

LOGX v0.13.x

A Flexible Log Exchange Utility

泛用訊息交換 / 收集 / 儲存工具

<http://www.LinuxDAQ-Labs.com/products/logx/>

Kuang-Chun Cheng

kccheng@LinuxDAQ-Labs.com, kcc1967@gmail.com



Overview

- Source (訊息來源)
- Filter (訊息過濾器)
 - 可用於訊息本身或 column value (DB, CSV, NVP) 的過濾，處理。
- Target (訊息目的地)



Overview

- Route (訊息處理流程安排)
 - 每一條 routing rule 會啟動一個獨立的監控單元 (Thread).
Logx 可同時監控多組的監控單元.
- 支援 Linux, Win32





Source



Source

- STDIN (text console)
 - 透過 STDIN, 支援命令列 debug 命令
- TCP/UDP
 - 接收來自 TCP/UDP 訊息



Source

- Database
 - 接收來自於任何支援 ODBC 資料庫的訊息
 - 多重 Table 讀取能力
 - 可針對個別 column 做 filtering 的功能
 - MySQL, Oracle, MS SQL, PostgreSQL, SQLite3 ...



Source

- File
 - 監控文字訊息檔
 - 支援 File rotation 追蹤 (counter, timed, last modified)
- exec, exec_pipe
 - 外部命令，允許使用者使用任何程式語言撰寫訊息收集監控 scripts. 如 PHP, TCL, Perl, Python ...



Source

- C (.so or .DLL)
 - 允許使用者外掛 .so 或 .DLL 模組以提高系統整體效能。



Source

- Windows 事件
 - Windows 事件收集
 - Windows 事件備份檔
- Rand
 - 亂數產生，內部壓力測試用
- Sysinfo
 - logx 本機系統資訊收集



Source

- Logxinfo
 - logx 資訊收集 (流量 ...)



Database Source

- DB 的 Tables 必須有一個 Table 具有 integer 型態的 key 用來追中目前紀錄在 DB 中 log 的筆數。
- 可同時監控多個資料庫 Table 的欄位
 - 處理一筆 log 散落在多個 table 的狀況
- 個別欄位支援 embedded filters, 可直接於資料後傳之前進行處理。



Filter



Filter

- **Regex (Regular Expression)**
 - 利用 regex 判斷訊息是否後傳
 - POSIX Extended Regular expression
- **exec, exec_pipe**
 - 利用外部 scripts, utilities 對 source 傳來的訊息進行處理



Filter

- C (.so or .DLL)
 - 利用 C 語言撰寫訊息處理模組



Filter

- CSV (Comma Separated Value)
 - CSV 格式訊息特別支援 . 可針對個別欄位再進行處理 . (類似 DB column 的 embedded filter)
- NVP (Name-Value-Pair)
 - 可針對個別欄位再進行處理 . (類似 DB column 的 embedded filter)



Filter

- Word Skip
 - 簡單的 filter 允許跳過 / 砍掉訊息的前幾個 words
- BSD syslog
 - 依 BSD syslog 的 priority, facility 決定是否將訊息後傳。



Target



Target

- **STDOUT/STDERR**
 - 標準輸出與錯誤輸出（顏色區分支援）
- **NULL**
 - 相當於 `/dev/null`, 用於把 `source` 當作 `command` 使用時 ...



Target

- TCP/UDP
 - 將處理過的訊息往其他網路目的地（分析套件或另一個 logx 程式）傳送。
- File
 - 將處理過的訊息以文字檔的模式儲存。



Target

- Database
 - 將處理過的訊息寫入 local 或遠端的資料庫 (透過 ODBC).
- BSD syslog
 - 將處理過的訊息寫入傳統 UNIX 的 syslog.



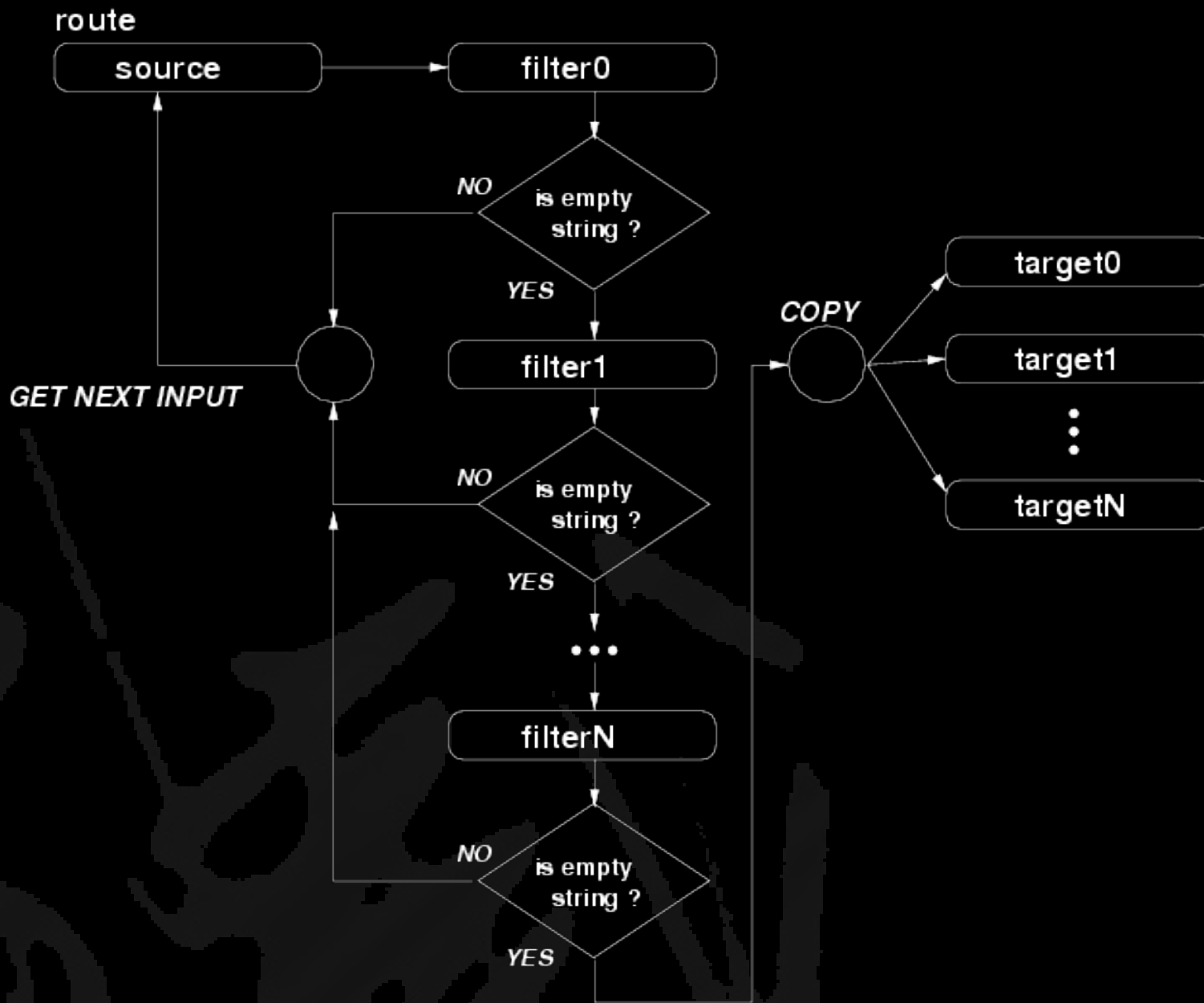
Route



Route

- 針對 Source, Filter 與 Target 做排列組合 .
- `<route name="TEST"
source="rand"
filter0="filter0" filter1="filter1"
target0="STDOUT" target1="DB" />`
- 每一個 `<route/>` 會啟動一個獨立的 thread.





特性

- 除了傳統網路設備（如 Cisco, McAfee, Tipping Point, Fortigate, TrendMicro ...）之外，透過 Logx 可監控到 Applications（如 IIS, Apache, sendmail, BIND, DHCP ...）與 Host 本身的狀態。達到全面監控的目標。
- 可為後端分析套件（如 Novell Sentinel）提供分散式前置事件收集與過濾，以降低分析套件的工作負擔。



特性

- 多重訊息交換管道支援，可適合不同客戶個別的需求。
 - 例如有些客戶不許內部資料透過“網路上的芳鄰”交換，此時可透過 logx 將 File 轉成 syslog 的格式外傳。
- 流量管制
- ...



Examples



UNIX/Linux System Log

/var/log/messages

- ```
<?xml version="1.0" encoding="UTF-8" ?>
<logx require="0.11.3">
 <source name="messages"
 type="file"
 file="/var/log/messages"
 seek_whence="END" seek_offset="0"
 udelay="1000000" />

 <route name="msg"
 source="messages"
 target="STDOUT" />
</logx>
```



# Apache2 Error (OpenSuSE)

/var/log/apache2/error\_log

```
■ <?xml version="1.0" encoding="UTF-8" ?>
 <logx require="0.11.3">
 <source name="apache_error"
 type="file"
 file="/var/log/apache2/error_log"
 seek_whence="END" seek_offset="0"
 udelay="1000000" />

 <route name="apache_error"
 source="apache_error"
 target="STDERR" />
 </logx>
```

