syslog-alert

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

Nic Boet

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

S&T2020

The SQLite & TCL Conference

Motivation

- per host message throttling
  - centralized log collector or can run on single host
- exclusions to filter out false alarm noise
  - events should be pre-filtered within syslog-ng
- selectively alert different or multiple support groups
- send shorter text messages to pagers or mobiles
  - Ie Syslog message without all the headers
- configuration files define patterns
- syslog-ng OSE smtp() driver in version 3.4 ~ 2014?
  - first Perl prototype 2009, TCL rewrite 2017
- learn TclOO - it’s not a purist sample ;)

Overview

1. Adjust template() within syslog-ng.conf to wrap variables as TCL List data structure {${LEVEL}} {${MSGHDR}} {${MSG}}
2. Reads standard input from syslog-ng OSE program() driver
3. Matched glob patterns queries SQLite to determine if an event has previously triggered an alert within the specified time period
4. New events query SQLite for sendmail recipients, and also updates when this message first alerted at

Contacts: Configuration

Defines which email and pager-mobile addresses are associated to a “group”

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

```
{Moon Cat} {disk} {moonkitten@example.com} {}
{Patch} {admin} {patch-work@, patch-home@} {patch-mobile@}
{Apricat} {admin} {apricat@} {apricat-pager@}
{Team Fur} {network} {snowball@, snowball2@, oreo@} {}
{Litterbox} {oncall} {distgroup@} {opager@, mobile@}
```

Contacts: Database

Populates into SQLite :memory: table

Read when a new alert is triggered to assemble which emails or pagers-mobiles are to be used for sendmail recipients.

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

Event logs: Configuration

Defines glob patterns for how stdio events from syslog-ng are to be processed. Associates an unique hash, throttling interval, and which group to notify.

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: Throttle hash

Decide what your unique “hash” will be for pairing the pattern with your alert interval. This 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: TCL switch

Dynamically construct the switch conditions body from config 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 database

SQLite :memory: table tracks the $hash and timestamp

Any $pattern messages received within the $recent threshold are discarded
Otherwise a new alert event is generated and inserted into the database

```
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
```

S&T 2020

Thank you

More information:

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

Bonus for any Gentoo Linux users out there, I’ve also published a few unofficial packages to my Oubliette overlay.

https://github.com/nabbi/oubliette-overlay