Subject: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Stanislav Kinsbursky on Mon, 13 Aug 2012 11:37:31 GMT
View Forum Message <> Reply to Message

v2:
1) rpc_clnt_set_nodename() prototype updated.
2) fixed errors in comment.

When child reaper exits, it can destroy mount namespace it belongs to, and if
there are NFS mounts inside, then it will try to umount them. But in this
point current->nsproxy is set to NULL and all namespaces will be destroyed one
by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
---
 net/sunrpc/clnt.c | 16 +++++++++++++---
 1 files changed, 13 insertions(+), 3 deletions(-)

diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
index 9a9676e..8fbcbc8 100644
--- a/net/sunrpc/clnt.c
+++ b/net/sunrpc/clnt.c
@@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
  return rpc_pipefs_notifier_unregister(&rpc_clients_block);
 }

-static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
+static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)
 {
+ const char *nodename;
+
+ /*
+ * We have to protect against dying child reaper, which has released
+ * its nsproxy already and is trying to destroy mount namespace.
+ */
+ if (current->nsproxy == NULL)
+ return;
+
+ nodename = utsname()->nodename;
  clnt->cl_nodelen = strlen(nodename);
  if (clnt->cl_nodelen > UNX_MAXNODENAME)
   clnt->cl_nodelen = UNX_MAXNODENAME;
@@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct rpc_create_args *args,
stru
  }

  /* save the nodename */

```
- rpc_clnt_set_nodename(clnt, utsname()->nodename);
+ rpc_clnt_set_nodename(clnt);
  rpc_register_client(clnt);
  return clnt;

@@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
  err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
  if (err != 0)
   goto out_no_path;
- rpc_clnt_set_nodename(new, utsname()->nodename);
+ rpc_clnt_set_nodename(new);
  if (new->cl_auth)
   atomic_inc(&new->cl_auth->au_count);
  atomic_inc(&clnt->cl_count);
```

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation

Posted by Jeff Layton on Mon, 13 Aug 2012 12:10:54 GMT

View Forum Message <> Reply to Message

On Mon, 13 Aug 2012 15:37:31 +0400
Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:

> v2:
> 1) rpc_clnt_set_nodename() prototype updated.
> 2) fixed errors in comment.
>
> When child reaper exits, it can destroy mount namespace it belongs to, and if
> there are NFS mounts inside, then it will try to umount them. But in this
> point current->nsproxy is set to NULL and all namespaces will be destroyed one
> by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> ---
> net/sunrpc/clnt.c | 16 ++++++++++++---
> 1 files changed, 13 insertions(+), 3 deletions(-)
>
> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> index 9a9676e..8fbcbc8 100644
> --- a/net/sunrpc/clnt.c
> +++ b/net/sunrpc/clnt.c
> @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
>   return rpc_pipefs_notifier_unregister(&rpc_clients_block);
> }
>
> -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
> +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)

> {
> + const char *nodename;
> +
> + /*
> + * We have to protect against dying child reaper, which has released
> + * its nsproxy already and is trying to destroy mount namespace.
> + */
> + if (current->nsproxy == NULL)
> + return;
> +
> + nodename = utsname()->nodename;
>   clnt->cl_nodelen = strlen(nodename);
>   if (clnt->cl_nodelen > UNX_MAXNODENAME)
>   clnt->cl_nodelen = UNX_MAXNODENAME;
> @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct rpc_create_args
*args, stru
>   }
>
>   /* save the nodename */
> - rpc_clnt_set_nodename(clnt, utsname()->nodename);
> + rpc_clnt_set_nodename(clnt);
>   rpc_register_client(clnt);
>   return clnt;
>
> @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
>   err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
>   if (err != 0)
>   goto out_no_path;
> - rpc_clnt_set_nodename(new, utsname()->nodename);
> + rpc_clnt_set_nodename(new);
>   if (new->cl_auth)
>   atomic_inc(&new->cl_auth->au_count);
>   atomic_inc(&clnt->cl_count);
>
> --
> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html

Acked-by: Jeff Layton <jlayton@redhat.com>

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Myklebust, Trond on Fri, 07 Sep 2012 22:32:16 GMT
View Forum Message <> Reply to Message

On Mon, 2012-08-13 at 08:10 -0400, Jeff Layton wrote:

> On Mon, 13 Aug 2012 15:37:31 +0400
> Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:
>
> > v2:
> > 1) rpc_clnt_set_nodename() prototype updated.
> > 2) fixed errors in comment.
> >
> > When child reaper exits, it can destroy mount namespace it belongs to, and if
> > there are NFS mounts inside, then it will try to umount them. But in this
> > point current->nsproxy is set to NULL and all namespaces will be destroyed one
> > by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
> >
> > Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> > ---
> > net/sunrpc/clnt.c | 16 +++++++++++++---
> > 1 files changed, 13 insertions(+), 3 deletions(-)
> >
> > diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> > index 9a9676e..8fbcbc8 100644
> > --- a/net/sunrpc/clnt.c
> > +++ b/net/sunrpc/clnt.c
> > @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
> >   return rpc_pipefs_notifier_unregister(&rpc_clients_block);
> > }
> >
> > -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
> > +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)
> > {
> > + const char *nodename;
> > +
> > + /*
> > + * We have to protect against dying child reaper, which has released
> > + * its nsproxy already and is trying to destroy mount namespace.
> > + */
> > + if (current->nsproxy == NULL)
> > + return;
> > +
> > + nodename = utsname()->nodename;
> >   clnt->cl_nodelen = strlen(nodename);
> >   if (clnt->cl_nodelen > UNX_MAXNODENAME)
> >   clnt->cl_nodelen = UNX_MAXNODENAME;
> > @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct rpc_create_args
*args, stru
> >   }
> >
> >   /* save the nodename */
> > - rpc_clnt_set_nodename(clnt, utsname()->nodename);
> > + rpc_clnt_set_nodename(clnt);

> >   rpc_register_client(clnt);
> >   return clnt;
> >
> > @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
> >   err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
> >   if (err != 0)
> >   goto out_no_path;
> > - rpc_clnt_set_nodename(new, utsname()->nodename);
> > + rpc_clnt_set_nodename(new);
> >   if (new->cl_auth)
> >   atomic_inc(&new->cl_auth->au_count);
> >   atomic_inc(&clnt->cl_count);
> >
> > --
> > To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
> > the body of a message to majordomo@vger.kernel.org
> > More majordomo info at http://vger.kernel.org/majordomo-info.html
>
> Acked-by: Jeff Layton <jlayton@redhat.com>

OK, colour me confused (again). Why should a umount trigger an
rpc_create() or rpc_clone_client()?

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Stanislav Kinsbursky on Sat, 08 Sep 2012 05:59:53 GMT
View Forum Message <> Reply to Message

> On Mon, 2012-08-13 at 08:10 -0400, Jeff Layton wrote:
>> On Mon, 13 Aug 2012 15:37:31 +0400
>> Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:
>>
>>> v2:
>>> 1) rpc_clnt_set_nodename() prototype updated.
>>> 2) fixed errors in comment.
>>>
>>> When child reaper exits, it can destroy mount namespace it belongs to, and if
>>> there are NFS mounts inside, then it will try to umount them. But in this

```
>>> point current->nsproxy is set to NULL and all namespaces will be destroyed one
>>> by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
>>>
>>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>>> ---
>>>   net/sunrpc/clnt.c |   16 ++++++++++++---
>>>   1 files changed, 13 insertions(+), 3 deletions(-)
>>>
>>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
>>> index 9a9676e..8fbcbc8 100644
>>> --- a/net/sunrpc/clnt.c
>>> +++ b/net/sunrpc/clnt.c
>>> @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
>>>    return rpc_pipefs_notifier_unregister(&rpc_clients_block);
>>>   }
>>>
>>> -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
>>> +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)
>>>   {
>>> + const char *nodename;
>>> +
>>> + /*
>>> +  * We have to protect against dying child reaper, which has released
>>> +  * its nsproxy already and is trying to destroy mount namespace.
>>> +  */
>>> + if (current->nsproxy == NULL)
>>> +  return;
>>> +
>>> + nodename = utsname()->nodename;
>>>    clnt->cl_nodelen = strlen(nodename);
>>>    if (clnt->cl_nodelen > UNX_MAXNODENAME)
>>>     clnt->cl_nodelen = UNX_MAXNODENAME;
>>> @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct rpc_create_args
*args, stru
>>>    }
>>>
>>>    /* save the nodename */
>>> - rpc_clnt_set_nodename(clnt, utsname()->nodename);
>>> + rpc_clnt_set_nodename(clnt);
>>>    rpc_register_client(clnt);
>>>    return clnt;
>>>
>>> @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
>>>    err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
>>>    if (err != 0)
>>>     goto out_no_path;
>>> - rpc_clnt_set_nodename(new, utsname()->nodename);
>>> + rpc_clnt_set_nodename(new);
```

>>>   if (new->cl_auth)
>>>   atomic_inc(&new->cl_auth->au_count);
>>>   atomic_inc(&clnt->cl_count);
>>>
>>> --
>>> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
>>> the body of a message to majordomo@vger.kernel.org
>>> More majordomo info at http://vger.kernel.org/majordomo-info.html
>> Acked-by: Jeff Layton <jlayton@redhat.com>
> OK, colour me confused (again).

What color?

> Why should a umount trigger an
> rpc_create() or rpc_clone_client()?

It calls nsm_create().
Here is the trace (https://bugzilla.redhat.com/show_bug.cgi?id=830862,
comment 68):

CR2: 0000000000000008
Process mysqld

Call Trace:
? __schedule+0x3c7
nsm_create+0x8b
nsm_mon_unmon+0x64
nlm_destroy_host_locked+0x6b
nlmclnt_release_host+0x88
nlmclnt_done+0x1a
nfs_destroy_server+0x24
nfs_free_server+0xce
nfs_kill_super+0x34
deactivate_locked_super+0x57
deactivate_super+0x4e
mntput_no_expire+0xcc
mntput+0x26
release_mounts+0x77
put_mnt_ns+0x78
free_nsproxy+0x1f
switch_task_namespaces+0x50
exit_task_namespaces+0x10
do_exit+0x456
do_group_exit+0x3f
sys_exit_group+0x17
system_call_fastpath+0x16
RIP rpc_create+0x401 [sunrpc]
Kernel panic - not syncing

>

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Myklebust, Trond on Sat, 08 Sep 2012 14:33:09 GMT
View Forum Message <> Reply to Message

On Sat, 2012-09-08 at 08:59 +0300, Stanislav Kinsbursky wrote:

> > On Mon, 2012-08-13 at 08:10 -0400, Jeff Layton wrote:
> >> On Mon, 13 Aug 2012 15:37:31 +0400
> >> Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:
> >>
> >>> v2:
> >>> 1) rpc_clnt_set_nodename() prototype updated.
> >>> 2) fixed errors in comment.
> >>>
> >>> When child reaper exits, it can destroy mount namespace it belongs to, and if
> >>> there are NFS mounts inside, then it will try to umount them. But in this
> >>> point current->nsproxy is set to NULL and all namespaces will be destroyed one
> >>> by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
> >>>
> >>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> >>> ---
> >>> net/sunrpc/clnt.c | 16 +++++++++++++---
> >>> 1 files changed, 13 insertions(+), 3 deletions(-)
> >>>
> >>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> >>> index 9a9676e..8fbcbc8 100644
> >>> --- a/net/sunrpc/clnt.c
> >>> +++ b/net/sunrpc/clnt.c
> >>> @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
> >>>   return rpc_pipefs_notifier_unregister(&rpc_clients_block);
> >>> }
> >>>
> >>> -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
> >>> +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)
> >>> {
> >>> + const char *nodename;
> >>> +
> >>> + /*
> >>> + * We have to protect against dying child reaper, which has released
> >>> + * its nsproxy already and is trying to destroy mount namespace.

> >>> + */
> >>> + if (current->nsproxy == NULL)
> >>> + return;
> >>> +
> >>> + nodename = utsname()->nodename;
> >>>   clnt->cl_nodelen = strlen(nodename);
> >>>   if (clnt->cl_nodelen > UNX_MAXNODENAME)
> >>>   clnt->cl_nodelen = UNX_MAXNODENAME;
> >>> @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct rpc_create_args
*args, stru
> >>>   }
> >>>
> >>>   /* save the nodename */
> >>> - rpc_clnt_set_nodename(clnt, utsname()->nodename);
> >>> + rpc_clnt_set_nodename(clnt);
> >>>   rpc_register_client(clnt);
> >>>   return clnt;
> >>>
> >>> @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
> >>>   err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
> >>>   if (err != 0)
> >>>   goto out_no_path;
> >>> - rpc_clnt_set_nodename(new, utsname()->nodename);
> >>> + rpc_clnt_set_nodename(new);
> >>>   if (new->cl_auth)
> >>>   atomic_inc(&new->cl_auth->au_count);
> >>>   atomic_inc(&clnt->cl_count);
> >>>
> >>> --
> >>> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
> >>> the body of a message to majordomo@vger.kernel.org
> >>> More majordomo info at http://vger.kernel.org/majordomo-info.html
> >> Acked-by: Jeff Layton <jlayton@redhat.com>
> > OK, colour me confused (again).
>
> What color?
>
> > Why should a umount trigger an
> > rpc_create() or rpc_clone_client()?
>
> It calls nsm_create().
> Here is the trace (https://bugzilla.redhat.com/show_bug.cgi?id=830862,
> comment 68):

Right, but if we're using NFSv3 lock monitoring, we know in advance that
we're going to need an nsm call to localhost. Why can't we just cache
the one that we used to start lock monitoring in the first place?

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Stanislav Kinsbursky on Mon, 10 Sep 2012 08:43:31 GMT
View Forum Message <> Reply to Message

> On Sat, 2012-09-08 at 08:59 +0300, Stanislav Kinsbursky wrote:

>>> On Mon, 2012-08-13 at 08:10 -0400, Jeff Layton wrote:
>>>> On Mon, 13 Aug 2012 15:37:31 +0400
>>>> Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:
>>>>
>>>>> v2:
>>>>> 1) rpc_clnt_set_nodename() prototype updated.
>>>>> 2) fixed errors in comment.
>>>>>
>>>>> When child reaper exits, it can destroy mount namespace it belongs to, and if
>>>>> there are NFS mounts inside, then it will try to umount them. But in this
>>>>> point current->nsproxy is set to NULL and all namespaces will be destroyed one
>>>>> by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
>>>>>
>>>>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>>>>> ---
>>>>> net/sunrpc/clnt.c | 16 +++++++++++++---
>>>>> 1 files changed, 13 insertions(+), 3 deletions(-)
>>>>>
>>>>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
>>>>> index 9a9676e..8fbcbc8 100644
>>>>> --- a/net/sunrpc/clnt.c
>>>>> +++ b/net/sunrpc/clnt.c
>>>>> @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
>>>>> return rpc_pipefs_notifier_unregister(&rpc_clients_block);
>>>>> }
>>>>>
>>>>> -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
>>>>> +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)
>>>>> {
>>>>> + const char *nodename;
>>>>> +

```
>>>>> + /*
>>>>> +  * We have to protect against dying child reaper, which has released
>>>>> +  * its nsproxy already and is trying to destroy mount namespace.
>>>>> +  */
>>>>> + if (current->nsproxy == NULL)
>>>>> +  return;
>>>>> +
>>>>> + nodename = utsname()->nodename;
>>>>>      clnt->cl_nodelen = strlen(nodename);
>>>>>      if (clnt->cl_nodelen > UNX_MAXNODENAME)
>>>>>       clnt->cl_nodelen = UNX_MAXNODENAME;
>>>>> @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct
rpc_create_args *args, stru
>>>>>      }
>>>>>
>>>>>      /* save the nodename */
>>>>> - rpc_clnt_set_nodename(clnt, utsname()->nodename);
>>>>> + rpc_clnt_set_nodename(clnt);
>>>>>      rpc_register_client(clnt);
>>>>>      return clnt;
>>>>>
>>>>> @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
>>>>>      err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
>>>>>      if (err != 0)
>>>>>       goto out_no_path;
>>>>> - rpc_clnt_set_nodename(new, utsname()->nodename);
>>>>> + rpc_clnt_set_nodename(new);
>>>>>      if (new->cl_auth)
>>>>>       atomic_inc(&new->cl_auth->au_count);
>>>>>      atomic_inc(&clnt->cl_count);
>>>>>
>>>>> --
>>>>> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
>>>>> the body of a message to majordomo@vger.kernel.org
>>>>> More majordomo info at  http://vger.kernel.org/majordomo-info.html
>>>> Acked-by: Jeff Layton <jlayton@redhat.com>
>>> OK, colour me confused (again).
>>
>> What color?
>>
>>> Why should a umount trigger an
>>> rpc_create() or rpc_clone_client()?
>>
>> It calls nsm_create().
>> Here is the trace (https://bugzilla.redhat.com/show_bug.cgi?id=830862,
>> comment 68):
>
> Right, but if we're using NFSv3 lock monitoring, we know in advance that
```

> we're going to need an nsm call to localhost. Why can't we just cache
> the one that we used to start lock monitoring in the first place?
>

Do you suggest to cache the call or the client for the call?

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Myklebust, Trond on Mon, 10 Sep 2012 15:27:20 GMT
View Forum Message <> Reply to Message

On Mon, 2012-09-10 at 12:43 +0400, Stanislav Kinsbursky wrote:

> > On Sat, 2012-09-08 at 08:59 +0300, Stanislav Kinsbursky wrote:

> >>> On Mon, 2012-08-13 at 08:10 -0400, Jeff Layton wrote:
> >>>> On Mon, 13 Aug 2012 15:37:31 +0400
> >>>> Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:
> >>>>
> >>>>> v2:
> >>>>> 1) rpc_clnt_set_nodename() prototype updated.
> >>>>> 2) fixed errors in comment.
> >>>>>
> >>>>> When child reaper exits, it can destroy mount namespace it belongs to, and if
> >>>>> there are NFS mounts inside, then it will try to umount them. But in this
> >>>>> point current->nsproxy is set to NULL and all namespaces will be destroyed one
> >>>>> by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
> >>>>>
> >>>>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> >>>>> ---
> >>>>> net/sunrpc/clnt.c | 16 +++++++++++++---
> >>>>> 1 files changed, 13 insertions(+), 3 deletions(-)
> >>>>>
> >>>>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
> >>>>> index 9a9676e..8fbcbc8 100644
> >>>>> --- a/net/sunrpc/clnt.c
> >>>>> +++ b/net/sunrpc/clnt.c
> >>>>> @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
> >>>>> 	return rpc_pipefs_notifier_unregister(&rpc_clients_block);
> >>>>> }
> >>>>>
> >>>>> -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
> >>>>> +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)

> >>>>>   {
> >>>>> + const char *nodename;
> >>>>> +
> >>>>> + /*
> >>>>> + * We have to protect against dying child reaper, which has released
> >>>>> + * its nsproxy already and is trying to destroy mount namespace.
> >>>>> + */
> >>>>> + if (current->nsproxy == NULL)
> >>>>> + return;
> >>>>> +
> >>>>> + nodename = utsname()->nodename;
> >>>>>   clnt->cl_nodelen = strlen(nodename);
> >>>>>   if (clnt->cl_nodelen > UNX_MAXNODENAME)
> >>>>>   clnt->cl_nodelen = UNX_MAXNODENAME;
> >>>>> @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct
rpc_create_args *args, stru
> >>>>>   }
> >>>>>
> >>>>>   /* save the nodename */
> >>>>> - rpc_clnt_set_nodename(clnt, utsname()->nodename);
> >>>>> + rpc_clnt_set_nodename(clnt);
> >>>>>   rpc_register_client(clnt);
> >>>>>   return clnt;
> >>>>>
> >>>>> @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
> >>>>>   err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
> >>>>>   if (err != 0)
> >>>>>   goto out_no_path;
> >>>>> - rpc_clnt_set_nodename(new, utsname()->nodename);
> >>>>> + rpc_clnt_set_nodename(new);
> >>>>>   if (new->cl_auth)
> >>>>>   atomic_inc(&new->cl_auth->au_count);
> >>>>>   atomic_inc(&clnt->cl_count);
> >>>>>
> >>>>> --
> >>>>> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
> >>>>> the body of a message to majordomo@vger.kernel.org
> >>>>> More majordomo info at http://vger.kernel.org/majordomo-info.html
> >>>> Acked-by: Jeff Layton <jlayton@redhat.com>
> >>> OK, colour me confused (again).
> >>
> >> What color?
> >>
> >>> Why should a umount trigger an
> >>> rpc_create() or rpc_clone_client()?
> >>
> >> It calls nsm_create().
> >> Here is the trace (https://bugzilla.redhat.com/show_bug.cgi?id=830862,

> >> comment 68):
> >
> > Right, but if we're using NFSv3 lock monitoring, we know in advance that
> > we're going to need an nsm call to localhost. Why can't we just cache
> > the one that we used to start lock monitoring in the first place?
> >
>
> Do you suggest to cache the call or the client for the call?

Hi Stanislav,

Sorry, I agree that the above was unclear. My intention was to suggest
that we should cache a reference to the rpc client that we used to
connect to rpc.statd when initiating lock monitoring.

Basically, I'm suggesting that we should do something similar to the
rpcbind rpc_client caching scheme in net/sunrpc/rpcb_clnt.c.

Cheers
  Trond

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Stanislav Kinsbursky on Mon, 10 Sep 2012 15:37:20 GMT
View Forum Message <> Reply to Message

> On Mon, 2012-09-10 at 12:43 +0400, Stanislav Kinsbursky wrote:

>>> On Sat, 2012-09-08 at 08:59 +0300, Stanislav Kinsbursky wrote:

>>>>> On Mon, 2012-08-13 at 08:10 -0400, Jeff Layton wrote:
>>>>>> On Mon, 13 Aug 2012 15:37:31 +0400
>>>>>> Stanislav Kinsbursky <skinsbursky@parallels.com> wrote:
>>>>>>
>>>>>>> v2:
>>>>>>> 1) rpc_clnt_set_nodename() prototype updated.
>>>>>>> 2) fixed errors in comment.
>>>>>>>

>>>>>>> When child reaper exits, it can destroy mount namespace it belongs to, and if
>>>>>>> there are NFS mounts inside, then it will try to umount them. But in this
>>>>>>> point current->nsproxy is set to NULL and all namespaces will be destroyed one
>>>>>>> by one. I.e. we can't dereference current->nsproxy to obtain uts namespace.
>>>>>>>
>>>>>>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>>>>>>> ---
>>>>>>> net/sunrpc/clnt.c | 16 ++++++++++++---
>>>>>>> 1 files changed, 13 insertions(+), 3 deletions(-)
>>>>>>>
>>>>>>> diff --git a/net/sunrpc/clnt.c b/net/sunrpc/clnt.c
>>>>>>> index 9a9676e..8fbcbc8 100644
>>>>>>> --- a/net/sunrpc/clnt.c
>>>>>>> +++ b/net/sunrpc/clnt.c
>>>>>>> @@ -277,8 +277,18 @@ void rpc_clients_notifier_unregister(void)
>>>>>>> return rpc_pipefs_notifier_unregister(&rpc_clients_block);
>>>>>>> }
>>>>>>>
>>>>>>> -static void rpc_clnt_set_nodename(struct rpc_clnt *clnt, const char *nodename)
>>>>>>> +static void rpc_clnt_set_nodename(struct rpc_clnt *clnt)
>>>>>>> {
>>>>>>> + const char *nodename;
>>>>>>> +
>>>>>>> + /*
>>>>>>> + * We have to protect against dying child reaper, which has released
>>>>>>> + * its nsproxy already and is trying to destroy mount namespace.
>>>>>>> + */
>>>>>>> + if (current->nsproxy == NULL)
>>>>>>> + return;
>>>>>>> +
>>>>>>> + nodename = utsname()->nodename;
>>>>>>> clnt->cl_nodelen = strlen(nodename);
>>>>>>> if (clnt->cl_nodelen > UNX_MAXNODENAME)
>>>>>>> clnt->cl_nodelen = UNX_MAXNODENAME;
>>>>>>> @@ -365,7 +375,7 @@ static struct rpc_clnt * rpc_new_client(const struct
rpc_create_args *args, stru
>>>>>>> }
>>>>>>>
>>>>>>> /* save the nodename */
>>>>>>> - rpc_clnt_set_nodename(clnt, utsname()->nodename);
>>>>>>> + rpc_clnt_set_nodename(clnt);
>>>>>>> rpc_register_client(clnt);
>>>>>>> return clnt;
>>>>>>>
>>>>>>> @@ -524,7 +534,7 @@ rpc_clone_client(struct rpc_clnt *clnt)
>>>>>>> err = rpc_setup_pipedir(new, clnt->cl_program->pipe_dir_name);
>>>>>>> if (err != 0)
>>>>>>> goto out_no_path;

>>>>>>> - rpc_clnt_set_nodename(new, utsname()->nodename);
>>>>>>> + rpc_clnt_set_nodename(new);
>>>>>>> 	if (new->cl_auth)
>>>>>>> 	 atomic_inc(&new->cl_auth->au_count);
>>>>>>> 	atomic_inc(&clnt->cl_count);
>>>>>>>
>>>>>>> --
>>>>>>> To unsubscribe from this list: send the line "unsubscribe linux-nfs" in
>>>>>>> the body of a message to majordomo@vger.kernel.org
>>>>>>> More majordomo info at http://vger.kernel.org/majordomo-info.html
>>>>>> Acked-by: Jeff Layton <jlayton@redhat.com>
>>>>> OK, colour me confused (again).
>>>>
>>>> What color?
>>>>
>>>>> Why should a umount trigger an
>>>>> rpc_create() or rpc_clone_client()?
>>>>
>>>> It calls nsm_create().
>>>> Here is the trace (https://bugzilla.redhat.com/show_bug.cgi?id=830862,
>>>> comment 68):
>>>
>>> Right, but if we're using NFSv3 lock monitoring, we know in advance that
>>> we're going to need an nsm call to localhost. Why can't we just cache
>>> the one that we used to start lock monitoring in the first place?
>>>
>>
>> Do you suggest to cache the call or the client for the call?
>
> Hi Stanislav,
>
> Sorry, I agree that the above was unclear. My intention was to suggest
> that we should cache a reference to the rpc client that we used to
> connect to rpc.statd when initiating lock monitoring.
>
> Basically, I'm suggesting that we should do something similar to the
> rpcbind rpc_client caching scheme in net/sunrpc/rpcb_clnt.c.
>

Hi, Trond.
So, if I understand you right, we can create rpc client (or increase usage
counter) on NSMPROC_MON call and destroy (or decrease usage counter) on
NSMPROC_UNMON call.
Will this solution works?

> Cheers
>   Trond
>

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Myklebust, Trond on Mon, 10 Sep 2012 15:41:57 GMT
View Forum Message <> Reply to Message

On Mon, 2012-09-10 at 19:37 +0400, Stanislav Kinsbursky wrote:
> Hi, Trond.
> So, if I understand you right, we can create rpc client (or increase usage
> counter) on NSMPROC_MON call and destroy (or decrease usage counter) on
> NSMPROC_UNMON call.
> Will this solution works?

The rpc client(s) will need to be per-net-namespace, which complicates
matters a little bit, but yes, creation at NSMPROC_MON, and destruction
at NSMPROC_UNMON should work.

--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Stanislav Kinsbursky on Mon, 10 Sep 2012 15:55:34 GMT
View Forum Message <> Reply to Message

> On Mon, 2012-09-10 at 19:37 +0400, Stanislav Kinsbursky wrote:
>> Hi, Trond.
>> So, if I understand you right, we can create rpc client (or increase usage
>> counter) on NSMPROC_MON call and destroy (or decrease usage counter) on
>> NSMPROC_UNMON call.
>> Will this solution works?
>
> The rpc client(s) will need to be per-net-namespace, which complicates
> matters a little bit, but yes, creation at NSMPROC_MON, and destruction

> at NSMPROC_UNMON should work.
>

Not really. We already have per-net Lockd data. So, adding one more
reference-counted rpc client doesn't look that complicated.
Ok, thanks. I'll try to implement this.

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Stanislav Kinsbursky on Thu, 13 Sep 2012 12:11:08 GMT
View Forum Message <> Reply to Message

> On Mon, 2012-09-10 at 19:37 +0400, Stanislav Kinsbursky wrote:
>> Hi, Trond.
>> So, if I understand you right, we can create rpc client (or increase usage
>> counter) on NSMPROC_MON call and destroy (or decrease usage counter) on
>> NSMPROC_UNMON call.
>> Will this solution works?
>
> The rpc client(s) will need to be per-net-namespace, which complicates
> matters a little bit, but yes, creation at NSMPROC_MON, and destruction
> at NSMPROC_UNMON should work.
>

Hi, Trond.
With reference-counted NSM client I got this:

BUG: unable to handle kernel NULL pointer dereference at 0000000000000008
IP: [<ffffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]
PGD 0
Oops: 0000 [#1] SMP DEBUG_PAGEALLOC
Modules linked in: nfsv3 nfs_acl nfs lockd veth sunrpc bridge stp llc i2c_piix4
i2c_core virtio_blk virtio_net floppy virtio_pci virtio_ring virtio
CPU 1
Pid: 1174, comm: bash Not tainted 3.5.0-kitchensink+ #44 Bochs Bochs
RIP: 0010:[<ffffffffa00c0d7f>] [<ffffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]
RSP: 0018:ffff88001d0f1a08 EFLAGS: 00010246
RAX: 0000000000000000 RBX: ffff88001d0f1c38 RCX: 0000000000000000
RDX: ffff88001c85803f RSI: ffff88001d23b204 RDI: ffff88001d0f1a48
RBP: ffff88001d0f1a18 R08: ffff88001c858040 R09: 0000000000000003
R10: ffff88001a5aba10 R11: ffff88001d0f1ac8 R12: ffff88001d0f1a48
R13: ffff88001a8a3138 R14: ffffffffa008d580 R15: ffffffffa00c0db5

FS: 00007f0d60eea700(0000) GS:ffff88001f700000(0000) knlGS:0000000000000000
CS: 0010 DS: 0000 ES: 0000 CR0: 000000008005003b
CR2: 0000000000000008 CR3: 000000001db3d000 CR4: 00000000000006e0
DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400
Process bash (pid: 1174, threadinfo ffff88001d0f0000, task ffff88001d1f4160)
Stack:
 ffff88001d0f1a48 ffff88001c858030 ffff88001d0f1a28 ffffffffa00c0dc9
 ffff88001d0f1ab8 ffffffffa00731a0 ffff88001a5aba10 ffff88001d0f1c38
 ffff88001c858040 ffff88001a8a3140 ffff88001c858854 ffff88001a8a3140
Call Trace:
 [<ffffffffa00c0dc9>] nsm_xdr_enc_unmon+0x14/0x16 [lockd]
 [<ffffffffa00731a0>] rpcauth_wrap_req+0xa1/0xb2 [sunrpc]
 [<ffffffffa006b83f>] ? xprt_prepare_transmit+0x83/0x93 [sunrpc]
 [<ffffffffa0068e54>] call_transmit+0x1e4/0x263 [sunrpc]
 [<ffffffffa00728e2>] __rpc_execute+0xe7/0x313 [sunrpc]
 [<ffffffffa0068c70>] ? call_transmit_status+0xb8/0xb8 [sunrpc]
 [<ffffffff81055ed9>] ? wake_up_bit+0x25/0x2a
 [<ffffffffa0072be0>] rpc_execute+0x9d/0xa5 [sunrpc]
 [<ffffffffa006a6ae>] rpc_run_task+0x7e/0x86 [sunrpc]
 [<ffffffffa006a7a3>] rpc_call_sync+0x44/0x65 [sunrpc]
 [<ffffffffa00c0883>] nsm_mon_unmon+0x81/0xad [lockd]
 [<ffffffffa00c0931>] nsm_unmonitor+0x82/0x13a [lockd]
 [<ffffffffa00bc251>] nlm_destroy_host_locked+0x93/0xc9 [lockd]
 [<ffffffffa00bc33a>] nlmclnt_release_host+0xb3/0xc3 [lockd]
 [<ffffffffa00ba09f>] nlmclnt_done+0x1a/0x26 [lockd]
 [<ffffffffa00d583e>] nfs_destroy_server+0x24/0x26 [nfs]
 [<ffffffffa00d5d85>] nfs_free_server+0xad/0x134 [nfs]
 [<ffffffffa00dd5ff>] nfs_kill_super+0x22/0x26 [nfs]
 [<ffffffff8112b278>] deactivate_locked_super+0x26/0x57
 [<ffffffff8112bd89>] deactivate_super+0x45/0x4c
 [<ffffffff811423ec>] mntput_no_expire+0x110/0x119
 [<ffffffff8114241f>] mntput+0x2a/0x2c
 [<ffffffff81142605>] release_mounts+0x72/0x84
 [<ffffffff811427cf>] put_mnt_ns+0x6f/0x7e
 [<ffffffff8105a3db>] free_nsproxy+0x1f/0x87
 [<ffffffff8105a49f>] switch_task_namespaces+0x5c/0x65
 [<ffffffff8105a4b8>] exit_task_namespaces+0x10/0x12
 [<ffffffff8103c232>] do_exit+0x69b/0x84f
 [<ffffffff8103c469>] do_group_exit+0x83/0xae
 [<ffffffff8103c4ab>] sys_exit_group+0x17/0x1b
 [<ffffffff814657e9>] system_call_fastpath+0x16/0x1b
Code: e5 41 54 53 66 66 66 66 90 48 89 f3 49 89 fc 48 8b 76 18 e8 93 ff ff ff 4c
89 e7 65 48 8b 04 25 c0 b9 00 00 48 8b 80 00 05 00 00 <48> 8b 70 08 48 83 c6 45
e8 73 ff ff ff 4c 89 e7 be 0c 00 00 00
RIP [<ffffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]

There are more places, where NFS and Lockd code dereference utsname().
In XDR layer, for instance.

So, probably, we have to tie NFS to utsns as well as to netns.
Add one more ns to xprt? What do you think?

--
Best regards,
Stanislav Kinsbursky

Subject: Re: [PATCH v2] SUNRPC: check current nsproxy before set of node name on client creation
Posted by Myklebust, Trond on Thu, 13 Sep 2012 13:30:25 GMT
View Forum Message <> Reply to Message

On Thu, 2012-09-13 at 16:11 +0400, Stanislav Kinsbursky wrote:

> > On Mon, 2012-09-10 at 19:37 +0400, Stanislav Kinsbursky wrote:
> >> Hi, Trond.
> >> So, if I understand you right, we can create rpc client (or increase usage
> >> counter) on NSMPROC_MON call and destroy (or decrease usage counter) on
> >> NSMPROC_UNMON call.
> >> Will this solution works?
> >
> > The rpc client(s) will need to be per-net-namespace, which complicates
> > matters a little bit, but yes, creation at NSMPROC_MON, and destruction
> > at NSMPROC_UNMON should work.
> >
>
> Hi, Trond.
> With reference-counted NSM client I got this:
>
> BUG: unable to handle kernel NULL pointer dereference at 0000000000000008
> IP: [<ffffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]
> PGD 0
> Oops: 0000 [#1] SMP DEBUG_PAGEALLOC
> Modules linked in: nfsv3 nfs_acl nfs lockd veth sunrpc bridge stp llc i2c_piix4
> i2c_core virtio_blk virtio_net floppy virtio_pci virtio_ring virtio
> CPU 1
> Pid: 1174, comm: bash Not tainted 3.5.0-kitchensink+ #44 Bochs Bochs
> RIP: 0010:[<ffffffffa00c0d7f>] [<ffffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]
> RSP: 0018:ffff88001d0f1a08 EFLAGS: 00010246
> RAX: 0000000000000000 RBX: ffff88001d0f1c38 RCX: 0000000000000000
> RDX: ffff88001c85803f RSI: ffff88001d23b204 RDI: ffff88001d0f1a48
> RBP: ffff88001d0f1a18 R08: ffff88001c858040 R09: 0000000000000003
> R10: ffff88001a5aba10 R11: ffff88001d0f1ac8 R12: ffff88001d0f1a48
> R13: ffff88001a8a3138 R14: ffffffffa008d580 R15: ffffffffa00c0db5

> FS: 00007f0d60eea700(0000) GS:ffff88001f700000(0000) knlGS:0000000000000000
> CS: 0010 DS: 0000 ES: 0000 CR0: 000000008005003b
> CR2: 0000000000000008 CR3: 000000001db3d000 CR4: 00000000000006e0
> DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
> DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400
> Process bash (pid: 1174, threadinfo ffff88001d0f0000, task ffff88001d1f4160)
> Stack:
> ffff88001d0f1a48 ffff88001c858030 ffff88001d0f1a28 ffffffffa00c0dc9
> ffff88001d0f1ab8 ffffffffa00731a0 ffff88001a5aba10 ffff88001d0f1c38
> ffff88001c858040 ffff88001a8a3140 ffff88001c858854 ffff88001a8a3140
> Call Trace:
> [<ffffffffa00c0dc9>] nsm_xdr_enc_unmon+0x14/0x16 [lockd]
> [<ffffffffa00731a0>] rpcauth_wrap_req+0xa1/0xb2 [sunrpc]
> [<ffffffffa006b83f>] ? xprt_prepare_transmit+0x83/0x93 [sunrpc]
> [<ffffffffa0068e54>] call_transmit+0x1e4/0x263 [sunrpc]
> [<ffffffffa00728e2>] __rpc_execute+0xe7/0x313 [sunrpc]
> [<ffffffffa0068c70>] ? call_transmit_status+0xb8/0xb8 [sunrpc]
> [<ffffffff81055ed9>] ? wake_up_bit+0x25/0x2a
> [<ffffffffa0072be0>] rpc_execute+0x9d/0xa5 [sunrpc]
> [<ffffffffa006a6ae>] rpc_run_task+0x7e/0x86 [sunrpc]
> [<ffffffffa006a7a3>] rpc_call_sync+0x44/0x65 [sunrpc]
> [<ffffffffa00c0883>] nsm_mon_unmon+0x81/0xad [lockd]
> [<ffffffffa00c0931>] nsm_unmonitor+0x82/0x13a [lockd]
> [<ffffffffa00bc251>] nlm_destroy_host_locked+0x93/0xc9 [lockd]
> [<ffffffffa00bc33a>] nlmclnt_release_host+0xb3/0xc3 [lockd]
> [<ffffffffa00ba09f>] nlmclnt_done+0x1a/0x26 [lockd]
> [<ffffffffa00d583e>] nfs_destroy_server+0x24/0x26 [nfs]
> [<ffffffffa00d5d85>] nfs_free_server+0xad/0x134 [nfs]
> [<ffffffffa00dd5ff>] nfs_kill_super+0x22/0x26 [nfs]
> [<ffffffff8112b278>] deactivate_locked_super+0x26/0x57
> [<ffffffff8112bd89>] deactivate_super+0x45/0x4c
> [<ffffffff811423ec>] mntput_no_expire+0x110/0x119
> [<ffffffff8114241f>] mntput+0x2a/0x2c
> [<ffffffff81142605>] release_mounts+0x72/0x84
> [<ffffffff811427cf>] put_mnt_ns+0x6f/0x7e
> [<ffffffff8105a3db>] free_nsproxy+0x1f/0x87
> [<ffffffff8105a49f>] switch_task_namespaces+0x5c/0x65
> [<ffffffff8105a4b8>] exit_task_namespaces+0x10/0x12
> [<ffffffff8103c232>] do_exit+0x69b/0x84f
> [<ffffffff8103c469>] do_group_exit+0x83/0xae
> [<ffffffff8103c4ab>] sys_exit_group+0x17/0x1b
> [<ffffffff814657e9>] system_call_fastpath+0x16/0x1b
> Code: e5 41 54 53 66 66 66 66 90 48 89 f3 49 89 fc 48 8b 76 18 e8 93 ff ff ff 4c
> 89 e7 65 48 8b 04 25 c0 b9 00 00 48 8b 80 00 05 00 00 <48> 8b 70 08 48 83 c6 45
> e8 73 ff ff ff 4c 89 e7 be 0c 00 00 00
> RIP [<ffffffffa00c0d7f>] encode_mon_id+0x2e/0x64 [lockd]
>
>

> There are more places, where NFS and Lockd code dereference utsname().
> In XDR layer, for instance.
>
> So, probably, we have to tie NFS to utsns as well as to netns.
> Add one more ns to xprt? What do you think?
>

We've already saved the utsname in the rpc_client as clnt->cl_nodename.
All XDR users can be trivially converted to use that.

Cheers
  Trond
--
Trond Myklebust
Linux NFS client maintainer

NetApp
Trond.Myklebust@netapp.com
www.netapp.com