

---

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 &&
```

---