Subject: [PATCH 1/2] c/r: Do not crash if socket has no peercred
Posted by Dan Smith on Mon, 07 Feb 2011 16:42:58 GMT
View Forum Message <> Reply to Message

Unconnected sockets don't have a valid sk_peercred yet. If we checkpoint
them, we'll segv on the NULL pointer.

Signed-off-by: Dan Smith <danms@us.ibm.com>
---
net/unix/checkpoint.c | 52 ++++++++++++++++++++++++++----------------
1 files changed, 29 insertions(+), 23 deletions(-)

```
diff --git a/net/unix/checkpoint.c b/net/unix/checkpoint.c
index 708df40..25464cc 100644
--- a/net/unix/checkpoint.c
+++ b/net/unix/checkpoint.c
@@ -159,16 +159,19 @@ int unix_checkpoint(struct ckpt_ctx *ctx, struct socket *sock)
   goto out;
  }

- /*
-  * intentionally drop 'const' qualifier for checkpoint_obj() to
-  * increment the usage count - it does not alter the credentials.
-  */
- cred = (struct cred *) get_cred(sock->sk->sk_peer_cred);
- un->peercred = checkpoint_obj(ctx, cred, CKPT_OBJ_CRED);
- put_cred(cred);
- if (un->peercred < 0) {
-  ret = un->peercred;
-  goto out;
+ if (sock->sk->sk_peer_cred) {
+  /*
+   * intentionally drop 'const' qualifier for
+   * checkpoint_obj() to increment the usage count - it
+   * does not alter the credentials.
+   */
+  cred = (struct cred *) get_cred(sock->sk->sk_peer_cred);
+  un->peercred = checkpoint_obj(ctx, cred, CKPT_OBJ_CRED);
+  put_cred(cred);
+  if (un->peercred < 0) {
+   ret = un->peercred;
+   goto out;
+  }
  }

  ret = ckpt_write_obj(ctx, (struct ckpt_hdr *) un);
@@ -438,19 +441,22 @@ static int unix_restore_connected(struct ckpt_ctx *ctx,
   addrlen = un->laddr_len;
```

```
  }

- cred = ckpt_obj_fetch(ctx, un->peercred, CKPT_OBJ_CRED);
- if (!cred) {
-  ckpt_err(ctx, -EINVAL, "%(O)Bad peer cred\n", un->peercred);
-  return -EINVAL;
- }
-
- if (may_setuid(ctx->realcred->user->user_ns, cred->uid) &&
-     may_setgid(cred->gid)) {
-  set_peercred(sk, task_tgid(current), cred);
- } else {
-  ckpt_err(ctx, -EPERM, "peercred %i:%i would require setuid",
-    cred->uid, cred->gid);
-  return -EPERM;
+ if (un->peercred) {
+  cred = ckpt_obj_fetch(ctx, un->peercred, CKPT_OBJ_CRED);
+  if (!cred) {
+   ckpt_err(ctx, -EINVAL,
+     "%(O)Bad peer cred\n", un->peercred);
+   return -EINVAL;
+  }
+  if (may_setuid(ctx->realcred->user->user_ns, cred->uid) &&
+      may_setgid(cred->gid)) {
+   set_peercred(sk, task_tgid(current), cred);
+  } else {
+   ckpt_err(ctx, -EPERM,
+     "peercred %i:%i would require setuid",
+     cred->uid, cred->gid);
+   return -EPERM;
+  }
  }

  if (!dead && (un->peer > 0)) {
--
1.7.2.2
```

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs