Subject: [PATCH][XFRM] Fix potential race vs xfrm_state(only)_find and xfrm_hash_resize.
Posted by Pavel Emelianov on Thu, 13 Dec 2007 10:56:14 GMT
View Forum Message <> Reply to Message

The _find calls calculate the hash value using the
xfrm_state_hmask, without the xfrm_state_lock. But the
value of this mask can change in the _resize call under
the state_lock, so we risk to fail in finding the desired
entry in hash.

I think, that the hash value is better to calculate
under the state lock.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/net/xfrm/xfrm_state.c b/net/xfrm/xfrm_state.c
index 1af522b..1face71 100644
--- a/net/xfrm/xfrm_state.c
+++ b/net/xfrm/xfrm_state.c
@@ -759,7 +759,7 @@ xfrm_state_find(xfrm_address_t *daddr, xfrm_address_t *saddr,
   struct xfrm_policy *pol, int *err,
   unsigned short family)
 {
- unsigned int h = xfrm_dst_hash(daddr, saddr, tmpl->reqid, family);
+ unsigned int h;
  struct hlist_node *entry;
  struct xfrm_state *x, *x0;
  int acquire_in_progress = 0;
@@ -767,6 +767,7 @@ xfrm_state_find(xfrm_address_t *daddr, xfrm_address_t *saddr,
  struct xfrm_state *best = NULL;

  spin_lock_bh(&xfrm_state_lock);
+ h = xfrm_dst_hash(daddr, saddr, tmpl->reqid, family);
  hlist_for_each_entry(x, entry, xfrm_state_bydst+h, bydst) {
   if (x->props.family == family &&
       x->props.reqid == tmpl->reqid &&
@@ -868,11 +869,12 @@ struct xfrm_state *
 xfrm_stateonly_find(xfrm_address_t *daddr, xfrm_address_t *saddr,
       unsigned short family, u8 mode, u8 proto, u32 reqid)
 {
- unsigned int h = xfrm_dst_hash(daddr, saddr, reqid, family);
+ unsigned int h;
  struct xfrm_state *rx = NULL, *x = NULL;
  struct hlist_node *entry;
```

```
  spin_lock(&xfrm_state_lock);
+ h = xfrm_dst_hash(daddr, saddr, reqid, family);
  hlist_for_each_entry(x, entry, xfrm_state_bydst+h, bydst) {
   if (x->props.family == family &&
       x->props.reqid == reqid &&
```