname),
   .mode  = 0444,
   .proc_handler = &proc_doutsstring,
   .strategy = &sysctl_string,
@@ -242,8 +242,8 @@ static ctl_table kern_table[] = {
  {
   .ctl_name = KERN_OSRELEASE,
   .procname = "osrelease",
-  .data  = system_utsname.release,
-  .maxlen  = sizeof(system_utsname.release),
+  .data  = init_uts_ns.name.release,
+  .maxlen  = sizeof(init_uts_ns.name.release),
   .mode  = 0444,
   .proc_handler = &proc_doutsstring,
   .strategy = &sysctl_string,
@@ -251,8 +251,8 @@ static ctl_table kern_table[] = {
  {
   .ctl_name = KERN_VERSION,
   .procname = "version",
-  .data  = system_utsname.version,
-  .maxlen  = sizeof(system_utsname.version),
+  .data  = init_uts_ns.name.version,
+  .maxlen  = sizeof(init_uts_ns.name.version),
   .mode  = 0444,
   .proc_handler = &proc_doutsstring,
   .strategy = &sysctl_string,
@@ -260,8 +260,8 @@ static ctl_table kern_table[] = {
  {
   .ctl_name = KERN_NODENAME,
   .procname = "hostname",
-  .data  = system_utsname.nodename,
-  .maxlen  = sizeof(system_utsname.nodename),
+  .data  = init_uts_ns.name.nodename,
+  .maxlen  = sizeof(init_uts_ns.name.nodename),
   .mode  = 0644,
   .proc_handler = &proc_doutsstring,
   .strategy = &sysctl_string,
@@ -269,8 +269,8 @@ static ctl_table kern_table[] = {
  {
   .ctl_name = KERN_DOMAINNAME,
   .procname = "domainname",
-  .data  = system_utsname.domainname,
-  .maxlen  = sizeof(system_utsname.domainname),
+  .data  = init_uts_ns.name.domainname,
+  .maxlen  = sizeof(init_uts_ns.name.domainname),
   .mode  = 0644,
   .proc_handler = &proc_doutsstring,
   .strategy = &sysctl_string,
@@ -1619,6 +1619,24 @@ static int proc_doutsstring(ctl_table *t
 {
  int r;

+ switch (table->ctl_name) {
+  case KERN_OSTYPE:
+   table->data = utsname()->sysname;
+   break;
+  case KERN_OSRELEASE:
+   table->data = utsname()->release;
+   break;
+  case KERN_VERSION:
+   table->data = utsname()->version;
+   break;
+  case KERN_NODENAME:
+   table->data = utsname()->nodename;
+   break;
+  case KERN_DOMAINNAME:
+   table->data = utsname()->domainname;
+   break;
+ }
+
  if (!write) {
```

```
  down_read(&uts_sem);
  r=proc_dostring(table,0,filp,buffer,lenp, ppos);
-- 
1.2.4
```

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by dev on Wed, 19 Apr 2006 15:10:22 GMT

View Forum Message <> Reply to Message

Serge,

can we do nothing with sysctls at this moment, instead of commiting hacks?

Thanks,
Kirill

```
> Sysctl uts patch.  This clearly will need to be done another way, but
> since sysctl itself needs to be container aware, 'the right thing' is
> a separate patchset.
> 
> Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
> ---
>  kernel/sysctl.c |   38 ++++++++++++++++++++++++++++----------
>  1 files changed, 28 insertions(+), 10 deletions(-)
> 
> 40f7e1320c82efb6e875fc3bf44408cdfd093f21
> diff --git a/kernel/sysctl.c b/kernel/sysctl.c
> index e82726f..c2b18ef 100644
> --- a/kernel/sysctl.c
> +++ b/kernel/sysctl.c
> @@ -233,8 +233,8 @@ static ctl_table kern_table[] = {
>   {
>    .ctl_name = KERN_OSTYPE,
>    .procname = "ostype",
> -  .data  = system_utsname.sysname,
> -  .maxlen  = sizeof(system_utsname.sysname),
> +  .data  = init_uts_ns.name.sysname,
> +  .maxlen  = sizeof(init_uts_ns.name.sysname),
>    .mode  = 0444,
>    .proc_handler = &proc_doutsstring,
>    .strategy = &sysctl_string,
> @@ -242,8 +242,8 @@ static ctl_table kern_table[] = {
>   {
>    .ctl_name = KERN_OSRELEASE,
>    .procname = "osrelease",
> -  .data  = system_utsname.release,
> -  .maxlen  = sizeof(system_utsname.release),
> +  .data  = init_uts_ns.name.release,
> +  .maxlen  = sizeof(init_uts_ns.name.release),
>    .mode  = 0444,
>    .proc_handler = &proc_doutsstring,
>    .strategy = &sysctl_string,
> @@ -251,8 +251,8 @@ static ctl_table kern_table[] = {
>   {
>    .ctl_name = KERN_VERSION,
>    .procname = "version",
> -  .data  = system_utsname.version,
> -  .maxlen  = sizeof(system_utsname.version),
> +  .data  = init_uts_ns.name.version,
> +  .maxlen  = sizeof(init_uts_ns.name.version),
>    .mode  = 0444,
>    .proc_handler = &proc_doutsstring,
>    .strategy = &sysctl_string,
> @@ -260,8 +260,8 @@ static ctl_table kern_table[] = {
>   {
>    .ctl_name = KERN_NODENAME,
>    .procname = "hostname",
> -  .data  = system_utsname.nodename,
> -  .maxlen  = sizeof(system_utsname.nodename),
> +  .data  = init_uts_ns.name.nodename,
> +  .maxlen  = sizeof(init_uts_ns.name.nodename),
>    .mode  = 0644,
>    .proc_handler = &proc_doutsstring,
>    .strategy = &sysctl_string,
> @@ -269,8 +269,8 @@ static ctl_table kern_table[] = {
>   {
>    .ctl_name = KERN_DOMAINNAME,
>    .procname = "domainname",
> -  .data  = system_utsname.domainname,
> -  .maxlen  = sizeof(system_utsname.domainname),
> +  .data  = init_uts_ns.name.domainname,
> +  .maxlen  = sizeof(init_uts_ns.name.domainname),
>    .mode  = 0644,
>    .proc_handler = &proc_doutsstring,
>    .strategy = &sysctl_string,
> @@ -1619,6 +1619,24 @@ static int proc_doutsstring(ctl_table *t
>  {
>   int r;
>
> + switch (table->ctl_name) {
> +  case KERN_OSTYPE:
> +   table->data = utsname()->sysname;
> +   break;
> +  case KERN_OSRELEASE:
> +   table->data = utsname()->release;
> +   break;
> +  case KERN_VERSION:
> +   table->data = utsname()->version;
> +   break;
> +  case KERN_NODENAME:
> +   table->data = utsname()->nodename;
> +   break;
> +  case KERN_DOMAINNAME:
> +   table->data = utsname()->domainname;
> +   break;
> + }
> +
>   if (!write) {
>    down_read(&uts_sem);
>    r=proc_dostring(table,0,filp,buffer,lenp, ppos);
```

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Wed, 19 Apr 2006 15:21:29 GMT

View Forum Message <> Reply to Message

Quoting Kirill Korotaev (dev@sw.ru):
> Serge,
>
> can we do nothing with sysctls at this moment, instead of commiting hacks?

Please look closer at the patch.

I *am* doing nothing with sysctls.

system_utsname no longer exists, and the way to get to that is by using
init_uts_ns.name. That's all this does.

-serge

```
>
> Thanks,
> Kirill
>
> >Sysctl uts patch.  This clearly will need to be done another way, but
> >since sysctl itself needs to be container aware, 'the right thing' is
> >a separate patchset.
> >
> >Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
> >---
> > kernel/sysctl.c |   38 ++++++++++++++++++++++++++++----------
> > 1 files changed, 28 insertions(+), 10 deletions(-)
> >
> >40f7e1320c82efb6e875fc3bf44408cdfd093f21
> >diff --git a/kernel/sysctl.c b/kernel/sysctl.c
> >index e82726f..c2b18ef 100644
> >--- a/kernel/sysctl.c
> >+++ b/kernel/sysctl.c
> >@@ -233,8 +233,8 @@ static ctl_table kern_table[] = {
> >  {
> >   .ctl_name = KERN_OSTYPE,
> >   .procname = "ostype",
> >-  .data  = system_utsname.sysname,
> >-  .maxlen  = sizeof(system_utsname.sysname),
> >+  .data  = init_uts_ns.name.sysname,
> >+  .maxlen  = sizeof(init_uts_ns.name.sysname),
> >   .mode  = 0444,
> >   .proc_handler = &proc_doutsstring,
> >   .strategy = &sysctl_string,
> >@@ -242,8 +242,8 @@ static ctl_table kern_table[] = {
> >  {
> >   .ctl_name = KERN_OSRELEASE,
> >   .procname = "osrelease",
> >-  .data  = system_utsname.release,
> >-  .maxlen  = sizeof(system_utsname.release),
> >+  .data  = init_uts_ns.name.release,
> >+  .maxlen  = sizeof(init_uts_ns.name.release),
> >   .mode  = 0444,
> >   .proc_handler = &proc_doutsstring,
> >   .strategy = &sysctl_string,
> >@@ -251,8 +251,8 @@ static ctl_table kern_table[] = {
> >  {
> >   .ctl_name = KERN_VERSION,
> >   .procname = "version",
> >-  .data  = system_utsname.version,
> >-  .maxlen  = sizeof(system_utsname.version),
> >+  .data  = init_uts_ns.name.version,
> >+  .maxlen  = sizeof(init_uts_ns.name.version),
> >   .mode  = 0444,
> >   .proc_handler = &proc_doutsstring,
> >   .strategy = &sysctl_string,
> >@@ -260,8 +260,8 @@ static ctl_table kern_table[] = {
> >  {
> >   .ctl_name = KERN_NODENAME,
> >   .procname = "hostname",
> >-  .data  = system_utsname.nodename,
> >-  .maxlen  = sizeof(system_utsname.nodename),
> >+  .data  = init_uts_ns.name.nodename,
> >+  .maxlen  = sizeof(init_uts_ns.name.nodename),
> >   .mode  = 0644,
> >   .proc_handler = &proc_doutsstring,
> >   .strategy = &sysctl_string,
> >@@ -269,8 +269,8 @@ static ctl_table kern_table[] = {
> >  {
> >   .ctl_name = KERN_DOMAINNAME,
> >   .procname = "domainname",
> >-  .data  = system_utsname.domainname,
> >-  .maxlen  = sizeof(system_utsname.domainname),
> >+  .data  = init_uts_ns.name.domainname,
> >+  .maxlen  = sizeof(init_uts_ns.name.domainname),
> >   .mode  = 0644,
> >   .proc_handler = &proc_doutsstring,
> >   .strategy = &sysctl_string,
> >@@ -1619,6 +1619,24 @@ static int proc_doutsstring(ctl_table *t
> > {
> >  int r;
> >
> >+ switch (table->ctl_name) {
> >+  case KERN_OSTYPE:
> >+   table->data = utsname()->sysname;
> >+   break;
> >+  case KERN_OSRELEASE:
> >+   table->data = utsname()->release;
> >+   break;
> >+  case KERN_VERSION:
> >+   table->data = utsname()->version;
> >+   break;
> >+  case KERN_NODENAME:
> >+   table->data = utsname()->nodename;
> >+   break;
> >+  case KERN_DOMAINNAME:
> >+   table->data = utsname()->domainname;
> >+   break;
> >+ }
> >+
> >  if (!write) {
> >   down_read(&uts_sem);
> >   r=proc_dostring(table,0,filp,buffer,lenp, ppos);
>
```

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 15:29:32 GMT
View Forum Message <> Reply to Message

Kirill Korotaev <dev@sw.ru> writes:

> Serge,
>

> can we do nothing with sysctls at this moment, instead of commiting hacks?

Except that we modify a static table changing the uts behaviour in
proc_doutsstring isn't all that bad.

I'm just about to start on something more comprehensive, in
the sysctl case.

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by dev on Wed, 19 Apr 2006 15:43:24 GMT

View Forum Message <> Reply to Message

Serge,

> Please look closer at the patch.
> I *am* doing nothing with sysctls.
>
> system_utsname no longer exists, and the way to get to that is by using
> init_uts_ns.name. That's all this does.

Sorry for being not concrete enough.
I mean switch () in the code. Until we decided how to virtualize
sysctls/proc, I believe no dead code/hacks should be commited. IMHO.

FYI, I strongly object against virtualizing sysctls this way as it is
not flexible and is a real hack from my POV.

Sure, change of system_utsname.sysname -> init_uts_ns.name.sysname is Ok.

Thanks,
Kirill

```
>>>Sysctl uts patch.  This clearly will need to be done another way, but
>>>since sysctl itself needs to be container aware, 'the right thing' is
>>>a separate patchset.
>>>
>>>Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
>>>---
>>>kernel/sysctl.c |   38 ++++++++++++++++++++++++++++----------
>>>1 files changed, 28 insertions(+), 10 deletions(-)
>>>
>>>40f7e1320c82efb6e875fc3bf44408cdfd093f21
>>>diff --git a/kernel/sysctl.c b/kernel/sysctl.c
>>>index e82726f..c2b18ef 100644
>>>--- a/kernel/sysctl.c
>>>+++ b/kernel/sysctl.c
```

```
>>>@@ -233,8 +233,8 @@ static ctl_table kern_table[] = {
>>> {
>>>  .ctl_name = KERN_OSTYPE,
>>>  .procname = "ostype",
>>>-  .data  = system_utsname.sysname,
>>>-  .maxlen  = sizeof(system_utsname.sysname),
>>>+  .data  = init_uts_ns.name.sysname,
>>>+  .maxlen  = sizeof(init_uts_ns.name.sysname),
>>>  .mode  = 0444,
>>>  .proc_handler = &proc_doutsstring,
>>>  .strategy = &sysctl_string,
>>>@@ -242,8 +242,8 @@ static ctl_table kern_table[] = {
>>> {
>>>  .ctl_name = KERN_OSRELEASE,
>>>  .procname = "osrelease",
>>>-  .data  = system_utsname.release,
>>>-  .maxlen  = sizeof(system_utsname.release),
>>>+  .data  = init_uts_ns.name.release,
>>>+  .maxlen  = sizeof(init_uts_ns.name.release),
>>>  .mode  = 0444,
>>>  .proc_handler = &proc_doutsstring,
>>>  .strategy = &sysctl_string,
>>>@@ -251,8 +251,8 @@ static ctl_table kern_table[] = {
>>> {
>>>  .ctl_name = KERN_VERSION,
>>>  .procname = "version",
>>>-  .data  = system_utsname.version,
>>>-  .maxlen  = sizeof(system_utsname.version),
>>>+  .data  = init_uts_ns.name.version,
>>>+  .maxlen  = sizeof(init_uts_ns.name.version),
>>>  .mode  = 0444,
>>>  .proc_handler = &proc_doutsstring,
>>>  .strategy = &sysctl_string,
>>>@@ -260,8 +260,8 @@ static ctl_table kern_table[] = {
>>> {
>>>  .ctl_name = KERN_NODENAME,
>>>  .procname = "hostname",
>>>-  .data  = system_utsname.nodename,
>>>-  .maxlen  = sizeof(system_utsname.nodename),
>>>+  .data  = init_uts_ns.name.nodename,
>>>+  .maxlen  = sizeof(init_uts_ns.name.nodename),
>>>  .mode  = 0644,
>>>  .proc_handler = &proc_doutsstring,
>>>  .strategy = &sysctl_string,
>>>@@ -269,8 +269,8 @@ static ctl_table kern_table[] = {
>>> {
>>>  .ctl_name = KERN_DOMAINNAME,
>>>  .procname = "domainname",
>>>-  .data  = system_utsname.domainname,
>>>-  .maxlen  = sizeof(system_utsname.domainname),
>>>+  .data  = init_uts_ns.name.domainname,
>>>+  .maxlen  = sizeof(init_uts_ns.name.domainname),
>>>  .mode  = 0644,
>>>  .proc_handler = &proc_doutsstring,
>>>  .strategy = &sysctl_string,
>>>@@ -1619,6 +1619,24 @@ static int proc_doutsstring(ctl_table *t
>>>{
>>> int r;
>>>
>>>+ switch (table->ctl_name) {
>>>+  case KERN_OSTYPE:
>>>+   table->data = utsname()->sysname;
>>>+   break;
>>>+  case KERN_OSRELEASE:
>>>+   table->data = utsname()->release;
>>>+   break;
>>>+  case KERN_VERSION:
>>>+   table->data = utsname()->version;
>>>+   break;
>>>+  case KERN_NODENAME:
>>>+   table->data = utsname()->nodename;
>>>+   break;
>>>+  case KERN_DOMAINNAME:
>>>+   table->data = utsname()->domainname;
>>>+   break;
>>>+ }
>>>+
>>> if (!write) {
>>>  down_read(&uts_sem);
>>>  r=proc_dostring(table,0,filp,buffer,lenp, ppos);
>>
>
```

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 15:52:42 GMT

View Forum Message <> Reply to Message

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

> Quoting Kirill Korotaev (dev@sw.ru):
>> Serge,
>>
>> can we do nothing with sysctls at this moment, instead of commiting hacks?
>
> Please look closer at the patch.

>
> I *am* doing nothing with sysctls.
>
> system_utsname no longer exists, and the way to get to that is by using
> init_uts_ns.name. That's all this does.

Ack. I probably read that question backwards.

Yes you must at least touch kernel/sysctl.c when you kill
system_utsname.

I read it as: Can we do nothing better with sysctls that commiting hacks?

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by Dave Hansen on Wed, 19 Apr 2006 16:23:34 GMT

View Forum Message <> Reply to Message

Besides ipc and utsnames, can anybody think of some other things in
sysctl that we really need to virtualize?

It seems to me that most of the other stuff is kernel-global and we
simply won't allow anything in a container to touch it.

That said, there may be things in the future that need to get added as
we separate out different subsystems. Things like min_free_kbytes could
have a container-centric meaning (although I think that is probably a
really bad one to mess with).

I have a slightly revamped way of doing the sysv namespace sysctl code.
I've attached a couple of (still pretty raw) patches. Do these still
fall in the "hacks" category?

-- Dave

File Attachments

1) sysv-do-sysctl-strategies2.patch, downloaded 309 times
2) sysv-do-sysctl-strategies1.patch, downloaded 315 times
3) sysv-do-sysctl-strategies0.patch, downloaded 292 times

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 16:52:54 GMT

View Forum Message <> Reply to Message

Dave Hansen <haveblue@us.ibm.com> writes:

> Besides ipc and utsnames, can anybody think of some other things in
> sysctl that we really need to virtualize?

All of the networking entries.

> It seems to me that most of the other stuff is kernel-global and we
> simply won't allow anything in a container to touch it.
>
> That said, there may be things in the future that need to get added as
> we separate out different subsystems. Things like min_free_kbytes could
> have a container-centric meaning (although I think that is probably a
> really bad one to mess with).
>
> I have a slightly revamped way of doing the sysv namespace sysctl code.
> I've attached a couple of (still pretty raw) patches. Do these still
> fall in the "hacks" category?

Only in that you attacked the wrong piece of the puzzle.
The strategy table entries simply need to die, or be rewritten
to use the appropriate proc entries.

The proc entries are the real interface, and the two pieces
don't share an implementation unfortunately.

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by Cedric Le Goater on Wed, 19 Apr 2006 16:54:09 GMT

View Forum Message <> Reply to Message

Hello !

Kirill Korotaev wrote:
> Serge,
>
>> Please look closer at the patch.
>> I *am* doing nothing with sysctls.
>>
>> system_utsname no longer exists, and the way to get to that is by using
>> init_uts_ns.name. That's all this does.
> Sorry for being not concrete enough.
> I mean switch () in the code. Until we decided how to virtualize
> sysctls/proc, I believe no dead code/hacks should be commited. IMHO.

How could we improve that hack ? Removing the modification of the static

table can easily be worked around but getting rid of the switch() statement
is more difficult. Any idea ?

> FYI, I strongly object against virtualizing sysctls this way as it is
> not flexible and is a real hack from my POV.

what is the issue with flexibility ?

thanks,

C.

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 17:10:01 GMT

View Forum Message <> Reply to Message

Cedric Le Goater <clg@fr.ibm.com> writes:

> Hello !
>
> Kirill Korotaev wrote:
>> Serge,
>>
>>> Please look closer at the patch.
>>> I *am* doing nothing with sysctls.
>>>
>>> system_utsname no longer exists, and the way to get to that is by using
>>> init_uts_ns.name. That's all this does.
>> Sorry for being not concrete enough.
>> I mean switch () in the code. Until we decided how to virtualize
>> sysctls/proc, I believe no dead code/hacks should be commited. IMHO.
>
> How could we improve that hack ? Removing the modification of the static
> table can easily be worked around but getting rid of the switch() statement
> is more difficult. Any idea ?

Store offsetof in data. Not that for such a small case it really matters,
but it probably improves maintenance by a little bit.

>> FYI, I strongly object against virtualizing sysctls this way as it is
>> not flexible and is a real hack from my POV.
>
> what is the issue with flexibility ?

The only other thing I would like to see is the process argument passed
in.

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Wed, 19 Apr 2006 17:10:56 GMT

View Forum Message <> Reply to Message

Quoting Kirill Korotaev (dev@sw.ru):
> Serge,
>
> >Please look closer at the patch.
> >I *am* doing nothing with sysctls.
> >
> >system_utsname no longer exists, and the way to get to that is by using
> >init_uts_ns.name. That's all this does.
> Sorry for being not concrete enough.
> I mean switch () in the code. Until we decided how to virtualize
> sysctls/proc, I believe no dead code/hacks should be commited. IMHO.
>
> FYI, I strongly object against virtualizing sysctls this way as it is
> not flexible and is a real hack from my POV.

Oops, I forgot that was there!

Sorry.

Yup, I'm fine with leaving that out. After all, nothing in the
non-debugging patchset allows userspace to clone the utsnamespace yet,
so it's tough to argue that leaving out that switch impacts
functionality :)

I believe Dave is working on a more acceptable sysctl adaptation, though
I'm not sure when he'll have a patch to submit. In any case, one clear
concise piece at a time.

thanks,
-serge

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by Dave Hansen on Wed, 19 Apr 2006 17:19:18 GMT

View Forum Message <> Reply to Message

On Wed, 2006-04-19 at 10:52 -0600, Eric W. Biederman wrote:
> Dave Hansen <haveblue@us.ibm.com> writes:
>
> > Besides ipc and utsnames, can anybody think of some other things in

> > sysctl that we really need to virtualize?
>
> All of the networking entries.
...
> Only in that you attacked the wrong piece of the puzzle.
> The strategy table entries simply need to die, or be rewritten
> to use the appropriate proc entries.

If we are limited to ipc, utsname, and network, I'd be worried trying to
justify _too_ much infrastructure. The network namespaces are not going
to be solved any time soon. Why not have something like this which is a
quite simple, understandable, minor hack?

> The proc entries are the real interface, and the two pieces
> don't share an implementation unfortunately.

You're saying that the proc interface doesn't use the ->strategy entry?
That isn't what I remember, but I could be completely wrong.

-- Dave

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 17:37:00 GMT
View Forum Message <> Reply to Message

Dave Hansen <haveblue@us.ibm.com> writes:

> On Wed, 2006-04-19 at 10:52 -0600, Eric W. Biederman wrote:
>> Dave Hansen <haveblue@us.ibm.com> writes:
>>
>> > Besides ipc and utsnames, can anybody think of some other things in
>> > sysctl that we really need to virtualize?
>>
>> All of the networking entries.
> ...
>> Only in that you attacked the wrong piece of the puzzle.
>> The strategy table entries simply need to die, or be rewritten
>> to use the appropriate proc entries.
>
> If we are limited to ipc, utsname, and network, I'd be worried trying to
> justify _too_ much infrastructure. The network namespaces are not going
> to be solved any time soon. Why not have something like this which is a
> quite simple, understandable, minor hack?

Because it doesn't affect what happens in /proc/sys !
Strategy routines only affect sys_sysctl.

As strategy routines I have no real problems with them.
I haven't looked terribly closely yet.

>> The proc entries are the real interface, and the two pieces
>> don't share an implementation unfortunately.
>
> You're saying that the proc interface doesn't use the ->strategy entry?
> That isn't what I remember, but I could be completely wrong.

Exactly.   I have a patch I will be sending out shortly that
make sys_sysctl a compile time option (so we can seriously start killing it)
and it compiles out the strategy routines and /proc/sys still works :)

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 17:48:47 GMT
View Forum Message <> Reply to Message

Dave Hansen <haveblue@us.ibm.com> writes:

> On Wed, 2006-04-19 at 10:52 -0600, Eric W. Biederman wrote:
>> Dave Hansen <haveblue@us.ibm.com> writes:
>>
>> > Besides ipc and utsnames, can anybody think of some other things in
>> > sysctl that we really need to virtualize?
>>
>> All of the networking entries.
> ...
>> Only in that you attacked the wrong piece of the puzzle.
>> The strategy table entries simply need to die, or be rewritten
>> to use the appropriate proc entries.
>
> If we are limited to ipc, utsname, and network, I'd be worried trying to
> justify _too_ much infrastructure. The network namespaces are not going
> to be solved any time soon. Why not have something like this which is a
> quite simple, understandable, minor hack?

As for the network namespaces. It actually isn't that hard, but
it is tedious and big.   Once we get ipc and uts it will probably be
the namespace to merge. I have the basic code sitting out on a branch.
Getting the little things like sysctl, sorted out are the primary
problems. At the same time we don't have to solve the problems for
the network namespace now. Just don't expect it way of in the
indefinite future, either.

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Wed, 19 Apr 2006 17:51:23 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Kirill Korotaev <dev@sw.ru> writes:
>
> > Serge,
> >
> > can we do nothing with sysctls at this moment, instead of commiting hacks?
>
> Except that we modify a static table changing the uts behaviour in
> proc_doutsstring isn't all that bad.
>
> I'm just about to start on something more comprehensive, in
> the sysctl case.

So assuming that I take out the switch(), leaving that for a better
solution by Eric (or Dave, or whoever),

Is it time to ask for the utsname namespace patch to be tried out
in -mm?

thanks,
-serge

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by ebiederm on Wed, 19 Apr 2006 18:27:01 GMT
View Forum Message <> Reply to Message

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

> Quoting Eric W. Biederman (ebiederm@xmission.com):
>> Kirill Korotaev <dev@sw.ru> writes:
>>
>> > Serge,
>> >
>> > can we do nothing with sysctls at this moment, instead of commiting hacks?
>>
>> Except that we modify a static table changing the uts behaviour in
>> proc_doutsstring isn't all that bad.
>>
>> I'm just about to start on something more comprehensive, in
>> the sysctl case.
>
> So assuming that I take out the switch(), leaving that for a better
> solution by Eric (or Dave, or whoever),

>
> Is it time to ask for the utsname namespace patch to be tried out
> in -mm?

Can we please suggest a syscall interface?

Eric

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Wed, 19 Apr 2006 20:24:14 GMT

View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > Quoting Eric W. Biederman (ebiederm@xmission.com):
> >> Kirill Korotaev <dev@sw.ru> writes:
> >>
> >> > Serge,
> >> >
> >> > can we do nothing with sysctls at this moment, instead of commiting hacks?
> >>
> >> Except that we modify a static table changing the uts behaviour in
> >> proc_doutsstring isn't all that bad.
> >>
> >> I'm just about to start on something more comprehensive, in
> >> the sysctl case.
> >
> > So assuming that I take out the switch(), leaving that for a better
> > solution by Eric (or Dave, or whoever),
> >
> > Is it time to ask for the utsname namespace patch to be tried out
> > in -mm?
>
> Can we please suggest a syscall interface?

We can, but I was hoping that would be a separate patch, separate
discussion.

Are you asking for a new syscall, specifically to unshare utsname()? Or
for discussion over whether we want to do
 one syscall per namespace
 extend CLONE_NEWns flags
 use unshare
 use namespacefs

-serge

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by Sam Vilain on Wed, 19 Apr 2006 21:44:54 GMT

View Forum Message <> Reply to Message

Eric W. Biederman wrote:

>>Is it time to ask for the utsname namespace patch to be tried out
>>in -mm?
>>
>>
>
>Can we please suggest a syscall interface?
>
>

What was wrong with the method of the one I posted / extracted from the
Linux-VServer project? I mean, apart from the baggage which I intend to
remove for the next posting.

The concept was - have a single syscall with versioned subcommands. We
can throw all of the namespace syscalls in there.

Sam.

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Thu, 20 Apr 2006 17:05:59 GMT

View Forum Message <> Reply to Message

Quoting Sam Vilain (sam@vilain.net):
> Eric W. Biederman wrote:
>
> >>Is it time to ask for the utsname namespace patch to be tried out
> >>in -mm?
> >>
> >>
> >
> >Can we please suggest a syscall interface?
> >
> >
>
> What was wrong with the method of the one I posted / extracted from the
> Linux-VServer project? I mean, apart from the baggage which I intend to
> remove for the next posting.
>
> The concept was - have a single syscall with versioned subcommands. We
> can throw all of the namespace syscalls in there.
>

> Sam.

Well IIUC on the whole having one syscall multiplexing onto various commands is frowned upon. But please resubmit when you're ready, and we'll see what ppl think of it.

Can you have a version on top of my utsname patches, hooking into the utsname unsharing fn?

thanks,
-serge

Subject: Re: [RFC][PATCH 4/5] utsname namespaces: sysctl hack
Posted by serue on Tue, 25 Apr 2006 22:00:22 GMT

View Forum Message <> Reply to Message

Quoting Sam Vilain (sam@vilain.net):
> Eric W. Biederman wrote:
>
> >>Is it time to ask for the utsname namespace patch to be tried out
> >>in -mm?
> >>
> >>
> >
> >Can we please suggest a syscall interface?

Eric,

Did you have any ideas for how you'd want to interface to look? Are you fine with the vserver approach?

> What was wrong with the method of the one I posted / extracted from the
> Linux-VServer project? I mean, apart from the baggage which I intend to
> remove for the next posting.

Sam,

Are you working on a next posting?

-serge