
Subject: [PATCH] Make CIFS mount work in a container.
Posted by [Rob Landley](#) on Sat, 22 Jan 2011 21:44:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Rob Landley <rlandley@parallels.com>

Teach cifs about network namespaces, so mounting uses addresses/routing visible from the container rather than from init context.

For a long drawn out test reproduction sequence, see:

<http://landley.livejournal.com/47024.html>
<http://landley.livejournal.com/47205.html>
<http://landley.livejournal.com/47476.html>

Signed-off-by: Rob Landley <rlandley@parallels.com>
Reviewed-by: Jeff Layton <jlayton@redhat.com>

Several other people commented on the patch without giving a formal byline, see:
<https://lists.linux-foundation.org/pipermail/containers/2011-January/026250.html>

This is version three of the patch. Discussion on the other lists seems to have died down, and it Works For Me (tm).

The patch is almost size neutral when CONFIG_NET_NS is disabled, everything but the switch from sock_create_kern() to __sock_create() (which has two more function arguments) drops out.

```
fs/cifs/cifsglob.h | 33 ++++++
fs/cifs/connect.c | 12 ++++++
2 files changed, 43 insertions(+), 2 deletions(-)
```

```
diff --git a/fs/cifs/cifsglob.h b/fs/cifs/cifsglob.h
index 606ca8b..54cd4ab 100644
--- a/fs/cifs/cifsglob.h
+++ b/fs/cifs/cifsglob.h
@@ -165,6 +165,9 @@ struct TCP_Server_Info {
     struct socket *ssocket;
     struct sockaddr_storage dstaddr;
     struct sockaddr_storage srcaddr; /* locally bind to this IP */
+#ifdef CONFIG_NET_NS
+ struct net *net;
+#endif
     wait_queue_head_t response_q;
     wait_queue_head_t request_q; /* if more than maxmpx to srvr must block*/
     struct list_head pending_mid_q;
@@ -224,6 +227,36 @@ struct TCP_Server_Info {
 };
```



```

        continue;
@@ -1607,6 +1610,8 @@ cifs_put_tcp_session(struct TCP_Server_Info *server)
    return;
}

+ put_net(cifs_net_ns(server));
+
    list_del_init(&server->tcp_ses_list);
    spin_unlock(&cifs_tcp_ses_lock);

@@ -1679,6 +1684,7 @@ cifs_get_tcp_session(struct smb_vol *volume_info)
    goto out_err;
}

+ cifs_set_net_ns(tcp_ses, get_net(current->nsproxy->net_ns));
    tcp_ses->hostname = extract_hostname(volume_info->UNC);
    if (IS_ERR(tcp_ses->hostname)) {
        rc = PTR_ERR(tcp_ses->hostname);
@@ -1754,6 +1760,8 @@ cifs_get_tcp_session(struct smb_vol *volume_info)
    out_err_crypto_release:
        cifs_crypto_shash_release(tcp_ses);

+ put_net(cifs_net_ns(tcp_ses));
+
    out_err:
        if (tcp_ses) {
            if (!IS_ERR(tcp_ses->hostname))
@@ -2265,8 +2273,8 @@ generic_ip_connect(struct TCP_Server_Info *server)
    }

    if (socket == NULL) {
- rc = sock_create_kern(sfamily, SOCK_STREAM,
-     IPPROTO_TCP, &socket);
+ rc = __sock_create(cifs_net_ns(server), sfamily, SOCK_STREAM,
+     IPPROTO_TCP, &socket, 1);
        if (rc < 0) {
            cERROR(1, "Error %d creating socket", rc);
            server->ssocket = NULL;

```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
