pfSense - Feature #6130

Alias-table failures can easily lead to serious security degradation should be caught

04/13/2016 03:25 AM - B. Derman

Status:	Duplicate	**Start date:**	04/13/2016
Priority:	Very High	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Rules / NAT	**Estimated time:**	0.00 hour
Target version:			
Plus Target Version:		**Release Notes:**	

Description

Failures that result in empty alias-tables being created (e.g., https://redmine.pfsense.org/issues/6119) or tables failing to be created (e.g., https://redmine.pfsense.org/issues/4513) are not detected.

Aliases are seriously useful in being able to define concepts and create much more "human consumable" rules. The increased clarity helps reduce complexity (well, to the User, anyway) and errors and thus aids security by helping ensure correct configurations.

Alias-table failures, by definition (pun intended), cause loss of functionality and, depending upon that functionality, can cause significant loss of security -- which is a prime purpose of pfSense.

As indicated in issue 6119, we had a device modified because of the loss of security due to this kind of failure. While it wasn't catastrophic, it easily could have been.

It would be much nicer (and safer) if these kind of failures were caught by pfSense. E.G., something as "simple" as warning when tables are defined (and used in a rule) but are missing or empty would really have helped with issues 6119 and 4513.

History

#1 - 04/14/2016 04:27 PM - Chris Buechler

- Status changed from New to Duplicate

duplicate of several alias-related validation issues fixed in 2.3

#2 - 04/15/2016 03:48 PM - B. Derman

But this wasn't about the issues with creating an alias table (which is why it's a separate report). This is about having a sanity check that's run every time there's an update that would alter/(re)define the alias tables.

This is important due to the fact that such failures/bugs (which, we can see, there have been ... and likely will continue to be -- hey, it's software) are critical to both the functionality of pfSense and, therefore, the security of the using organization's devices.

I propose that any such issues and resulting failures are potentially so critical to security that there deserves to be (at least) some sanity checking performed that will catch (at least) the "empty table" and "missing table" cases that result from any alias/rule-creation issues. I'm guessing that such a check is not even that difficult, especially in relation to its value.

The alternative is that you're exposing pfSense to a failure that could have spectacular (negative) results and publicity for pfSense.

In my case, since I no longer have any confidence in this aspect of pfSense (having suffered 2 different failures), I now am compelled to spend the time to check each of the (currently) 112 tables every time I make a change ... a significant cost, in time.

Don't get me wrong, I think pfSense is a great product and that y'all do an exemplary job.

It's just that it seems obvious that this is a critically important issue that needs to be addressed ... assuming you care about security.

#3 - 04/15/2016 04:15 PM - Chris Buechler

B. Derman wrote:

> But this wasn't about the issues with creating an alias table (which is why it's a separate report). This is about having a sanity check that's run every time there's an update that would alter/(re)define the alias tables.

That's precisely what I changed in 2.3, those alias contents, when updated, are fully validated. Rules specifying aliases that don't exist aren't included in the ruleset (which has been true for a number of years with the exception of a ports URL table alias edge case fixed in 2.3).

If you're aware of a specific circumstance that doesn't work in 2.3, please provide specifics to replicate, I'll fix it. I see no indication that's the case though.