pfSense Packages - Bug #11777

Input validation prevents DNS Resolver from being disabled

04/03/2021 05:51 PM - Martin Thygesen

Status:	New	**Start date:**	04/03/2021
Priority:	Very Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Unbound	**Estimated time:**	0.00 hour
Target version:			
Affected Version:	2.5.0	**Affected Architecture:**	amd64

Description

When trying to disable unbound, the following error prevents the service from being turned off.

The following input errors were detected:

```
The generated config file cannot be parsed by unbound. Please correct the following errors:
    [1617488979] unbound-checkconf[18250:0] fatal error: outgoing-interface: fe80::20c:29ff:fec7:63f1%em0 present twice, cannot bind same ports twice.
-----------------
```

taxonomy
user was operating with unbound dns resolver normally
system was setup for network interfaces LAN & DMZ & LOOPBACK (v4 & v6)
system was setup for outbound network interfaces WAN (v4 & v6)
user installed bind package to replace unbound but did not activate it.
user tried to disable unbound and was presented with this error message.
user stopped the unbound service from the dashboard and retried to disable the configuration, outcome failed

Workaround:
user adjusted the network interfaces to loopback
user adjusted the outbound network interface setting to loopback
user saved the config
user disabled unbound, and was successful in disabling the service.

Recommendation:
remove some of the check conditions that prevent the service config from being disabled
the service can clearly be stopped without the configuration process being impacted.
the parse of the configuration is too restrictive in this case.

History

#1 - 04/03/2021 05:57 PM - Martin Thygesen

sorry noob mistake name= services_unbound.php : unbound dns resolver error on disable
affected architecture is amd64 however should read : all/undefined

the process of installing the bind service first seems co-incidental and not the root cause of the parse issue during the disable process.

#2 - 04/05/2021 08:21 AM - Jim Pingle

- Subject changed from unbound dns resolver error on disable to Input validation prevents DNS Resolver from being disabled

- Priority changed from Normal to Very Low

This is kind of a tricky situation since someone may want to work on their DNS Resolver configuration while it's already disabled, so they can be sure it's correct before enabling it. Thus, it's good to have the configuration validated while the service is disabled. Otherwise someone could think they have it setup right but then be surprised when they flip the switch to turn it on.

I don't think it would be a good idea to allow it to be saved as disabled with an invalid configuration, but I'll leave it open for now in case others have an idea on how to handle it differently.

#3 - 04/05/2021 05:51 PM - Martin Thygesen

Jim Pingle wrote:

> This is kind of a tricky situation since someone may want to work on their DNS Resolver configuration while it's already disabled, so they can be sure it's correct before enabling it. Thus, it's good to have the configuration validated while the service is disabled. Otherwise someone could think they have it setup right but then be surprised when they flip the switch to turn it on.
>
> I don't think it would be a good idea to allow it to be saved as disabled with an invalid configuration, but I'll leave it open for now in case others have an idea on how to handle it differently.

I think the core of the issue is that overall when someone is trying to disable the service, these checks are stopping them from completing that task
This is the higher level task the validation on disable seems to be blocking this.
There is no doubt that validation at various states is needed however is it needed on start or on exit or during ever save event?
Validation on exit seems to be superfluous to the task at hand
Lack of knowledge of the full workflow here makes it unclear if the validation is linked to a save or other action that is trigger co-incidental the disable process.

The questions is "does that really need to happen then and there?", given that the validation will likely be carried out again on starting the service as well as when viewing the configuration in the UX during a load/edit event.