

## Re: UDP sockets bound to ANY send answers with wrong src ip address

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-06/2569.html>

---

**From:** Denis Vlasenko ([vda\\_at\\_port.imtp.ilyichevsk.odessa.ua](mailto:vda_at_port.imtp.ilyichevsk.odessa.ua))

**Date:** 06/11/04

To: YOSHIFUJI Hideaki <[yoshfuji@linux-ipv6.org](mailto:yoshfuji@linux-ipv6.org)>

Date: Fri, 11 Jun 2004 12:30:35 +0300

On Wednesday 09 June 2004 15:24, YOSHIFUJI Hideaki wrote:

> Denis Vlasenko <[vda@port.imtp.ilyichevsk.odessa.ua](mailto:vda@port.imtp.ilyichevsk.odessa.ua)> says:

> > I observe that UDP sockets listening on ANY

> > send response packets with ip addr derived from

> > ip address of interface which is used to send 'em

> > instead of using dst ip address of client's packet.

>

> use IP\_PKTINFO when responding the client.

Thanks!

With your help and some googling I've found and adapted code to get dst ip of UDP packet.

Small test program successfully ran and reported correct dst addresses of incoming UDP packets.

Now, I am trying to fix (or shall I say 'improve?') dnscache. You may find some code below my sig. It's a start.

The problem is, how to `_send replies_` with correct src ip?

I can bind a temporary socket to needed src address,

do a `sendto()`, then close socket. This will work,

but this can introduce a race – any incoming

packet to this (ip,port) will inadvertently

be classified as belonging to temp socket!

This is going to be a nasty bug, manifesting itself only under load.

I looked into `sendmsg()`. Looks like ther is no way to indicate source ip.

Shall I use some other technique?

--

vda

```
#if defined IP_RECVDSTADDR
```

## Linux-Kernel: Re: UDP sockets bound to ANY send answers with wrong src ip address

```
# define DSTADDR_SOCKOPT IP_RECVDSTADDR
# define DSTADDR_DATASIZE (MSG_SPACE(sizeof(struct in_addr)))
# define dstaddr(x) (MSG_DATA(x))
#elif defined IP_PKTINFO
# define DSTADDR_SOCKOPT IP_PKTINFO
# define DSTADDR_DATASIZE (MSG_SPACE(sizeof(struct in_pktinfo)))
# define dstaddr(x) (&(((struct in_pktinfo *) (MSG_DATA(x)))->ipi_addr))
#else
# error "can't determine socket option"
#endif
int socket_recv4_dst(int s, char *buf, int len, char ip[4], uint16 *port, char ipdst[4])
{
    int r;
    struct iovec iov[1];
    struct sockaddr_in sa;
    union control_data cmsg;
    struct cmsghdr *cmsgptr;
    struct msghdr msg;
    iov[0].iov_base = buf;
    iov[0].iov_len = len;
    memset(&msg, 0, sizeof msg);
    msg.msg_name = &sa;
    msg.msg_namelen = sizeof sa;
    msg.msg_iov = iov;
    msg.msg_iovlen = 1;
    msg.msg_control = &cmsg;
    msg.msg_controllen = sizeof cmsg;
    { // FIXME: we need to do it ONCE! move it into socket_bind4_dstaddropt()
        int sockopt;
        sockopt = 1;
        if (setsockopt(s, IPPROTO_IP, DSTADDR_SOCKOPT, &sockopt, sizeof sockopt) == -1)
            return -1;
    }
    //r = recvfrom(s, buf, len, 0, (struct sockaddr *) &sa, &dummy);
    r = recvmsg(s, &msg, 0);
    if (r == -1) return -1;
    // Here we retrieve destination IP and memorize it
    for (cmsgptr = MSG_FIRSTHDR(&msg);
        cmsgptr != NULL;
        cmsgptr = MSG_NXTHDR(&msg, cmsgptr)) {
        if (cmsgptr->cmsg_level == IPPROTO_IP
            && cmsgptr->cmsg_type == DSTADDR_SOCKOPT) {
            byte_copy(ipdst, 4, (char *) dstaddr(cmsgptr));
        }
    }
    return r;
}
-
```

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

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

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