
Subject: [RFC PATCH 2/2] SUNRPC: connect local transports with
unix_stream_connect_root() helper

Posted by [Stanislav Kinsbursky](#) on Fri, 10 Aug 2012 12:57:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

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 using special helper
unix_stream_connect_root(), which lookup socket file starting from passed root.

Signed-off-by: Stanislav Kinsbursky <skinsbursky@parallels.com>

net/sunrpc/xprtsock.c | 28 ++++++-----
1 files changed, 25 insertions(+), 3 deletions(-)

diff --git a/net/sunrpc/xprtsock.c b/net/sunrpc/xprtsock.c
index 890b03f..01a6f2a 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
@@ -45,6 +46,7 @@
#include <net/checksum.h>
#include <net/udp.h>
#include <net/tcp.h>
+#include <net/af_unix.h>

#include "sunrpc.h"

@@ -255,6 +257,11 @@ 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;
+ };

/*
@@ -1873,7 +1880,8 @@ static int xs_local_finish_connecting(struct rpc_xprt *xprt,
/* Tell the socket layer to start connecting... */
xprt->stat.connect_count++;
xprt->stat.connect_start = jiffies;
- return kernel_connect(sock, xs_addr(xprt), xprt->addrlen, 0);
+ return unix_stream_connect_root(&transport->root, sock, xs_addr(xprt),
+   xprt->addrlen, 0);
}

/**
@@ -2213,6 +2221,18 @@ static void xs_connect(struct rpc_task *task)
}
}

+static void xs_local_destroy(struct rpc_xprt *xprt)
+{
+ struct sock_xprt *transport = container_of(xprt, struct sock_xprt, xprt);
+ struct path root = transport->root;
+
+ dprintk("RPC:   xs_local_destroy xprt %p\n", xprt);
+
+ xs_destroy(xprt);
+
+ path_put(&root);
+}
+
/**
 * xs_local_print_stats - display AF_LOCAL socket-specific stats
 * @xprt: rpc_xprt struct containing statistics
@@ -2431,7 +2451,7 @@ static struct rpc_xprt_ops xs_local_ops = {
 .send_request = xs_local_send_request,
 .set_retrans_timeout = xprt_set_retrans_timeout_def,
 .close = xs_close,
- .destroy = xs_destroy,
+ .destroy = xs_local_destroy,
 .print_stats = xs_local_print_stats,
};

@@ -2607,8 +2627,10 @@ static struct rpc_xprt *xs_setup_local(struct xprt_create *args)
dprintk("RPC:   set up xprt to %s via AF_LOCAL\n",

```

```
xprt->address_strings[RPC_DISPLAY_ADDR]);  
  
- if (try_module_get(THIS_MODULE))  
+ if (try_module_get(THIS_MODULE)) {  
+ get_fs_root(current->fs, &transport->root);  
    return xprt;  
+ }  
    ret = ERR_PTR(-EINVAL);  
out_err:  
    xprt_free(xprt);
```
