
Subject: [PATCH 3/3] net - removing superfluous rcu_read_lock_held check

Posted by [Jiri Olsa](#) on Mon, 01 Nov 2010 19:15:09 GMT

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

hi,
the rcu_dereference_check is defined as

```
#define rcu_dereference_check(p, c) \
    __rcu_dereference_check((p), rcu_read_lock_held() || (c), __rcu)
```

so the caller does not need to specify rcu_read_lock_held() condition.

wbr,
jirka

Signed-off-by: Jiri Olsa <jolsa@redhat.com>

```
include/linux/rtnetlink.h      | 3 +--
include/net/sock.h              | 3 +--
net/mac80211/sta_info.c         | 4 ----
net/netlabel/netlabel_domainhash.c | 3 +--
net/netlabel/netlabel_unlabeled.c | 3 +--
5 files changed, 4 insertions(+), 12 deletions(-)
```

diff --git a/include/linux/rtnetlink.h b/include/linux/rtnetlink.h

index d42f274..dfe5ba1 100644

--- a/include/linux/rtnetlink.h

+++ b/include/linux/rtnetlink.h

```
@@ -758,8 +758,7 @@ extern int lockdep_rtnl_is_held(void);
 * or RTNL. Note : Please prefer rtnl_dereference() or rcu_dereference()
 */
```

```
#define rcu_dereference_rtnl(p) \
- rcu_dereference_check(p, rcu_read_lock_held() || \
- lockdep_rtnl_is_held())
+ rcu_dereference_check(p, lockdep_rtnl_is_held())
```

/**

* rtnl_dereference - fetch RCU pointer when updates are prevented by RTNL

diff --git a/include/net/sock.h b/include/net/sock.h

index c7a7362..bee3e9c 100644

--- a/include/net/sock.h

+++ b/include/net/sock.h

```
@@ -1274,8 +1274,7 @@ extern unsigned long sock_i_ino(struct sock *sk);
 static inline struct dst_entry *
```

```
__sk_dst_get(struct sock *sk)
```

```
{
```

```
- return rcu_dereference_check(sk->sk_dst_cache, rcu_read_lock_held() ||
```

```

-      sock_owned_by_user(sk) ||
+ return rcu_dereference_check(sk->sk_dst_cache, sock_owned_by_user(sk) ||
      lockdep_is_held(&sk->sk_lock.slock));
}

```

diff --git a/net/mac80211/sta_info.c b/net/mac80211/sta_info.c

index 6d8f897..c879217 100644

--- a/net/mac80211/sta_info.c

+++ b/net/mac80211/sta_info.c

```

@@ -94,7 +94,6 @@ struct sta_info *sta_info_get(struct ieee80211_sub_if_data *sdata,
 struct sta_info *sta;

```

```

    sta = rcu_dereference_check(local->sta_hash[STA_HASH(addr)],
-    rcu_read_lock_held() ||
    lockdep_is_held(&local->sta_lock) ||
    lockdep_is_held(&local->sta_mtx));
    while (sta) {
@@ -102,7 +101,6 @@ struct sta_info *sta_info_get(struct ieee80211_sub_if_data *sdata,
    memcmp(sta->sta.addr, addr, ETH_ALEN) == 0)
    break;
    sta = rcu_dereference_check(sta->hnext,
-    rcu_read_lock_held() ||
    lockdep_is_held(&local->sta_lock) ||
    lockdep_is_held(&local->sta_mtx));
}
@@ -120,7 +118,6 @@ struct sta_info *sta_info_get_bss(struct ieee80211_sub_if_data *sdata,
 struct sta_info *sta;

```

```

    sta = rcu_dereference_check(local->sta_hash[STA_HASH(addr)],
-    rcu_read_lock_held() ||
    lockdep_is_held(&local->sta_lock) ||
    lockdep_is_held(&local->sta_mtx));
    while (sta) {
@@ -129,7 +126,6 @@ struct sta_info *sta_info_get_bss(struct ieee80211_sub_if_data *sdata,
    memcmp(sta->sta.addr, addr, ETH_ALEN) == 0)
    break;
    sta = rcu_dereference_check(sta->hnext,
-    rcu_read_lock_held() ||
    lockdep_is_held(&local->sta_lock) ||
    lockdep_is_held(&local->sta_mtx));
}

```

diff --git a/net/netlabel/netlabel_domainhash.c b/net/netlabel/netlabel_domainhash.c

index d37b7f8..82795a4 100644

--- a/net/netlabel/netlabel_domainhash.c

+++ b/net/netlabel/netlabel_domainhash.c

```

@@ -55,8 +55,7 @@ struct netlbl_domhsh_tbl {
 * should be okay */
static DEFINE_SPINLOCK(netlbl_domhsh_lock);

```

```

#define netlbl_domhsh_rcu_deref(p) \
- rcu_dereference_check(p, rcu_read_lock_held() || \
-   lockdep_is_held(&netlbl_domhsh_lock))
+ rcu_dereference_check(p, lockdep_is_held(&netlbl_domhsh_lock))
static struct netlbl_domhsh_tbl *netlbl_domhsh = NULL;
static struct netlbl_dom_map *netlbl_domhsh_def = NULL;

diff --git a/net/netlabel/netlabel_unlabeled.c b/net/netlabel/netlabel_unlabeled.c
index e2b0a68..d2f982f 100644
--- a/net/netlabel/netlabel_unlabeled.c
+++ b/net/netlabel/netlabel_unlabeled.c
@@ -116,8 +116,7 @@ struct netlbl_unlhsh_walk_arg {
 * hash table should be okay */
static DEFINE_SPINLOCK(netlbl_unlhsh_lock);
#define netlbl_unlhsh_rcu_deref(p) \
- rcu_dereference_check(p, rcu_read_lock_held() || \
-   lockdep_is_held(&netlbl_unlhsh_lock))
+ rcu_dereference_check(p, lockdep_is_held(&netlbl_unlhsh_lock))
static struct netlbl_unlhsh_tbl *netlbl_unlhsh = NULL;
static struct netlbl_unlhsh_iface *netlbl_unlhsh_def = NULL;

--
1.7.1

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
