
Subject: [PATCH 0/6][IPV6]: Introduce the INET6_TW_MATCH macro.

Posted by [Pavel Emelianov](#) on Thu, 31 Jan 2008 12:29:20 GMT

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

We have INET_MATCH, INET_TW_MATCH and INET6_MATCH to test sockets and twbuckets for matching, but ipv6 twbuckets are tested manually.

Here's the INET6_TW_MATCH to help with it.

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

```
include/linux/ipv6.h      | 8 ++++++++
net/ipv6/inet6_hashtables.c | 21 +++-----
2 files changed, 11 insertions(+), 18 deletions(-)
```

```
diff --git a/include/linux/ipv6.h b/include/linux/ipv6.h
```

```
index 5d35a4c..c347860 100644
```

```
--- a/include/linux/ipv6.h
```

```
+++ b/include/linux/ipv6.h
```

```
@@ -465,6 +465,14 @@ static inline struct raw6_sock *raw6_sk(const struct sock *sk)
    ipv6_addr_equal(&inet6_sk(__sk)->rcv_saddr, (__daddr)) && \
    (!((__sk)->sk_bound_dev_if) || ((__sk)->sk_bound_dev_if == (__dif)))
```

```
+#define INET6_TW_MATCH(__sk, __hash, __saddr, __daddr, __ports, __dif) \
+ (((__sk)->sk_hash == (__hash)) && \
+ *((__portpair *)&(inet6_twsksk(__sk)->tw_dport)) == (__ports)) && \
+ ((__sk)->sk_family == PF_INET6) && \
+ (ipv6_addr_equal(&inet6_twsksk(__sk)->tw_v6_daddr, (__saddr)) && \
+ (ipv6_addr_equal(&inet6_twsksk(__sk)->tw_v6_rcv_saddr, (__daddr)) && \
+ (!((__sk)->sk_bound_dev_if) || ((__sk)->sk_bound_dev_if == (__dif)))) \
+
#endif /* __KERNEL__ */
```

```
#endif /* _IPV6_H */
```

```
diff --git a/net/ipv6/inet6_hashtables.c b/net/ipv6/inet6_hashtables.c
```

```
index a66a7d8..06b01be 100644
```

```
--- a/net/ipv6/inet6_hashtables.c
```

```
+++ b/net/ipv6/inet6_hashtables.c
```

```
@@ -80,17 +80,8 @@ struct sock *__inet6_lookup_established(struct inet_hashinfo *hashinfo,
 }
 /* Must check for a TIME_WAIT'er before going to listener hash. */
 sk_for_each(sk, node, &head->twchain) {
- const struct inet_timewait_sock *tw = inet_twsks(sk);
-
- if(*((__portpair *)&(tw->tw_dport)) == ports &&
- sk->sk_family == PF_INET6) {
```

```

- const struct inet6_timewait_sock *tw6 = inet6_twsk(sk);
-
- if (ipv6_addr_equal(&tw6->tw_v6_daddr, saddr) &&
-     ipv6_addr_equal(&tw6->tw_v6_rcv_saddr, daddr) &&
-     (!sk->sk_bound_dev_if || sk->sk_bound_dev_if == dif))
-     goto hit;
- }
+ if (INET6_TW_MATCH(sk, hash, saddr, daddr, ports, dif))
+     goto hit;
+ }
    read_unlock(lock);
    return NULL;
@@ -185,15 +176,9 @@ static int __inet6_check_established(struct inet_timewait_death_row
*death_row,

```

```

/* Check TIME-WAIT sockets first. */
sk_for_each(sk2, node, &head->twchain) {
- const struct inet6_timewait_sock *tw6 = inet6_twsk(sk2);
-
-     tw = inet_twsk(sk2);

- if((__portpair *)&(tw->tw_dport)) == ports &&
-     sk2->sk_family == PF_INET6 &&
-     ipv6_addr_equal(&tw6->tw_v6_daddr, saddr) &&
-     ipv6_addr_equal(&tw6->tw_v6_rcv_saddr, daddr) &&
-     (!sk2->sk_bound_dev_if || sk2->sk_bound_dev_if == dif)) {
+ if (INET6_TW_MATCH(sk2, hash, saddr, daddr, ports, dif)) {
    if (twsk_unique(sk, sk2, twp))
        goto unique;
    else
--
1.5.3.4

```
