syslog-alert

a notification program for throttling per host syslog-ng messages with flexible mailbox selection

Nic Boet

https://github.com/nabbi/syslog-alert

Motivation & Goals

- syslog-ng OSE smtp() introduced in version 3.4 ~2014?
  - my first Perl prototype 2009, TCL rewrite began 2017
  - lacks native throttling support
- per host message throttling
  - run on either centralized log collector or standalone host
  - avoid flooding mailboxes
- selectively notify specific or multiple support groups
  - send shortened emails to pagers or mobile phones
- exclusion filters for false alarm noise
- replace commercial product running on MS Windows
  - lacked per host throttling, suppressed all events
- learn TclOO - it's not a purist sample ;)

Overview

1. syslog-ng.conf template() export wraps syslog message variables as TCL List data structure
   `{${LEVEL}} {${MSGHDR}} {${MSG}}`
2. Reads standard input from syslog-ng program() driver
3. Matched glob pattern queries *SQLite events table* to determine if an event should be suppressed or generate an alert
4. New or alertable events then query *SQLite contacts table* for sendmail recipients, and records event hash and timestamp for subsequent throttling

Contacts: Config file to DB

Config imports mailbox as “groups” to SQLite :memory: table

TCL List: {name} {group} {email} {page}

```
{Moon Cat} {disk} {moonkitten@example.com} {}
{Patch} {admin} {patch-work@, patch-home@} {patch-mobile@}
```

Database read when alert pattern triggered, assembles emails or pagers-mobiles for sendmail recipients.

```
foreach g “admin oncall”
 {SELECT * FROM contacts WHERE "group"=:g}
 return [join $results ", "]
```

Event logs: Configuration File

Defines glob patterns and exceptions, unique hashes, throttling intervals, and groups to notify when processing stdio events from syslog-ng program()

TCL LIst: {{pattern1} {pattern2}} {{exclude1} {exclude2}} {**hash**} {delay} {email} {page} {ignore} {eval custom tcl code action}

```
{{"*mdadm*"}} {{msg="*/dev/md0*"}} {"$log(host) raid event"} {86400} {admin disk} {oncall} {0} {}

{{"*alert*"} {"*crit*"}} {} {"$log(host) $log(level) sendmail subject"} {3600} {admin oncall} {} {0} {}
```

Event logs: TCL switch{}

TCL conditional switch body is dynamically constructed as a new TclOO method from reading configuration file

```
append new "oo::define Alert method patterns {log} {"
append new "switch -glob -- $log(all) {[$oo my Import] }}"
eval $new

while [gets stdin line] >= 0
 $oo patterns $line
```

Event logs: Throttle hash

Assigning an unique "hash" for pairing with the pattern establishes the per host+event throttling.

Pattern: "*BLOCK_BPDUGUARD*"
Hash: "$log(host) BPDU Block"

```
site-one %SPANTREE-SP-2-BLOCK_BPDUGUARD: Received BPDU on
port GigabitEthernet2/0/30  with BPDU Guard enabled.
Disabling port.
```

```
site-two %SPANTREE-SP-2-BLOCK_BPDUGUARD: Received BPDU on
port GigabitEthernet1/0/1  with BPDU Guard enabled.
Disabling port.
```

Event logs: Throttle database

SQLite :memory: table tracks the $hash and timestamp

Any $pattern messages received within $recent threshold are discarded, otherwise a new alert is generated and time recorded

```
if [Db exists {SELECT 1 FROM alert WHERE hash=:hash}]
if [expr {$now-$time}]  > $delta
 {UPDATE alert SET time=:now WHERE hash=:hash}
 return true
```

Event logs: Customizable

Extends functionally by evaluating tcl code from config file, injected before sendmail exec

Overrides email subject (default subject is $hash)

```
{{default}} {} {"$log(host) $log(msg)"} {3600} \
  {admin} {} {} {set subject "$log(host) event"}
```

run an external script (which is also throttled!)

```
{"*trigger*"} {} {"$log(host) script exec"} {3600} \
  {admin} {} {} {exec /usr/local/bin/something-cool.sh}
```

Thank you

syslog-ng + TclOO + SQLite + Sendmail

https://github.com/nabbi/syslog-alert