Subject: [PATCH] SUNRPC: return negative value in case rpcbind client creation error
Posted by Stanislav Kinsbursky on Fri, 20 Jul 2012 11:57:48 GMT
View Forum Message <> Reply to Message

Without this patch kernel will panic on LockD start, because lockd_up() checks
lockd_up_net() result for negative value.
>From my pow it's better to return negative value from rpcbind routines instead
of replacing all such checks like in lockd_up().

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
---
 net/sunrpc/rpcb_clnt.c | 4 ++--
 1 files changed, 2 insertions(+), 2 deletions(-)

diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
index 92509ff..a70acae 100644
--- a/net/sunrpc/rpcb_clnt.c
+++ b/net/sunrpc/rpcb_clnt.c
@@ -251,7 +251,7 @@ static int rpcb_create_local_unix(struct net *net)
  if (IS_ERR(clnt)) {
   dprintk("RPC: failed to create AF_LOCAL rpcbind "
     "client (errno %ld).\n", PTR_ERR(clnt));
- result = -PTR_ERR(clnt);
+ result = PTR_ERR(clnt);
   goto out;
  }

@@ -298,7 +298,7 @@ static int rpcb_create_local_net(struct net *net)
  if (IS_ERR(clnt)) {
   dprintk("RPC: failed to create local rpcbind "
     "client (errno %ld).\n", PTR_ERR(clnt));
- result = -PTR_ERR(clnt);
+ result = PTR_ERR(clnt);
   goto out;
  }

Subject: Re: [PATCH] SUNRPC: return negative value in case rpcbind client creation error
Posted by Myklebust, Trond on Mon, 30 Jul 2012 23:12:05 GMT
View Forum Message <> Reply to Message

On Fri, 2012-07-20 at 15:57 +0400, Stanislav Kinsbursky wrote:
> Without this patch kernel will panic on LockD start, because lockd_up() checks
> lockd_up_net() result for negative value.
> >From my pow it's better to return negative value from rpcbind routines instead
> of replacing all such checks like in lockd_up().

>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> ---
> net/sunrpc/rpcb_clnt.c | 4 ++--
> 1 files changed, 2 insertions(+), 2 deletions(-)
>
> diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
> index 92509ff..a70acae 100644
> --- a/net/sunrpc/rpcb_clnt.c
> +++ b/net/sunrpc/rpcb_clnt.c
> @@ -251,7 +251,7 @@ static int rpcb_create_local_unix(struct net *net)
>   if (IS_ERR(clnt)) {
>   dprintk("RPC: failed to create AF_LOCAL rpcbind "
>   "client (errno %ld).\n", PTR_ERR(clnt));
> - result = -PTR_ERR(clnt);
> + result = PTR_ERR(clnt);
>   goto out;
>   }
>
> @@ -298,7 +298,7 @@ static int rpcb_create_local_net(struct net *net)
>   if (IS_ERR(clnt)) {
>   dprintk("RPC: failed to create local rpcbind "
>   "client (errno %ld).\n", PTR_ERR(clnt));
> - result = -PTR_ERR(clnt);
> + result = PTR_ERR(clnt);
>   goto out;
>   }

Who is supposed to carry this patch? Is it Bruce or is it me?

Cheers
  Trond

Subject: Re: [PATCH] SUNRPC: return negative value in case rpcbind client creation error
Posted by bfields on Mon, 30 Jul 2012 23:23:16 GMT
View Forum Message <> Reply to Message

On Mon, Jul 30, 2012 at 11:12:05PM +0000, Myklebust, Trond wrote:
> On Fri, 2012-07-20 at 15:57 +0400, Stanislav Kinsbursky wrote:
> > Without this patch kernel will panic on LockD start, because lockd_up() checks
> > lockd_up_net() result for negative value.
> > >From my pow it's better to return negative value from rpcbind routines instead
> > of replacing all such checks like in lockd_up().
> >
> > Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
> > ---

> > net/sunrpc/rpcb_clnt.c | 4 ++--
> > 1 files changed, 2 insertions(+), 2 deletions(-)
> >
> > diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
> > index 92509ff..a70acae 100644
> > --- a/net/sunrpc/rpcb_clnt.c
> > +++ b/net/sunrpc/rpcb_clnt.c
> > @@ -251,7 +251,7 @@ static int rpcb_create_local_unix(struct net *net)
> >   if (IS_ERR(clnt)) {
> >   dprintk("RPC: failed to create AF_LOCAL rpcbind "
> >   "client (errno %ld).\n", PTR_ERR(clnt));
> > - result = -PTR_ERR(clnt);
> > + result = PTR_ERR(clnt);
> >   goto out;
> >   }
> >
> > @@ -298,7 +298,7 @@ static int rpcb_create_local_net(struct net *net)
> >   if (IS_ERR(clnt)) {
> >   dprintk("RPC: failed to create local rpcbind "
> >   "client (errno %ld).\n", PTR_ERR(clnt));
> > - result = -PTR_ERR(clnt);
> > + result = PTR_ERR(clnt);
> >   goto out;
> >   }
>
> Who is supposed to carry this patch? Is it Bruce or is it me?

Works either way. Either way--it looks like the bug was introduced with

c526611dd631b2802b6b0221ffb306c5fa25c86c "SUNRPC: Use a cached RPC client and transport for rpcbind upcalls" and 7402ab19cdd5943c7dd4f3399afe3abda8077ef5 "SUNRPC: Use AF_LOCAL for rpcbind upcalls"

and should go to stable as well.

(Looks like I said that before but accidentally dropped everyone off the cc.)

--b.

Subject: Re: [PATCH] SUNRPC: return negative value in case rpcbind client creation error
Posted by Stanislav Kinsbursky on Tue, 31 Jul 2012 07:46:33 GMT
View Forum Message <> Reply to Message

> On Fri, 2012-07-20 at 15:57 +0400, Stanislav Kinsbursky wrote:
>> Without this patch kernel will panic on LockD start, because lockd_up() checks
>> lockd_up_net() result for negative value.
>> >From my pow it's better to return negative value from rpcbind routines instead
>> of replacing all such checks like in lockd_up().
>>
>> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>> ---
>> net/sunrpc/rpcb_clnt.c | 4 ++--
>> 1 files changed, 2 insertions(+), 2 deletions(-)
>>
>> diff --git a/net/sunrpc/rpcb_clnt.c b/net/sunrpc/rpcb_clnt.c
>> index 92509ff..a70acae 100644
>> --- a/net/sunrpc/rpcb_clnt.c
>> +++ b/net/sunrpc/rpcb_clnt.c
>> @@ -251,7 +251,7 @@ static int rpcb_create_local_unix(struct net *net)
>>   if (IS_ERR(clnt)) {
>>   dprintk("RPC: failed to create AF_LOCAL rpcbind "
>>   "client (errno %ld).\n", PTR_ERR(clnt));
>> - result = -PTR_ERR(clnt);
>> + result = PTR_ERR(clnt);
>>   goto out;
>>   }
>>
>> @@ -298,7 +298,7 @@ static int rpcb_create_local_net(struct net *net)
>>   if (IS_ERR(clnt)) {
>>   dprintk("RPC: failed to create local rpcbind "
>>   "client (errno %ld).\n", PTR_ERR(clnt));
>> - result = -PTR_ERR(clnt);
>> + result = PTR_ERR(clnt);
>>   goto out;
>>   }
>
> Who is supposed to carry this patch? Is it Bruce or is it me?
>

I don't know, Trond. It's up to you and Bruce.
This is a bug fix and the bug is very old. The only reason, why it was found
just now, is that all the callers of these functions were checking the result
for zero.
And I agreed with Bruce, that is have to marked for stable branches (at least
for 3.4-3.5 kernels).

--
Best regards,
Stanislav Kinsbursky