Config blocks

In order to parse parts of a config, configs are split up into blocks. Conditions with the type ConfigBlock will then parse against these blocks. These blocks can be selected by the Rule_start and Rule_end properties of the rule. These string can also be regular expressions. If multiple blocks match, all of them will be evaluated for compliance. Rule_start will match the first line of the block.

In general, config blocks are split up based on indentation. Also logical block ends are empty lines or lines only containing a ! or a #. Blocks can be hierarchical, so blocks within blocks will also work. In this case the block also has a path, which consists out of its and all of its parents' first lines concatenated together, which you can select with your Rule_start.

Junos

Junos configs are heavily indented and therefore their blocks will be very hierarchical and quite a lot of sub blocks all start with the same text. For this, you should select the blocks you want to check by their paths.

Ciena

Ciena configs contain blocks like:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! RCOS QUEUE MAP CONFIG:
!
traffic-services queuing queue-map create rcos-map NNI-NNI
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 1 queue 1
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 2 queue 2
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 3 queue 3
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 4 queue 4
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 5 queue 5
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 6 queue 6
traffic-services queuing queue-map set rcos-map NNI-NNI rcos 7 queue 7
```

Blocks can be matched against their title (in this case RCOS QUEUE MAP CONFIG). Also there is no hierarchy.

Checkpoint

Checkpoint configs lack any sort of indentation or logical spacing. Instead the parser looks at the starting keywords, and groups blocks together, regardless of whether they are preceded by “add” or “set”.

So for example the following piece of config:

```
set inactivity-timeout 10
```

```
set expert-password-hash $1$cBBBDBBW$FmeO/rhfGDhZpHlKM4ROO1
set user admin shell /bin/bash
set user admin password-hash $1$R5wwe24I$8mFvR4y7rxuwVIDBcI6E/.
set user monitor shell /etc/cli.sh
set user monitor password-hash *
```

Will be split up like:

```
set inactivity-timeout 10

set expert-password-hash $1$cBBBDBBW$FmeO/rhfGDhZpHlKM4ROO1

set user admin shell /bin/bash
set user admin password-hash $1$R5wwe24I$8mFvR4y7rxuwVIDBcI6E/.

set user monitor shell /etc/cli.sh
set user monitor password-hash *
```

And the following piece with interfaces:

```
set timezone America / New_York
set interface eth0 state on
set interface eth0 auto-negotiation on
set interface eth0 ipv4-address 192.168.178.40 mask-length 24
set interface eth1 state off
set interface eth2 state off
set interface eth3 state off
set interface lo state on
set interface lo ipv4-address 127.0.0.1 mask-length 8
```

Will be split like:

```
set timezone America / New_York

set interface eth0 state on
set interface eth0 auto-negotiation on
set interface eth0 ipv4-address 192.168.178.40 mask-length 24

set interface eth1 state off

set interface eth2 state off

set interface eth3 state off

set interface lo state on
set interface lo ipv4-address 127.0.0.1 mask-length 8
```

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:reference:compliance:cmpl_parsing

Last update: **2024/07/03 12:31**