strongSwan - Bug #1229

connmark module, wrong rule deleted

08.12.2015 20:37 - Saso Slavicic

Status:	Closed	**Start date:**	08.12.2015
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.4.0		
Affected version:	5.3.5	**Resolution:**	Fixed

Description

When two clients behind same NAT connect with connmark module active, two similar rules with different marks are inserted into iptables.

When the first client disconnects, the second client will loose connectivity because the first rule in the chain (last inserted) will be deleted instead of the correct client rule.
This happens because rules are very similar and have to be matched across all fields to find correct rule.

A sample patch to fix the issue is attached. Tested on OpenWRT.

Associated revisions

Revision 7c9e7eb9 - 10.03.2016 17:26 - Tobias Brunner

connmark: Compare the complete rules when deleting them

By settings a matchmask that covers the complete rule we ensure that the correct rule is deleted (i.e. matches and targets with potentially different marks are also compared).

Since data after the passed pointer is actually dereferenced when comparing we definitely have to pass an array that is at least as long as the ipt_entry.

Fixes #1229.

Revision 7d22a75b - 10.03.2016 17:26 - Tobias Brunner

forecast: Compare the complete rules when deleting them

Same as the change in the connmark plugin.

References #1229.

History

#1 - 09.03.2016 12:22 - Tobias Brunner

- Tracker changed from Issue to Bug

- Category set to libcharon

- Status changed from New to Feedback

- Assignee set to Tobias Brunner

- Target version set to 5.4.0

- Resolution set to Fixed

I can't really reproduce the issue in the ikev2/host2host-transport-connmark scenario (different kernels from 3.8.1 to 4.4.3 and iptables-dev 1.4.14-3.1). I can terminate either CHILD_SA and the correct firewall rules are removed.

I haven't found much information regarding the *matchmask* parameter for iptc_delete_entry() so I had a look at the *libiptc* sources and we should definitely set it to an array that is at least as long as the passed entry because data after the passed matchmask pointer is actually dereferenced. Whether that works or triggers this issue (or a crash) probably depends on the compiler and the data after the static string that was passed so far. I pushed a slightly modified patch to the *1229-1230-connmark-fixes* branch.

#2 - 10.03.2016 17:25 - Tobias Brunner

- Status changed from Feedback to Closed

Files

iptc_delete_entry.patch	426 Bytes	08.12.2015	Saso Slavicic