
Subject: [PATCH 4/5] unify netlink kernel socket recognition

Posted by [den](#) on Fri, 05 Oct 2007 14:46:34 GMT

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

There are currently two ways to determine whether the netlink socket is a kernel one or a user one. This patch creates a single inline call for this purpose and unifies all the calls in the af_netlink.c

No similar calls are found outside af_netlink.c.

Signed-off-by: Denis V. Lunev <den@openvz.org>

Acked-by: Alexey Kuznetsov <kuznet@ms2.inr.ac.ru>

--- ./net/netlink/af_netlink.c.nlk4 2007-08-26 19:30:38.000000000 +0400

+++ ./net/netlink/af_netlink.c 2007-10-02 12:27:33.000000000 +0400

```
@@ -92,6 +92,11 @@ static inline struct netlink_sock *nlk_s
    return container_of(sk, struct netlink_sock, sk);
}
```

```
+static inline int netlink_is_kernel(struct sock *sk)
```

```
+{
```

```
+ return nlk_sk(sk)->flags & NETLINK_KERNEL_SOCKET;
```

```
+}
```

```
+
```

```
struct nl_pid_hash {
```

```
    struct hlist_head *table;
```

```
    unsigned long rehash_time;
```

```
@@ -489,7 +494,7 @@ static int netlink_release(struct socket
    module_put(nlk->module);
```

```
    netlink_table_grab();
```

```
- if (nlk->flags & NETLINK_KERNEL_SOCKET) {
```

```
+ if (netlink_is_kernel(sk)) {
```

```
    kfree(nl_table[sk->sk_protocol].listeners);
```

```
    nl_table[sk->sk_protocol].module = NULL;
```

```
    nl_table[sk->sk_protocol].registered = 0;
```

```
@@ -716,7 +721,7 @@ static struct sock *netlink_getsockbypid
```

```
    /* Don't bother queuing skb if kernel socket has no input function */
```

```
    nlk = nlk_sk(sock);
```

```
- if ((nlk->pid == 0 && !nlk->data_ready) ||
```

```
+ if ((netlink_is_kernel(sock) && !nlk->data_ready) ||
```

```
    (sock->sk_state == NETLINK_CONNECTED &&
```

```
    nlk->dst_pid != nlk_sk(ssk)->pid)) {
```

```
    sock_put(sock);
```

```
@@ -762,7 +767,7 @@ int netlink_attachskb(struct sock *sk, s
```

```
    test_bit(0, &nlk->state)) {
```

```
    DECLARE_WAITQUEUE(wait, current);
```

```
    if (!timeo) {  
-   if (!ssk || nlk_sk(ssk)->pid == 0)  
+   if (!ssk || netlink_is_kernel(ssk))  
        netlink_overrun(sk);  
        sock_put(sk);  
        kfree_skb(skb);  
@@ -861,7 +875,7 @@ int netlink_has_listeners(struct sock *s  
    int res = 0;  
    unsigned long *listeners;  
  
- BUG_ON(!(nlk_sk(sk)->flags & NETLINK_KERNEL_SOCKET));  
+ BUG_ON(!netlink_is_kernel(sk));  
  
    rcu_read_lock();  
    listeners = rcu_dereference(nl_table[sk->sk_protocol].listeners);
```
