Subject: Re: [PATCH][ICMP]: Dst entry leak in icmp_send host re-lookup code (v2).
Posted by Julian Anastasov on Wed, 02 Apr 2008 09:19:06 GMT
View Forum Message <> Reply to Message

Hello,

On Tue, 1 Apr 2008, Herbert Xu wrote:

> On Wed, Mar 26, 2008 at 12:25:40PM +0300, Pavel Emelyanov wrote:
> > Commit 8b7817f3a959ed99d7443afc12f78a7e1fcc2063 ([IPSEC]: Add ICMP host
> > relookup support) introduced some dst leaks on error paths: the rt
> > pointer can be forgotten to be put. Fix it bu going to a proper label.
>
> I just remembered that we have exactly the same code path in IPv6
> and sure enough it also has the same bug.

OK, we found there was a leak, but why it happens? Initially,
I thought it was caused by saddr=0 provided to ip_route_input. Some
debugging shows that in the case with forwarded skb (with attached
input route) saddr is set to 0 but later xfrm_decode_session_reverse
rebuilds fl with addresses from packet. So, it was not that we
play with saddr=0. In my test setup with 2 interfaces ip_route_input
failed because I don't have route to the original destination
which is now provided as saddr to ip_route_input. No ICMP was sent
to sender while previous kernels send ICMP.

As result, this new code adds some new checks that are not
valid for all cases. When kernel wants to say that destination is
unreachable it can not do it. We should talk with sender without
considering destination address.

May be this code should be reverted for 2.6.25 or some extra
checks should be added considering the different variants where
icmp_send can be called:

- original packet is incoming, destined to localhost (rt->fl.iif!=0
 and rt->rt_flags & RTCF_LOCAL)
- original packet is incoming, destined to remote host (rt->fl.iif!=0
 and !(rt->rt_flags & RTCF_LOCAL))
- original packet is outgoing, destined to localhost
- original packet is outgoing, destined to remote host

In my case even SNAT happened before icmp_send, so
ip_route_input failed for saddr=ORIGINAL_TARGET and
daddr=MASQ_ADDR_WHICH_IS_LOCAL indev=MADDR_DEVICE

Regards

--
Julian Anastasov <ja@ssi.bg>