pfSense - Bug #5800

Rule to block carp from (self) ineffective when pfsync is in use

01/22/2016 01:41 PM - Jim Pingle

Status:	Resolved	**Start date:**	01/22/2016
Priority:	Normal	**Due date:**	
Assignee:	Jim Pingle	**% Done:**	0%
Category:	CARP	**Estimated time:**	0.00 hour
Target version:	2.3		
Plus Target Version:		**Affected Version:**	2.2.x
Release Notes:	Default	**Affected Architecture:**	

Description

Adding this primarily for documentation purposes. Normally with our default ruleset pf will block CARP packets coming in from (self) to prevent L2 packet duplication from affecting the CARP status. However when pfsync is enabled, there is a bit of a quandary here:

- CARP heartbeat packet from primary leaves the primary
- CARP heartbeat packet is delivered both to the secondary and also back to the primary by the misconfigured L2
- The primary rejects this first CARP heartbeat packet and the secondary accepts it (both correct actions)
- The secondary creates a state allowing the heartbeat packet
- The state is synchronized back to the primary
- The primary then accepts subsequent duplicate heartbeat packets, causing itself to be demoted due to the packets arriving faster than it transmits, ending up with flapping VIPs and what appears to the user as a dual backup situation

Disabling pfsync and clearing the CARP states allows the CARP VIPs to function until the L2 is repaired.

Not sure if there is any good way around that. Given that it's an L2 problem there is only so much we can do to stop foot-shooting in that area. Perhaps pass carp in expected directions/interfaces with no state?

Associated revisions

Revision b9037cbe - 01/26/2016 11:10 PM - Chris Buechler

Use 'no state' on CARP pass rules. Ticket #5800

History

#1 - 01/22/2016 02:09 PM - Jim Thompson

- Assignee set to Jim Pingle

assigned to Pingle until he assigns it elsewhere.

#2 - 01/22/2016 02:45 PM - Chris Buechler

Rather than trying to block the traffic, CARP should really just ignore its own advertisements. Fix the root issue rather than trying to block traffic to work around it. mgrooms posted a patch for this years back:
https://lists.freebsd.org/pipermail/freebsd-net/2009-July/022545.html

that won't cleanly apply to FreeBSD 10.x, though it's probably straight forward enough to bring it up to date for Luiz.

#3 - 01/22/2016 02:46 PM - Jim Pingle

- Assignee changed from Jim Pingle to Luiz Souza

#4 - 01/26/2016 03:31 AM - Jim Thompson

- Assignee changed from Luiz Souza to Jim Pingle

abort patches like this.

Either determine that it's a problem in upstream, or deal with it in another way.

#5 - 01/26/2016 03:18 PM - Chris Buechler

- Status changed from New to Confirmed

it's a problem in upstream, for which we should be able to get a fix along the lines of what was linked above committed to FreeBSD.

#6 - 01/26/2016 03:38 PM - Jim Pingle

- Assignee changed from Jim Pingle to Luiz Souza

Back to Luiz to analyze the patch and get it pushed upstream.

#7 - 01/26/2016 06:31 PM - Jim Thompson

- Assignee changed from Luiz Souza to Jim Pingle

#8 - 01/27/2016 12:17 AM - Chris Buechler

- Status changed from Confirmed to Resolved

- Affected Version changed from All to 2.2.x

Workaround applied and confirmed to fix subject issue.