strongSwan - Bug #3225

"Error sending to PF_KEY socket: No buffer space available" in verbose on FreeBSD

25.10.2019 10:18 - Jean-François Hren

Status:	Closed	**Start date:**	25.10.2019
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	kernel-interface		
Target version:	5.8.2		
Affected version:	5.8.1	**Resolution:**	Fixed

Description

Hello,

We got the message "_error sending to PF_KEY socket: No buffer space available_" in the verbose and no tunnel can be started or rekeyed when it happened.
We searched for the problem and found something interesting.
When an "_echo 'flush tcp;' | setkey -c_" is done, a PFKEY message is sent to the kernel to flush SA of the specified proto (here tcp but the same can be said for esp and other commands can lead to same problem.)
When the flush is done, a PFKEY message is sent back to all raw sockets with family set to PF_KEY and protocol set to PF_KEY_V2. In our case, this message is sent to setkey but also to both sockets opened by Charon: the one used to send message to the kernel and the one used to listen for events from the kernel. The second socket discards the message since it has a PID but the first one does nothing. If the same command is done enough times, the receive buffer of the first socket becomes full and when Charon wants to send a PFKEY message, the message is sent correctly but the kernel answer cannot since the receive buffer is full leading to the error message in the verbose.

I think a solution would be to drain the receive buffer of the socket used to send PFKEY message to the kernel before attempting to send one. Anything waiting in the receive buffer is of no use to us anyway. I attached a patch with this approach.

Associated revisions

Revision 62e7c68b - 25.10.2019 13:53 - Tobias Brunner

kernel-pfkey: Clear receive buffer before sending request

Many of the messages sent by the kernel, including confirmations to our
requests, are sent as broadcasts to all PF_KEY sockets. So if an
external tool is used to manage SAs/policies (e.g. unrelated to IPsec)
the receive buffer might be filled, resulting in errors like these:

```
error sending to PF_KEY socket: No buffer space available
```

To avoid this, just clear the buffer before sending any message.

Fixes #3225.

History

#1 - 25.10.2019 11:14 - Tobias Brunner

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

- Target version set to 5.8.2

- Resolution set to Fixed

> I think a solution would be to drain the receive buffer of the socket used to send PFKEY message to the kernel before attempting to send one. Anything waiting in the receive buffer is of no use to us anyway. I attached a patch with this approach.

Hm, interesting. Never noticed that the Linux kernel does this too. Actually, I wasn't aware that many messages we handle as reply to our requests are sent as broadcasts to all sockets (e.g. confirmations to operations on SAs and policies). So whenever an external tool is used to modify SAs and policies (not only to flush them) we get messages on all sockets.

That's definitely better solved by Netlink/XFRM where messages are multicasted to only those registered for specific messages/events (there not even our event socket receives messages when SAs or policies are managed as it is only registered for kernel-generated events like acquires and expires).

I pushed a fix to the *3225-pfkey-clear-buf* branch.

#2 - 25.10.2019 11:51 - Jean-François Hren

I tested the branch and it works perfectly for us.
Thank you for your reactivity.

#3 - 25.10.2019 13:54 - Tobias Brunner

- Status changed from Feedback to Closed

Pushed to master.

Files

fix_rcv_buffer_pfkey.patch	946 Bytes	25.10.2019	Jean-François Hren