pfSense - Bug #8285

Actions on stale data may result in catastrophic results

01/16/2018 08:08 PM - Mahmoud Al-Qudsi

Status:	New	**Start date:**	01/16/2018
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Web Interface	**Estimated time:**	0.00 hour
Target version:			
Affected Version:	2.4.x	**Affected Architecture:**	

Description

It seems that a number of pages in pfSense use links that specify only the index of an item in its category rather than a unique id, and that indexes are dynamically generated and not monotonically increasing (a la autoincremented int primary keys). In addition, there is no nonce embedded including a digest of the current state, meaning pfSense has no way to tell if a request came from a view that contained stale data.

As a simple example, assume a user opens two the firewall rules list in two tabs/pages. The current rules are as follows:

- rule1: rule to be deleted
- rule2: critical rule that must be kept in place

If the user deletes rule1 in the first tab, then navigates to the second tab some time later and sees the rule that should be deleted, he/she will press "delete" on "rule1", but "rule2" will be deleted since pfSense has now assigned it index 0, which is what the delete link and POST content specify.

This does not just affect cases where a user opens two pages, if there are two admins logged on to pfSense at once and one deletes a rule, user2 trying to delete the same rule will inadvertently delete the wrong rule. After a cursory check, it seems that most actions that affect list elements are vulnerable to this issue.

Solution: use unique internal id of rule in all operations, or include a digest of the current state either as a separate (mandatory) parameter or as a component of the csrf.