NCCM Syslog

In order to detect a config change NetYCE deploys a syslog server that listens to network events. Since each node, but certainly the network as a whole, can issue large amounts of syslog messages, these events need to be filtered. Only the events indicating a configuration change was made is of interest to the NCCM.

Syslog messages can originate directly from each device if it is configured with the NetYCE server(s) as target. It is common practice to have the device send syslog messages to multiple servers using the udp protocol. When NetYCE servers form this target the NCCM process will deduplicate these messages and retrieve the changed configuration only once.

In many existing networks the syslog servers are already part of the network monitoring solution and all nodes have these addresses configured as their syslog targets. To allow NetYCE to receive the syslog messages it needs to trigger the NCCM a forwarding rule needs to be activated on the existing syslog receivers to the NetYCE server(s). In most cases this forwarding can also incorporate a filter to reduce the number of syslog messages (eg by dropping 'debug' priority messages). This route normally also has the side effect of changing the format of the syslog messages forwarded.

Delays

When a device signals its configuration has changed The NCCM will not immediately be triggered to retrieve its configuration. Assuming an operator using the CLI on the device made this change, we postpone scheduling the configuration retrieval by 10 minutes to allow the operator to finish his session.

After these 10 minutes the retrieval will be scheduled within the next 5 minutes to be fetched by the NCCM. However, if the NCCM is too busy handling all requests within that 5 minute batch, the request will be re-scheduled for the next batch. The NCCM is designed to take advantage of multiple NetYCE servers that can perform NCCM tasks. The `etc/sched_rules.conf` configuration file can be setup with rules to direct the NCCM to those servers that can interact with the device at hand. If multiple servers apply, the load will be distributed.

Al in all, allow for 10-15 minutes for the NCCM to complete when triggered by syslog.

For configuration changes initiated by NetYCE jobs, the NCCM will be updated immediately by the job itself. The resulting syslog messages will be ignored as they are processed within the 10 minute window.

YCE Events daemon

The task of filtering, deduplication and detecting a configuration change message is built into the daemon process **yce_events**. This daemon is controlled by the configuration file `etc/yce_events.conf`. This file can be modified to update the message patterns as received from the network when needed. The distribution version of this file `system/yce_events.conf` will be

copied to its operational location when missing.

The `etc/yce_events.conf` file has for each supported vendor a configuration section. The various sections only differ in the **`pattern`** as it is a regular expression (regex) that must match the received syslog message indicating the configuration was changed. If a vendor uses distinctly different messages to indicate the change, then multiple sections can be included.

```
#
# Juniper
#
type=SingleWithSuppress
ptype=RegExp
name=Junos
pattern=.*\s(.*)\smgd\[\d+\]:\sUI_COMMIT_PROGRESS:(.*)commit\scomplete
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# F5 BIGIP
#
type=SingleWithSuppress
ptype=RegExp
name=F5_BigIP
pattern=[\w]{3} [\d]{1,2} [\S]{8} (\S*) notice [\w]+\[\d+\]: .* AUDIT - .*
status=\[Command OK\] cmd_data=save .* config
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# HP_C7 normal save or save main
#
type=SingleWithSuppress
ptype=RegExp
name=HP_C7
pattern=[a-zA-
Z]{3}\s{1,2}\d{1,2}\s\d{2}:\d{2}:\d{2}\s(.*)\s%%10SHELL\/6\/SHELL_CMD_CONFIR
M:\sConfirm\soption\sof\scommand\ssave\s
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# HP_C7 normal save main force or save force
#
type=SingleWithSuppress
ptype=RegExp
name=HP_C7_b
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{2}:\d{2}:\d{2}\s(.*)\s%%10SHELL\/6\/SHELL_CMD:\s.*Command
\sis\ssave\s
desc=config save for $1
action=event config_changed_for_$1
```

```
window=600
#
# Arista_EOS
#
type=SingleWithSuppress
ptype=RegExp
name=Arista_EOS
pattern=[a-zA-Z]{3}\s\d{1,2}\s\d{2}:\d{2}:\d{2}\s(.*)\sConfigAgent:\s%SYS-5-
CONFIG_STARTUP:\sStartup\sconfig\ssaved
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# Cisco_Nexus
#
type=SingleWithSuppress
ptype=RegExp
name=Cisco_Nexus
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{2}:\d{2}:\d{2}\s((\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3
}))\s:\s\d{4}\s[a-zA-Z]{3}\s\d{1,2}\s\d{2}:\d{2}:\d{2}\s[A-Z]{3}:\s%VSHD-5-
VSHD_SYSLOG_CONFIG_I:\sConfigured\sfrom
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# Cisco_IOS
#
type=SingleWithSuppress
ptype=RegExp
name=Cisco_IOS
pattern=[a-zA-
Z]{3}\s+\d{1,2}\s\d{2}:\d{2}:\d{2}\s((\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,
3}))\s(\d+){0,1}:\s+(\*[a-zA-
Z]{3}\s+\d{1,2}\s\d{2}:\d{2}:\d{2}(:|.\d{3}:)\s){0,1}%SYS-5-
CONFIG_I:\sConfigured
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# HP_C5 normal save, save main, save main force or save force
#
type=SingleWithSuppress
ptype=RegExp
name=HP_C5
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{2}:\d{2}:\d{2}\s(.*)\s%%10CFM/5/CFM_SAVECONFIG_SUCCESSFUL
LY\(l\):\sConfiguration\sis\ssaved\ssuccessfully
desc=config save for $1
action=event config_changed_for_$1
window=600
```

```
#
# Avaya_ERS save
#
type=SingleWithSuppress
ptype=RegExp
name=Avaya_ERS
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\s(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\
d{1,3})\s\d{2}:\d{2}:\d{2}:\d{2}\s(.*)\s:Trap:\s\sbsnConfigurationSavedToNvr
am
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# CI_6 save configuration
#
type=SingleWithSuppress
ptype=RegExp
name=CI_6
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\s((\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(
\d{1,3}))\s\[[a-
z]{5}\]\s((\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3}))\s([0-9A-Fa-f]{2}[:-
]){5}([0-9A-Fa-f]{2})\s\d{4}\sCONFIG-5-CONFIG_SAVE:
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# Aruba_MC write memory
#
type=SingleWithSuppress
ptype=RegExp
name=Aruba_MC
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\s(.*)\s<.*\sCOMMAND:<write\smemory>\
s--\scommand\sexecuted\ssuccessfully
desc=config save for $1
action=event config_changed_for_$1
window=600
#
# Alcatel_OmniSwitch
#
type=SingleWithSuppress
ptype=RegExp
name=Alcatel_OmniSwitch
pattern=[a-zA-
Z]{3}\s\d{1,2}\s\d{1,2}:\d{1,2}:\d{1,2}\s(.*)\sCLI\(67\).*write\smemory.*
desc=config save for $1
action=event config_changed_for_$1
window=600
```

Each vendor has its own pattern(s) to filter against. Note that all of these patterns are regexes, and the first pattern matched into the parentheses (( )) needs to be the node's hostname. For those nodes that do not include the hostname in their syslog messages (F5's Bigip is an example of this), the IP-address is needed.

The distributed version of the configuration file has the patterns for the direct syslog message. Depending on the forwarder this pattern must be updated to reflect the modified message.

As the NCCM stores the configurations using the device hostname, the IP-address will be used to do a reverse DNS lookup on that IP-address. Should the DNS provide no results, the NCCM will not be able to retrieve the configuration (without a node name the login credentials and vendor-type will be unknown).

Once a configuration change message is processed, the node's entry in the “Nccm selection” will be modified, most notably its next poll time will be set to ten minutes from now (to allow the node to finish with everything it's doing), provided the node hasn't been disabled. The NCCM will detect the update and perform its nccm poll and where applicable, its compliance check. For more information regarding Compliance take a look at the Compliance userguide