

byte swapped udp length in diskless bootp request ?

byte swapped udp length in diskless bootp request ?

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/current/2006-11/msg00758.html>

- *From:* Luigi Rizzo <rizzo@xxxxxxx>
 - *Date:* Thu, 30 Nov 2006 10:55:37 -0800
-

i was just trying to diskless-boot a -current kernel,
and when it was time for the kernel to acquire the address
i was getting the usual

DHCP/BOOTP timeout for server 255.255.255.255

Usually it is because of lack of connectivity, but
a bit of inspection on the server showed (as you can see
below) that the UDP len field is byte-swapped – the 05bc
in the packet is in little-endian format, causing the
server to reject it.

I am trying to follow the code in sys/nfsclient/bootp_subr.c
(which should send the packet) but it seemd to call sosend()
(at line 755) to generate the packet, so it looks really strange
that the bug is in such a central place... any ideas ?

If that matters, the kernel is cross-compiled on a 6.2-RC1
box using a relatively fresh source tree.

cheers
luigi

TCPDUMP OUTPUT ON THE SERVER SIDE:

```
r1# tcpdump -nli em0 -s 0 -veX port 67
tcpdump: listening on em0, link-type EN10MB (Ethernet), capture size 65535 bytes
19:37:30.633525 00:40:f4:34:ad:09 > ff:ff:ff:ff:ff:ff, ethertype IPv4 (0x0800), length 1502: truncated-ip -
46645 bytes missing! (tos 0x0, ttl 1, id 83, offset 0, flags [none], proto: UDP (17), length: 48133, bad cksum
5bc (->fd95)!) 0.0.0.0.68 > 255.255.255.255.67: BOOTP/DHCP, Request from 00:40:f4:34:ad:09, length:
48105, xid:0xffff0001, secs:400, flags: [Broadcast]
Client Ethernet Address: 00:40:f4:34:ad:09
Vendor-rfc1048:
MSZ:1460
VC:"FreeBSD:i386:7.0-CURRENT"
DHCP:DISCOVER
0x0000: 4500 bc05 0053 0000 0111 05bc 0000 0000 E....S.....
0x0010: ffff ffff 0044 0043 05bc 0000 0101 0600 .....D.C.....
```

byte swapped udp length in diskless bootp request ?

byte swapped udp length in diskless bootp request ?

0x0020: ffff 0001 0190 8000 0000 0000 0000 0000
0x0030: 0000 0000 0000 0000 0040 f434 ad09 0000@.4....

... and so on. The rest of the packet has all
the good data up to

0x05a0: 0000 0000 0000 0000 0000 0000 0000 0000
0x05b0: 0000 0000 0000 0000 0000 0000 0000 0000
0x05c0: 0000 0000 0000 0000 0000 0000 0000 0000

freebsd-current@xxxxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-current>
To unsubscribe, send any mail to "freebsd-current-unsubscribe@xxxxxxxxxxxxx"