
Subject: [PATCH 2/3] kernel, cred, kvm, security - removing superfluous rcu_read_lock_held check

Posted by [Jiri Olsa](#) on Mon, 01 Nov 2010 19:15:08 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/cred.h    | 1 -
include/linux/fdtable.h | 1 -
include/linux/kvm_host.h | 1 -
kernel/exit.c           | 1 -
kernel/pid.c            | 1 -
kernel/rcutorture.c     | 2 --
security/keys/keyring.c | 1 -
7 files changed, 0 insertions(+), 8 deletions(-)
```

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

index 4aaeab3..a6b9afc 100644

--- a/include/linux/cred.h

+++ b/include/linux/cred.h

```
@@ -283,7 +283,6 @@ static inline void put_cred(const struct cred *_cred)
({
    const struct task_struct *__t = (task); \
    rcu_dereference_check(__t->real_cred, \
-    rcu_read_lock_held() || \
    task_is_dead(__t)); \
})
```

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

index 133c0ba..df7e3cf 100644

--- a/include/linux/fdtable.h

+++ b/include/linux/fdtable.h

```
@@ -60,7 +60,6 @@ struct files_struct {
```

```
#define rcu_dereference_check_fdtable(files, fdtdfd) \
    (rcu_dereference_check((fdtdfd), \
```

```

-      rcu_read_lock_held() || \
        lockdep_is_held(&(files)->file_lock) || \
        atomic_read(&(files)->count) == 1 || \
        rcu_my_thread_group_empty()))
diff --git a/include/linux/kvm_host.h b/include/linux/kvm_host.h
index a055742..a90a7e3 100644
--- a/include/linux/kvm_host.h
+++ b/include/linux/kvm_host.h
@@ -256,7 +256,6 @@ void kvm_put_kvm(struct kvm *kvm);
static inline struct kvm_memslots *kvm_memslots(struct kvm *kvm)
{
    return rcu_dereference_check(kvm->memslots,
-   -   srcu_read_lock_held(&kvm->srcu)
-       || lockdep_is_held(&kvm->slots_lock));
}

diff --git a/kernel/exit.c b/kernel/exit.c
index b194feb..f753342 100644
--- a/kernel/exit.c
+++ b/kernel/exit.c
@@ -85,7 +85,6 @@ static void __exit_signal(struct task_struct *tsk)
    struct tty_struct *uninitialized_var(tty);

    sighand = rcu_dereference_check(tsk->sighand,
-   -   rcu_read_lock_held() ||
        lockdep_tasklist_lock_is_held());
    spin_lock(&sighand->siglock);

diff --git a/kernel/pid.c b/kernel/pid.c
index 39b65b6..c02adda 100644
--- a/kernel/pid.c
+++ b/kernel/pid.c
@@ -402,7 +402,6 @@ struct task_struct *pid_task(struct pid *pid, enum pid_type type)
    if (pid) {
        struct hlist_node *first;
        first = rcu_dereference_check(hlist_first_rcu(&pid->tasks[type]),
-   -   rcu_read_lock_held() ||
        lockdep_tasklist_lock_is_held());
        if (first)
            result = hlist_entry(first, struct task_struct, pids[(type)].node);
diff --git a/kernel/rcutorture.c b/kernel/rcutorture.c
index 9d8e8fb..0956a73 100644
--- a/kernel/rcutorture.c
+++ b/kernel/rcutorture.c
@@ -807,7 +807,6 @@ static void rcu_torture_timer(unsigned long unused)
    idx = cur_ops->readlock();
    completed = cur_ops->completed();
    p = rcu_dereference_check(rcu_torture_current,

```

```

- rcu_read_lock_held() ||
  rcu_read_lock_bh_held() ||
  rcu_read_lock_sched_held() ||
  srcu_read_lock_held(&srcu_ctl));
@@ -868,7 +867,6 @@ rcu_torture_reader(void *arg)
  idx = cur_ops->readlock();
  completed = cur_ops->completed();
  p = rcu_dereference_check(rcu_torture_current,
- rcu_read_lock_held() ||
  rcu_read_lock_bh_held() ||
  rcu_read_lock_sched_held() ||
  srcu_read_lock_held(&srcu_ctl));
diff --git a/security/keys/keyring.c b/security/keys/keyring.c
index d37f713..73c23f2 100644
--- a/security/keys/keyring.c
+++ b/security/keys/keyring.c
@@ -157,7 +157,6 @@ static void keyring_destroy(struct key *keyring)
 }

  klist = rcu_dereference_check(keyring->payload.subscriptions,
- rcu_read_lock_held() ||
  atomic_read(&keyring->usage) == 0);
  if (klist) {
    for (loop = klist->nkeys - 1; loop >= 0; loop--)
--
1.7.1

```

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