Subject: Re: [PATCH] SUNRPC: set desired file system root before connecting local transports
Posted by Myklebust, Trond on Wed, 07 Mar 2012 00:19:48 GMT
View Forum Message <> Reply to Message

On Wed, 2012-02-29 at 18:59 +0400, Stanislav Kinsbursky wrote:
> Today, there is a problem in connecting of local SUNRPC thansports. These
> transports uses UNIX sockets and connection itself is done by rpciod workqueue.
> But UNIX sockets lookup is done in context of process file system root. I.e.
> all local thunsports are connecting in rpciod context.
> This works nice until we will try to mount NFS from process with other root -
> for example in container. This container can have it's own (nested) root and
> rcpbind process, listening on it's own unix sockets. But NFS mount attempt in
> this container will register new service (Lockd for example) in global rpcbind
> - not containers's one.
> This patch solves the problem by switching rpciod kernel thread's file system
> root to right one (stored on transport) while connecting of local transports.
>
> Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>
>
> ---
> fs/fs_struct.c | 1 +
> net/sunrpc/xprtsock.c | 32 ++++++++++++++++++++++++++++++--
> 2 files changed, 31 insertions(+), 2 deletions(-)
>
> diff --git a/fs/fs_struct.c b/fs/fs_struct.c
> index 78b519c..0f984c3 100644
> --- a/fs/fs_struct.c
> +++ b/fs/fs_struct.c
> @@ -36,6 +36,7 @@ void set_fs_root(struct fs_struct *fs, struct path *path)
>   if (old_root.dentry)
>   path_put_longterm(&old_root);
> }
> +EXPORT_SYMBOL_GPL(set_fs_root);
>
> /*
>   * Replace the fs->{pwdmnt,pwd} with {mnt,dentry}. Put the old values.
> diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c
> index 4c8281d..c94c181 100644
> --- a/net/sunrpc/xprtsock.c
> +++ b/net/sunrpc/xprtsock.c
> @@ -37,6 +37,7 @@
> #include <linux/sunrpc/svcsock.h>
> #include <linux/sunrpc/xprtsock.h>
> #include <linux/file.h>
> +#include <linux/fs_struct.h>
> #ifdef CONFIG_SUNRPC_BACKCHANNEL
> #include <linux/sunrpc/bc_xprt.h>

```
>  #endif
> @@ -255,6 +256,11 @@ struct sock_xprt {
>    void   (*old_state_change)(struct sock *);
>    void   (*old_write_space)(struct sock *);
>    void   (*old_error_report)(struct sock *);
> +
> + /*
> +  * Saved transport creator root. Required for local transports only.
> +  */
> + struct path  root;
>  };
>
>  /*
> @@ -1891,6 +1897,7 @@ static void xs_local_setup_socket(struct work_struct *work)
>    struct rpc_xprt *xprt = &transport->xprt;
>    struct socket *sock;
>    int status = -EIO;
> + struct path root;
>
>    if (xprt->shutdown)
>     goto out;
> @@ -1908,7 +1915,14 @@ static void xs_local_setup_socket(struct work_struct *work)
>    dprintk("RPC:       worker connecting xprt %p via AF_LOCAL to %s\n",
>      xprt, xprt->address_strings[RPC_DISPLAY_ADDR]);
>
> + get_fs_root(current->fs, &root);
> + set_fs_root(current->fs, &transport->root);
> +
>    status = xs_local_finish_connecting(xprt, sock);
> +
> + set_fs_root(current->fs, &root);
> + path_put(&root);
> +
>    switch (status) {
>    case 0:
```

Hi Stanislav,

What happens here if the mount namespace of the process that originally created the sock_xprt no longer exists? Should we care about that case?

Cheers
  Trond

--
Trond Myklebust
Linux NFS client maintainer

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