天使からの手紙(Report mail from Tenshi)

～ログ監視ツール「Tenshi」を使う～

国立天文台　チリ観測所　安井　孝

１．はじめに

チリ観測所（三鷹）にあるWebサーバや解析環境を支援する複数のサーバのログはrsyslogを利用して日々リモートホストに蓄積されている。これらのログの中から人手で問題点をもれなく発見することは決して容易な事ではない。

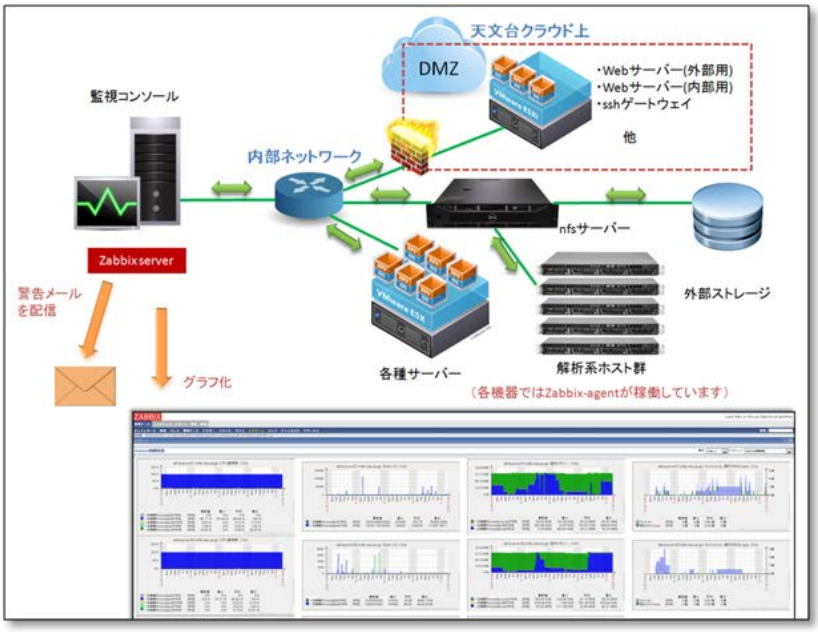
ログデータを効率的に分類・フィルタリングし、レポートメールを配信することができるログ監視ツール「Tenshi」を設定し、現在安定に稼働している。管理の考え方やツール設定のコツについて纏める。

２．対象機材

チリ観測所(三鷹)のComputingチームが管理・運用しているホスト・ハードウェア群の中でDMZ上にあって外部からのアタックなどが予想されるマシンや可用性の面で常時監視が必要な機材について行うこととする。

３．目的とポイント

複数のホストを監視する目的で設置されたホスト(以後:監視コンソール)があり、常時Zabbix-serverを稼働させ、対象のホストに直接インストールされたZabbix-agentと連携して異常がないかを検知している。ただし、すべてのイベントやデータ取得をきめ細かく設定するには限度がある。



これとは別に、各ホストからrsyslogのしくみを使って監視コンソールの特定のフォルダにホスト毎にログファイルを集めている。しかしながら今までは、蓄積される一方で、インシデントが発生したとき以外にはログファイルを十分活用できていなかった。

世の中にはログ管理ツールと呼ばれるものが多数あり、データベースを用いて統合的に処理し、検索機能が充実したものなど手法や目的に応じて様々なものがある。

今年初めから幾つかのツールについて調査し、我々の目的に合ったものを導入すべく検討を始め、いくつかのツールについては試用版をインストールして試した。

１）我々の要件

・複数のホストから出力されたログファイルを扱えること
・あまり作りこみをせず容易に導入可能であること
・緊急度や粒度を考慮して様々なレポートまたはメール配信ができること
・導入コストが可能な限り低いこと

２）検討したログ管理ツールのリスト（一部）

・Logwatch
・LogAnalyzer
・fluentd
・Splunk
・tenshi　　いくつかの候補から最終的に「tenshi」というツールを採用することにした。

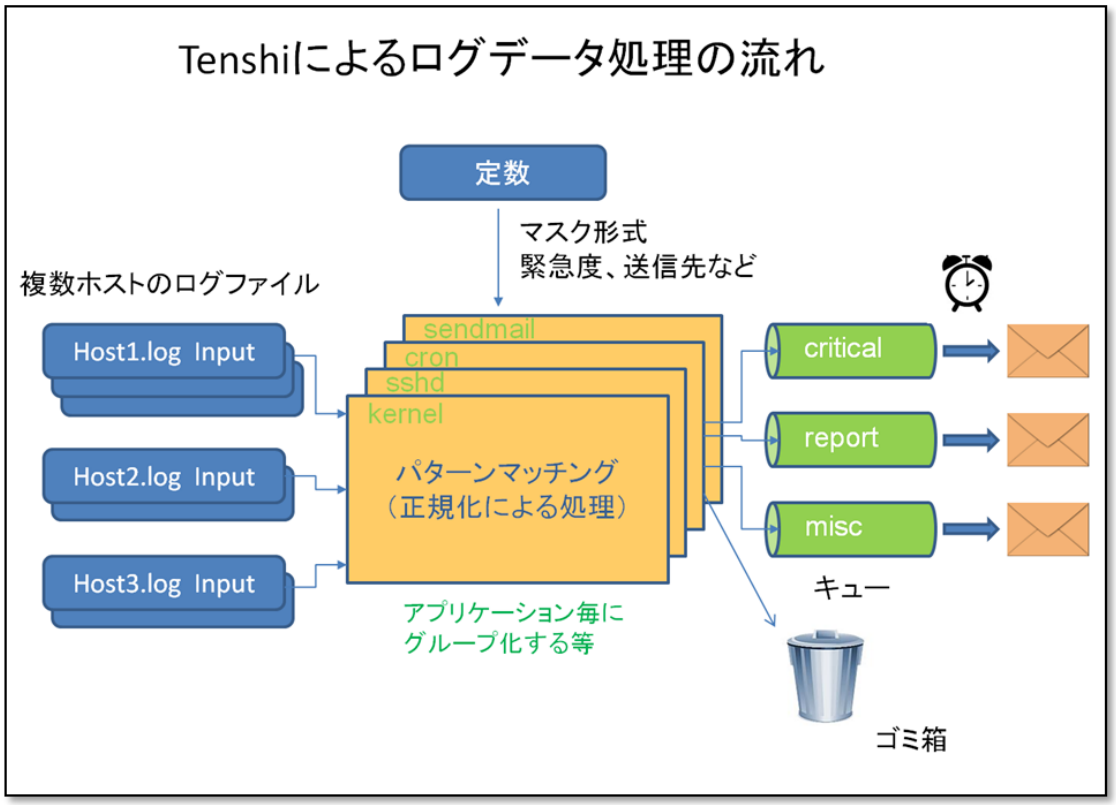
４．tenshiの特徴

システム（ホスト）が出力するログデータを効率的に分類・フィルタリングし、メールを通じてレポートを配信できるソフトウェアである。ソースファイル名を指定することにより複数のホストからのログファイルを同時に処理することができる。また、スタンドアローンモードで1台ずつのホストにtenshiをインストールすることも可能である。

2008年ころから開発されていて（当初はwasabiという名前であった）、何度かアップグレードされていて比較的安定していると考えらえる。Ubuntuのレポジトリに登録されておりapt-getコマンドで容易にインストールが可能である。最新バージョンは0.15　(2014年8月4日）である。

複数のログファイルをtailでウォッチしており、正規表現でユーザーが指定した条件にマッチしたものを幾つかのキューに入れ、設定した時間に指定した相手先にログ内容をまとめて送信することができる（時間設定はcrontabと同形式である）。

ツールはperlで書かれていて許容的フリーソフトウェアライセンスであるISCライセンスに従う。



５．Tenshiの設定

以下の条件を順に設定していくと判りやすい。　　(/etc/tenshi/tenshi.conf) #1

1）　ログファイルの取得
2）　キューの出力
　2.1 キューの種類、重要度
　2.2 メールの配信、宛先、時間間隔
3）　キューの設定
　3.1　分類、フィルタリング（正規表現）
　3.2　マスク
　3.3　ごみ箱
　3.4　未分類
4）運用しながら手直しして収斂させていく

```
## general settings
set uid tenshi
set gid tenshi
set pidfile /var/run/tenshi/tenshi.pid
#set logfile /var/log/syslog
#set logfile /var/log/auth.log
#set logfile /var/log/messages
set logfile /var/log/remote/alma-i***.log
set logfile /var/log/remote/alma-e***.log
set logfile /var/log/remote/alma-s***.log
set logfile /var/log/remote/alma-d***.log
set logfile /var/log/remote/alma***.nao.ac.jp.log
set tail_multiple on
 :
set sleep 5
set limit 800
set pager_limit 2
set mask ___
set mailserver localhost
set subject tenshi report
set hidepid on
set logprefix ^¥S+T¥d{2}:¥d{2}:¥d{2}.¥d{6}¥s¥w+¥.¥w+¥s(¥S+)
```

ログファイルの指定

複数の入力を設定

マスクの形式

(/etc/tenshi/tenshi.conf) #2

```
## queues# syntax: set queue <queue_name> <mail_from> [pager:]<mail_to> <cron_spec>
[<subject>] set uid tenshi
set queue report almaj_sysadm@alma.mtk.nao.ac.jp almaj_sysadm@alma.mtk.nao.ac.jp
[30 12 * * *] Report from Tenshi
set queue misc     almaj_sysadm@alma.mtk.nao.ac.jp almaj_sysadm@alma.mtk.nao.ac.jp
[35 12 * * *] Misc from Tenshi
set queue critical almaj_sysadm@alma.mtk.nao.ac.jp almaj_sysadm@alma.mtk.nao.ac.jp
[now] tenshi CRITICAL report
  :

## regexp definitions# syntax: <queue_name>[,<queue_name>..] <regexp>
repeat ^(?:last message repeated|above message repeats) (¥¥d+) times?
 :
trash ^message repeated (.+) times
trash ^init: tty ¥((.+)¥) main process ¥((.+)¥) killed by TERM signal
critical ^init
 :
#instead of listing all regexps here, we spearated them in multiple ext. files
includedir /etc/tenshi/includes-active

#catch everything else here
misc .*
```

出力の設定

要／不要 緊急度

分類されないものは、最後はここへ

(/etc/tenshi/include-active/ssh.conf)

```
## ssh rules include rules
## (c)2005 by Tadeusz Pietraszek (tadek@pietraszek.org)

 :
report   ^sshd: Did not receive identification string from 133¥.40¥.
trash    ^sshd: Did not receive identification string from
 :
report   ^sshd: fatal: Timeout before authentication for (.+)

#for Host1
trash    ^sshd: User root from n????-???.mtk.nao.ac.jp
trash    ^sshd: Protocol major versions differ for UNKNOWN

#for Host2
trash    ^sshd: Accepted password for root from 133¥.40¥.(?:20*|22*|6*)¥.(.+) port
(.+) ssh2
trash    ^sshd: fatal: Read from socket failed: Connection reset by peer
 :
 :
```

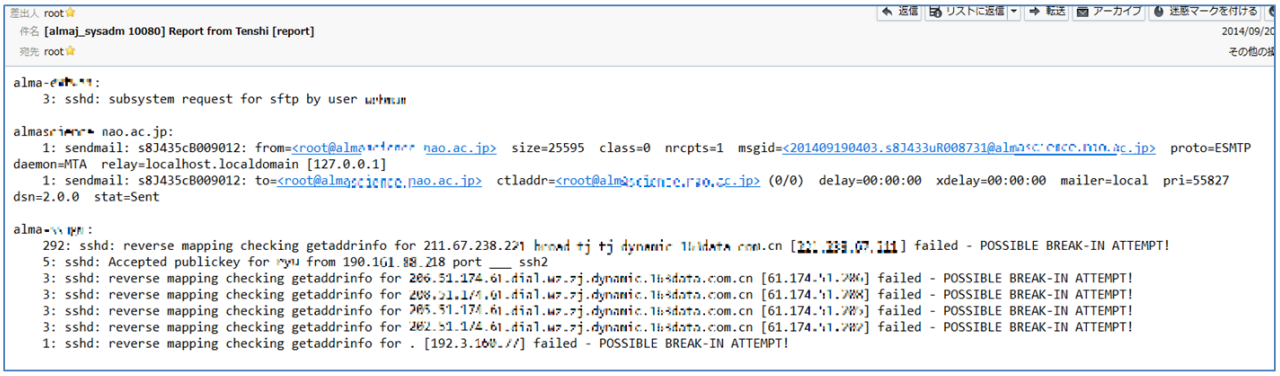
グループ化することにより
種々の対応が可能

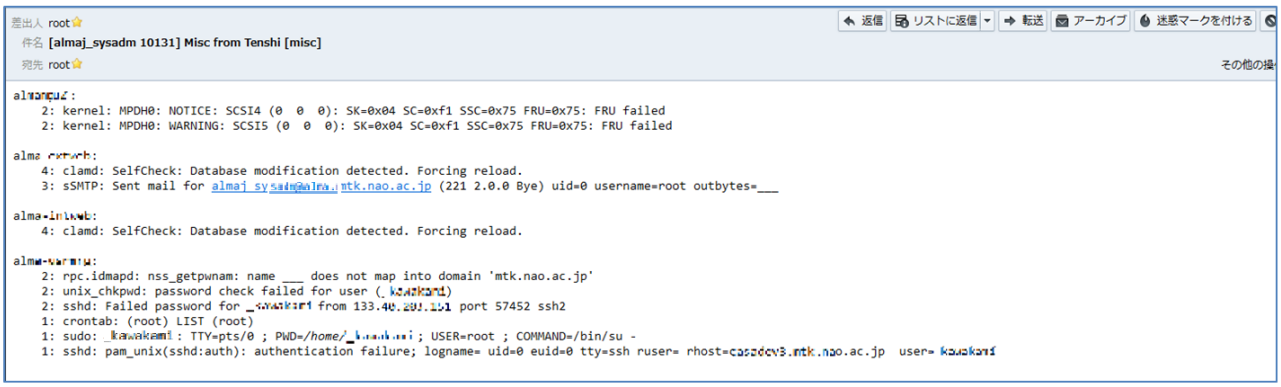
６．レポート出力（メール配信）のサンプル

メールはホスト毎に纏められ、同様なパターンのログがサマリーの形で出力されている。現在我々は緊急度に応じて、次の４つの分類でキューを作成している。

1. critical -- 即座に対応する必要がある可能性のあるもの
2. report -- 1日をサイクルとして監視すべき内容のもの
3. misc -- 未分類または上記2より緊急度が低いが観察が必要なもの
4. trash -- 既知の情報のうちそのまま捨てても構わないもの

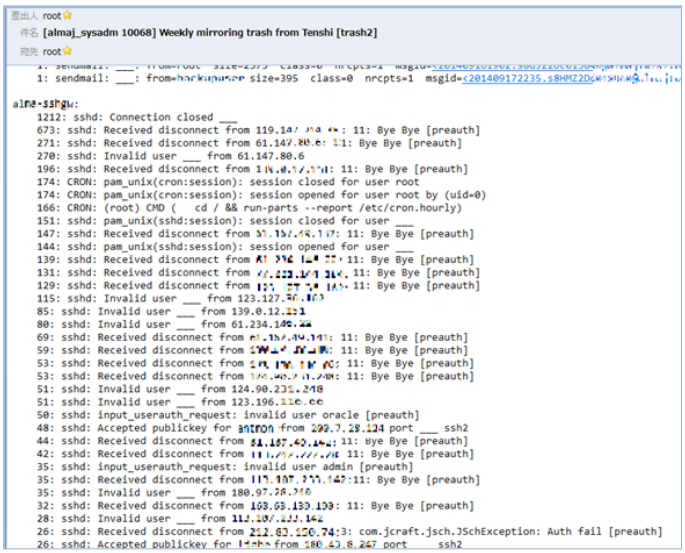
下記は毎日1回送られてくるtenshiからのレポート（report）と未分類（misc）のメールである。





これにより、外部からの脅威やホストから不自然なメールの送信がなかったかどうかなどをチェックすることができる。不要なログの内容をごみ箱（trash）のキューに流して溜めることなく捨てることができるが、実際にどのような内容のものが捨てられているのかを知るため、1週間分をtrash2というキューを作ってそこへ一時的に溜めることで、再確認することができるような工夫をしている。次ページの図はtrash2を廃棄する前の状態を示しており、平常時の1週間にsshを用いた接続がトータルで何回あったとか、DMZ上にあるゲートウェイサーバに対して繰り返し様々な不正アクセスが何度も試みられている様子などがよく判る。

７．まとめ



短いマニュアルの他はTenshiについて余り情報がなく、かなり手探りで始めなければならなかった。しかし、試行錯誤を繰り返しながら設定ファイルをいろいろと弄っているうちに、このツールの面白さが段々と判ってきて、かなり奥が深いものであることに気づかされた。マスクを利用して不要な情報を整理したり、正規表現を使ったフィルタの記述と掛けるフィルタの順序を変えたりすることで、工夫次第でいろいろな動作をさせることができるのである。

インストール自体は数分で終了するのだが、稼動させながら少しずつ設定ファイルを変更してゆく方針で行っていったこともあり、設定が収束して落ち着くまで約２～３週間の期間を要した。現在は見慣れない新しいメッセージが出たときにどのように処理するかを考え、珠に修正するだけであり、安定して稼動していると言える。

世の中に有償／無償を含め、ログ管理ツールと呼ばれるものは多く存在するが、規模や用途により様々なものがあり、どれが一番いいかと結論付けることはできない。DNZにあるサーバ群は常に様々なセキュリティ上の脅威にさらされており、新しい脆弱性をついたゼロデイ攻撃などにより対応しなければいけない仕事が増え続けている。ディジタルフォレンジクスの観点からログの管理も重要なのは勿論であるが、日夜、丁寧に綿密な監視を続けていくには人的リソースの面から難しい。そういった面で一層の効率化が要求されている。複数のホストから集められたログファイルを処理するという当初の目的は達せられているが、設定ファイルの見直しも必須の作業である。また、何でもかんでも１つのところに集約されればいいというものでもない。同僚が「卵は一つのカゴに盛るな」という格言があることを教えてくれた。元々は相場から出たものなのでリスクを分散して投資をせよという意味だと思うのだが、同様のことがシステムの危機管理にも通じることだと思われる。別のツールで管理すべきマシンもあるだろうし、分散して管理した方がシステムダウンなどがあったときに全部駄目になることもない。他の管理ツールとの併せ技で必要な要件を実現させていけば良いのだ。場合によっては監視ツールを監視することも起こりうる。

サーバなどホスト型のハードウェアだけでなくネットワーク機器などログファイルを記録あるいは転送できるものであれば、そのログ情報をこのtenshiを使って処理すればレポートメールを送信することができるので、まだまだ適用範囲は広くなると考えられる。今回このツールがたまたま我々の目的にあっていただけなのかも知れないし、欲を言えばいろいろな機能をつけて拡張していくこともできるだろう。しかし敢えてそれをせず、軽い負荷で動作するところにこのツールの魅力がある。単純そうに見えて実は奥の深いこのツールに出会えて良かったと思っている。

とかく運用・管理の仕事は無味乾燥なものになりがちである。しかし最近、昼休みに2通と週末に1通、天使から来る手紙を開封する楽しみが増えたと言っても過言ではない。

参考：

tenshiツールは以下のサイトから

Tenshi - Inverse Path - Research

<http://www.inversepath.com/tenshi.html>