Platform Independent Event Correlation Tool for Network Management

Risto Vaarandi

Platform Independent Event Correlation Tool for Network Management

Risto Vaarandi
Department of Computer Engineering, Tallinn Technical University, Estonia
risto.vaarandi@eyp.ee

Abstract

Event correlation plays an important role in today's network management, reducing large amounts of network events to smaller and more meaningful sets of alarms. Most of the commercially available event correlation tools have a design that is over-complicated for majority of small and medium range applications, all of them are platform dependent, and last but not least, they are expensive for academic use. This paper presents a lightweight open source network management tool called *sec* designed to implement platform independent event correlation.

1. Introduction

For large managed networks with hundreds or thousands of managed objects, the amount of events presented to the network operator often becomes too large for the fault causes to be identified quickly. Especially undesireable situation is the *event storm* – a flood of events triggered by a single hardware, software, or network failure. In order to reduce the amount of events seen by the operator and to cope with various kinds of event storms, event correlation is used. Events can be correlated centrally at network management servers, locally in agents by processing agent output, or locally at event source, e.g., in managed network element itself.

Though a number of powerful event correlation systems are available on the market, they tend to be quite expensive. Commercial systems are also platform dependent - customers are supplied with program binaries that run on a limited number of operating systems. Furthermore, several commercial systems have been designed for one particular network management platform only. Since a lot of research has been done in the field of event correlation recently, some experimental correlation engine prototypes have been created, but most such prototypes are not publicly available on the Internet. Although many excellent open source network management solutions exist [1], there is no freeware correlation engine available yet which would be mature enough for use in a production environment.

In this paper the author presents a lightweight platform independent tool for rule-based event correlation called *sec* (Simple Event Correlator). The primary design goal of *sec* was to create an open source tool that could be used for both central and local event correlation, regardless of the underlying operating system, and that could be integrated into an arbitrary network management system.

2. Description of sec

Sec is a rule-based event correlation tool that receives its input events from a file stream, and produces output events by executing user-specified shell commands. Regular files, named pipes, and standard input are currently supported as input. To be able to handle input events regardless of their format, *sec* uses regular expressions for recognizing them. *Sec* regular expressions can match complex patterns spanning over several input lines.

To achieve independence from operating system platforms, the author decided to write *sec* in Perl. Since Perl runs on almost every operating system flavour and has become a standard part of many OS distributons [2], applications written in Perl are able to run on a wide range of operating systems. In addition, the support for regular expressions is integrated directly into Perl language core, and Perl programs are almost as fast as programs written in C.

Some event correlation engines provide the user with predefined correlation rule types that can be parameterized and used as building blocks for defining more complex event correlation operations [3, 4]. *Sec* takes a similar approach and supports the following rule types as building blocks (a number of event correlation guides like [3, 4, 5] were used to derive the rule types):

- **Single** – match input event and execute an action.
- **SingleWithScript** - match input event and execute an action, if an external script (e.g., query to a network topology database) returns success code for its exit value.
- **SingleWithSuppress** - match input event and execute an action, but ignore following matching events for the next *t* seconds.
- **Pair** - match input event, execute an action immediately, and ignore following matching events until some other input event arrives. On the arrival of the second event execute another action.
- **PairWithWindow** - match input event and wait for *t* seconds for other input event to arrive. If that event is not observed within the given time window, execute an action. If the event arrives on time, execute another action.
- **SingleWithThreshold** - count matching input events during *t* seconds and if a given threshold *n* is exceeded, execute an action.
- **SingleWith2Thresholds** - count matching input events during *t* seconds and if a given threshold *n* is exceeded, execute an action. The counting continues after the execution - when no more than *n'* events have been observed during last *t'* seconds, execute another action.
- **Suppress** - suppress matching input event.
- **Calendar** - execute an action at specific times.

Rules allow not only shell commands to be executed as actions, but it is also possible to create and delete *contexts* that can activate or deactivate rules, generate *input events* for other rules, and *reset* active event correlation operations (e.g., reset ongoing event counting). If necessary, Boolean expressions consisting of context names and Boolean operators (AND, OR, NOT) can be specified as a part of a rule definition. The truth value of the expression decides whether the rule can be applied at a given moment. By combining several rules with appropriate actions and Boolean context expressions, more complex event correlation schemes can be formed. As an example, the following lines present a simple rule base for correlating *interface down* and *interface up* events:

1. Every day at 11 PM, create the context NIGHT with the lifetime of 8 hours.
2. If event *"<router> <interface> down"* has been observed and no *interface up* event will be received for the same interface within 15 seconds, generate event *"<router> <interface> OUTAGE"*, otherwise generate event *"<router> <interface> BOUNCE"*.
3. If event *"<router> <interface> OUTAGE"* has been observed, report it; then wait for event *"<router> <interface> up"* and report it when it occurs.
4. Count events *"<router> <interface> BOUNCE"* and if more than 5 events for the same interface are observed within 60 minutes, report event *"<router> <interface> UNSTABLE"*. This rule can only be applied when context NIGHT does not exist (i.e., NOT(NIGHT) is true).

3. Experience, availability, and future work

Sec 1.0 was released on March 23, 2001. By now, the tool has been adopted by several companies all around the world. A year's experience with *sec* has proven it to be an excellent tool for central and local event correlation, performing well under high event processing loads. The author has received reports of successful use of *sec* on Solaris, HP-UX, Tru64 UNIX, Linux, and Windows2000 platforms. *Sec* has been successfully integrated with HP OpenView Network Node Manager, HP OpenView ITO management server, and ITO agents.

Sec is distributed under the terms of GNU General Public License, and can be downloaded from *http://kodu.neti.ee/~risto/sec/*.

For a future work, the author plans to use *sec* in other research areas which are similar to network management and which could benefit from event correlation techniques (e.g., intrusion detection).

References

[1] Shane O'Donnell, "Network Management: Open Source Solutions to Proprietary Problems", in *Proceedings of the 28th SIGUCCS Conference on User Services*, Richmond, VA, 2000, pp. 208-217.

[2] Peter Wainwright et al., "Professional Perl Programming", Birmingham, UK, Wrox Press Ltd., 2001.

[3] Event Correlation Services – Designer's Guide, HP document J1095-90304, Hewlett-Packard Company, 1998.

[4] Integrated Network Management Solutions Using NetView Version 5.1, Tivoli Redbook SG24-5285-00, IBM Corp., 1999.

[5] Cisco Network Monitoring and Event Correlation Guidelines, Reference Guide, Cisco Systems Inc., 1999.