

Re: [KJ] [PATCH] net/core/flow.c: compare data with memcmp

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-12/msg08173.html>

- *From:* Nishanth Aravamudan <nacc@xxxxxxxxxxx>
 - *Date:* Sun, 31 Dec 2006 11:49:39 -0800
-

On 31.12.2006 [17:37:05 +0100], Daniel Marjamäki wrote:

```
From: Daniel Marjamäki
This has been tested by me.
Signed-off-by: Daniel Marjamäki <daniel.marjamaki@xxxxxxxxxx>
--- linux-2.6.20-rc2/net/core/flow.c 2006-12-27 09:59:56.000000000 +0100
+++ linux/net/core/flow.c 2006-12-31 18:26:06.000000000 +0100
@@ -144,29 +144,16 @@ typedef u32 flow_compare_t;
```

```
extern void flowi_is_missized(void);
```

```
/* I hear what you're saying, use memcmp. But memcmp cannot make
 * important assumptions that we can here, such as alignment and
 * constant size.
 */
```

```
static int flow_key_compare(struct flowi *key1, struct flowi *key2)
{
    flow_compare_t *k1, *k1_lim, *k2;
    const int n_elem = sizeof(struct flowi) / sizeof(flow_compare_t);

    if (sizeof(struct flowi) % sizeof(flow_compare_t))
        flowi_is_missized();

    k1 = (flow_compare_t *) key1;
    k1_lim = k1 + n_elem;

    k2 = (flow_compare_t *) key2;

    do {
        if (*k1++ != *k2++)
            return 1;
    } while (k1 < k1_lim);
    /* Number of elements to compare */
    const int n_elem = sizeof(struct flowi) / sizeof(flow_compare_t);

    return 0;
    /* Compare all elements in key1 and key2. */
    return memcmp(key1, key2, n_elem * sizeof(flow_compare_t));
```

Re: [KJ] [PATCH] net/core/flow.c: compare data with memcmp

}

Small nit, I don't think either of your comments make the code any clearer than the code on its own, so drop them both.

Thanks,
Nish

--

Nishanth Aravamudan <nacc@xxxxxxxxxx>
IBM Linux Technology Center

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>