
Subject: [PATCH net-2.6.25][IPV6]: Introduce the INET6_TW_MATCH() macro.
Posted by [Pavel Emelianov](#) on Thu, 24 Jan 2008 15:54:49 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.

Since the commit b3652b2dc5ec6ccd946ae9136b30c6babb81305a [IPV6]: Mismatched tw match in __inet6_check_established. is accepted this patch will not break the __inet6_check_est. logic.

This will also help with per-namespace socket lookup patches in the nearest future.

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

```
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_tws(__sk)->tw_dport)) == (__ports)) && \
+ ((__sk)->sk_family == PF_INET6) && \
+ (ipv6_addr_equal(&inet6_tws(__sk)->tw_v6_daddr, (__saddr))) && \
+ (ipv6_addr_equal(&inet6_tws(__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 d0b3447..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_twsk(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 == sk->sk_bound_dev_if) {
+ if (INET6_TW_MATCH(sk2, hash, saddr, daddr, ports, dif)) {
    if (twsk_unique(sk, sk2, twp))
        goto unique;
    else

```
