Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Log monitoring and analysis with rsyslog and Splunk

A. Messina, I. Fontana, G. Giacalone

Rapporto Tecnico N.:
RT-ICAR-PA-15-07

Dicembre 2015

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) –
Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sede di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sede di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Log monitoring and analysis with rsyslog and Splunk

A. Messina[1], I. Fontana[2], G. Giacalone[2]

Rapporto Tecnico N.:
RT-ICAR-PA-15-07 **Dicembre 2015**

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Palermo, Viale delle Scienze edificio 11, 90128 Palermo.
[2] Istituto per l'Ambiente Marino Costiero, IAMC-CNR, Sede di Capo Granitola, Via del Mare n. 3, 90121 Torretta Granitola – Campobello di Mazara.

Index

1 Introduction

Each computer system, regardless of the Operating System (OS), has a mechanism that records (almost) all the performed activities. Such information, commonly called *log* in information technology, is not normally related to the average end user, but it plays a fundamental role in the life of the system administrator when problems arise, because the included occurred errors are the starting point of the resolution process.

Therefore, one of the key aspect in the management of an IT infrastructure is the logging and the conservation, as in the logbook of a ship or in a book where to write down daily activities, of everything happening in the system (system logs, application logs, database logs, security logs, and so on), that is information about the health status and performance of the system and the running applications.

From the point of view of the integrity of the acquired data and its careful use, keeping the log just on devices generating them is an important limitation for several reasons:

- their recover becomes impossible when a hardware failure occurs on the media on which these files are stored;
- in the case of an intrusion, the intruder first deactivates the log service and erases the existing log, making it impossible to reconstruct what happened;
- in the case of continuous monitoring of numerous devices, maintaining a detailed control it is very complicated or even impossible, because:
    i. you have to define policies for managing logs of each system;
    ii. you have to access physically to every system and this also implicates an increased difficulty in analysis, reporting and archiving;
    iii. you have to re-configure all systems for every change of policy.

To try to overcome the problems mentioned above, you should centralize the logs on a dedicated host, which with appropriate routines allows the conservation and an easy analysis.

A centralized *log server* is a log monitoring point on a network to which all type of devices, including Windows or Linux servers, routers, switches or any other device, can send their logs and can filter the logs in order to see the information more easily.

A *log analyser* is a software solution to view, monitor and analyse the events recorded in the security, system and application logs.

One of the main characteristics differentiating the Unix or Unix-like systems (Linux) from others, is certainly their native capacity to provide information in the form of log, relating to the status of system services, kernel, device discovery, access to system and any system event.

In this work, a centralized *log server* has been implemented, that is a host with the task of collecting the logs of all other hosts on the network, to allow both to have a certainly valid copy of the log and to improve and simplify considerably the log management of different machines.

In this structure, all the devices under monitoring can also send their logs to the remote *log server*. The logs can be transmitted in the clear or in an encrypted manner, to prevent them from being intercepted and/or altered.

Generally, the protocol used is the User Datagram Protocol (UDP) through port 514, for special applications where monitoring is essential; otherwise, where certain events can or should trigger actions by the *log server*, Transmission Control Protocol (TCP) implementations are used.

In the *log server*, the daemon log must necessarily run as it lets the host listen to the messages from the network; obviously, in the messages received the host name where the message was generated will appear.

After having centralized all logs, analysing the collected data can be useful to obtain comparable statistics over the years (weekly, monthly, quarterly) according to the monitoring requirements.

To get such information, these two ways can be chosen:

i. analyse the log files by simply reading them sequentially;
ii. use the help of complex structures organizing the log file contents so that they can be processed to obtain structured information.

There is a multitude of platforms analysing the logs and producing output, typically in HyperText Markup Language (HTML), containing tables, diagrams, graphs and statistics in order to summarize all the information.

2 The System Log protocol

2.1 Introduction

The protocol System Log (SysLog), defined in [1][2], was originally written by Eric Allman in 1981, and it was the standard up to a short time ago.

Syslog is the service used by demons running in background and by the kernel when they need to record the system messages. The service is generally called *syslog* and it is managed by the program *syslogd*, which is responsible for collecting all these messages and saving them or send them in an appropriate manner. This daemon is an integral part of many Unix/Linux distributions and it does not need to be downloaded or installed.

2.2 The protocol

Syslog uses the UDP protocol and the port 514 for communication, and, being a connectionless protocol, it does not provide any acknowledgments. Therefore, in practice, even the syslog server does not send acknowledgments back to the sender when it receives log messages; as a result, the device generates and sends log messages without knowing if the syslog server has received its messages.

The size of the syslog package (see Figure 1) is limited to 1024 bytes and it carries the following information: Facility, Severity, Hostname, Timestamp, Message.

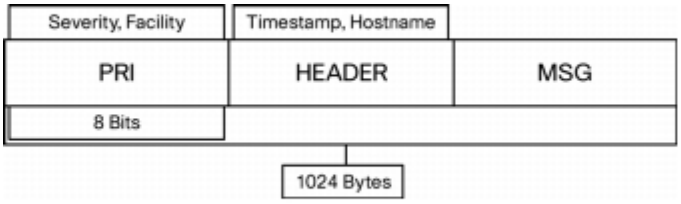
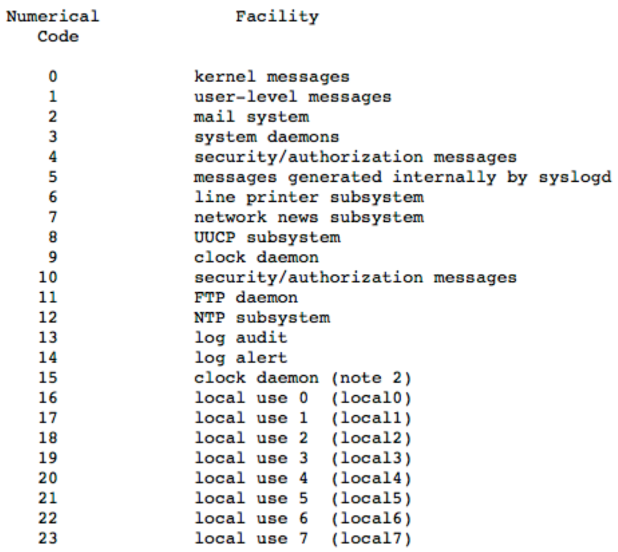


Figure 1 - Syslog packet structure

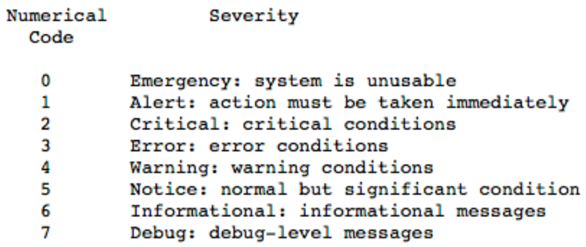
The block PRI (Priority) is a sequence of 8 bits and it contains the "Facility" and "Severity" of the message. The three least significant bits represent the severity of the message, while the other five bits identify the facility.

- **Facility**. Syslog messages are widely classified on the basis of the sources generating them. Such sources can be the operating system, the processor or an application. These categories, called facility, are represented by integers, as shown in the following table:



Numerical Code	Facility
0	kernel messages
1	user-level messages
2	mail system
3	system daemons
4	security/authorization messages
5	messages generated internally by syslogd
6	line printer subsystem
7	network news subsystem
8	UUCP subsystem
9	clock daemon
10	security/authorization messages
11	FTP daemon
12	NTP subsystem
13	log audit
14	log alert
15	clock daemon (note 2)
16	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

- **Severity**. The source or facility generating the syslog message, also specifies the severity of the message using a whole single digit number, as shown below:



Numerical Code	Severity
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

The HEADER contains the “Timestamp” and “Hostname” fields.

- **Timestamp**. The *timestamp* field is used for the local time, in the format MMMM DD HH:MM:SS, of the device when the message has been generated. If the type asterisk (*) or dot (.) precede a syslog message, this indicates problems with the Network Time Protocol (NTP) [3]. The type "*" indicates that the time is not self-synchronized or it has never been set. The type"." means that the time is in auto-sync but it does not reach any configured NTP server.
- **Hostname**. The *hostname* field is the host name (as configured on the host itself) or IP address. In devices such as routers or firewalls using multiple interfaces, syslog uses the interface IP address from which the message is transmitted

The MSG is the text of the syslog message, along with some additional information on the process that generated the message.

Recently, the traditional syslog Unix, that only relies on the UDP protocol, has been replaced by the package *rsyslog,* which includes support for the TCP protocol and many other features.

3 rsyslog

3.1 Introduction

Rsyslog [4] is an open-source software utility used on UNIX and Unix-like computer systems for forwarding log messages in an IP network. It implements the basic syslog protocol, extends it with content-based filtering, rich filtering capabilities, flexible configuration options and adds features such as using TCP for transport.

The official website defines the utility as "the rocket-fast system for log processing".

3.2 The extended rsyslog protocol

Rsyslog uses the standard BSD syslog protocol, as specified in RFC 3164. Because RFC 3164 is just an informational description and not a real standard, some various incompatible extensions emerged and Rsyslog supports many of these extensions.

The most important extensions of the original protocol supported by rsyslog are:

- ISO 8601 [5] timestamp with millisecond granularity and timezone information
- the addition of the name of relays in the host fields to make it possible to track the path a given message has traversed
- reliable transport using TCP
- support GSS-API [6] and TLS [7]
- logging directly into various database engines.
- support for RFC 5424, RFC 5425, RFC 5426
- support for RELP [8]
- support for buffered operation modes where messages are buffered locally if the receiver is not ready
- complete input/output support for systemd journal

On the client side, *rsyslogd* is responsible for collecting service messages coming from applications and the kernel, and distributing them in local log files or to the remote log server via TCP protocol.

On the server side, *rsyslogd* saves the information sent by the various network devices on a special file created when required, so as to keep track of the most important events.

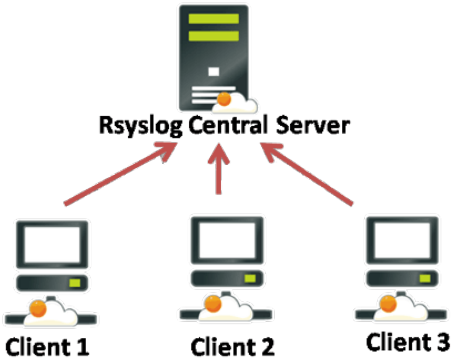


Figure 2 - Centralized Logging System using Rsyslog

In both cases, rsyslogd obeys the */etc/rsyslog.conf* configuration file and supports the same syntax used to syslogd.

One of the advantages of rsyslog is that the tool has been developed in a modular way, and it is possible to activate additional features, such as managing directly the kernel logs, or providing the syslog service by enabling the relevant modules.

Apart from writing logs on files, rsyslog allows you to register in different platforms (MySQL, PostgreSQL) and modes. Preceding the "|" type to the name of a file indicates that you are referring to a file from which another programme can read the data. It is possible to send the output specifying as a file the one of a terminal device (es. */dev/tty10*). It is possible to send messages to lists of users, identified by usernames separated by commas, or by using the "*" type the messages are sent to anyone is connected to the system.

Another interesting feature of syslog is that it is possible to send all messages to one or more remote machines. This can be done using the "@" type followed by the name of the destination machine, that is, by specifying the hostname or its network address; on the contrary, if that machine has a syslog enabled to listen via network, it will receive all the messages.

3.3 Rsyslog vs Syslog-NG

For the log server implementation, it was considered useful to make an analysis of the major open-source centralized logging systems. In particular, a comparison was made between Rsyslog and Syslog-NG [9].

In this work, Rsyslog was preferred to the most popular Syslog-NG for the following reasons:

- **License and software features.** Syslog-NG is dual-licensed. A commercial product has been forked from the General Public License (GPL) open-source project, and the most advanced features are found only in the commercial offering. Listed below some interesting features:
- **Truly reliable message delivery (RELP).** Rsyslog addresses the unreliability of the TCP protocol through the development of the RELP protocol.
- **Compliance with IETF regarding reliable TCP transport (RFC 3195).** Rsyslog is compliant with the standards regarding reliable TCP transport [10].
- **Native support for traffic encryption (TLS/SSL).** Rsyslog natively supports TLS, whereas the GPL fork of Syslog-NG does not.
- **SNMP support.** Rsyslog supports SNMP traps [11].
- **Hostname and program name blocks.** Rsyslog supports powerful BSD-style hostname and program name blocks for easy multi-host implementations.
- **On-disk message spooling.** Rsyslog has on-disk file spooling features that are lacking in GPL Syslog-NG.
- **Organized configuration files**. Rsyslog has configuration include file support that Syslog-NG lacks. This allows to split the configuration file into multiple files.
- **Native support for email alerts**. Rsyslog natively supports the ability to send email alerts based on log message content. Syslog-NG needs to pipe data to an external process.

4 Splunk

4.1 Introduction

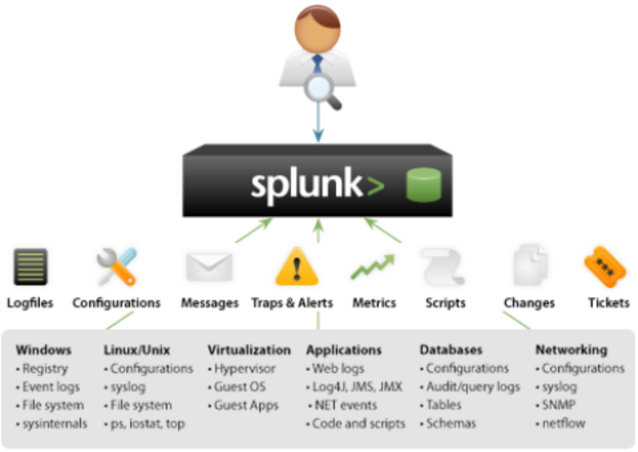
The networks and complex IT infrastructures generate massive streams of machine data, vital information such as security risks, consumption capacity, service levels, fraudulent activities and much more, in a series of unpredictable formats difficult to process and to analyse with conventional methods and in real time.

The availability of software or framework able to perform *Log Analysis* is wide: Scribe, Flume, LogStash, Splunk, Graylog2, Ossec, and so on.

In the case of software for *Log Analysis*, research can only fall back on portable, scalable and dynamic application, which performs the tasks assigned in real time and has the ability to alert administrators through systems such as mail or graphical interfaces.

The term *dynamic* refers to the ability of a software to perform different tasks through simple configurations; *portable* is the ability of a software to adapt to different operating systems; finally, the term *scalable* refers to a software capable of operating effectively on different network architectures, reducing bottlenecks that might form and extending or reducing any probable resources available depending on the workload.

Among the platforms listed above, Splunk seems the most used, as it supports extensive resources for real-time log analysis, it interprets data when trying to provide a more complete context, and it was created not by chance for an effective management of massive amounts of big data generated by machines. The software provides a unified way to organize and extract useful information from massive amounts of machine data generated by different sources.



Splunk is an enterprise-level monitoring system, made simple, collecting and indexing machine data in real time generated by technological infrastructure and all IT systems: physical, virtual and in the "cloud". The licenses are not cheap, but it has a proprietary license in limited testing at an amount of log equal to 500MB per day. This choice meets the prerogatives of this work and is, momentarily, enough for our needs.

4.2 Platform features

Some characteristics of Splunk are listed below:

- Download and installation in a few minutes. Immediately after installation, access to a user-friendly web interface for users and a powerful engine to index machine data.
- Possibility to use as input the syslog or syslog-ng and other input sources using TPC/UDP protocol, now present in most of the operating systems on the market.
- The application is developed in python, based on a search engine allowing to select, aggregate, and display data in highly customizable reports. It allows analysing historical and real-time data and then quickly viewing and sharing all data, regardless of their structure, diversity and size.
- With Splunk, customized applications (App) can be created, to offer a "tailored" user experience, according to the different roles and use cases. The ability to create, use, and share the apps within the structure of use or make them available within the Splunk community. In the site of the community (http://www.splunkbase.com), a considerable and growing amount of apps can be found, created by users, partners and Splunk. There are Apps helping to visualize data geographically or offering views of conformity preconfigured, or App for different technologies, such as Windows, Linux, Unix, Virtualization, Networking and much more.
- Searching for real-time alerts to automatically trigger actions such as the automated sending of e-mail, the execution of the script action or publication of RSS feeds. The alarms can also be used to send SNMP traps to the system management console or to generate tickets to be sent to the service desk. It is possible to set alarms at different levels of granularity basing them on a wide variety of thresholds.

5 Implementation

5.1 Introduction

We have implemented a centralized log monitoring system with rsyslog and Splunk within the main IT infrastructure of the Sicilian Biodiversity Observatory Project (ORBS) [13], wherein we can found eight virtualization nodes, two storage server and one firewall (see Figure 3).

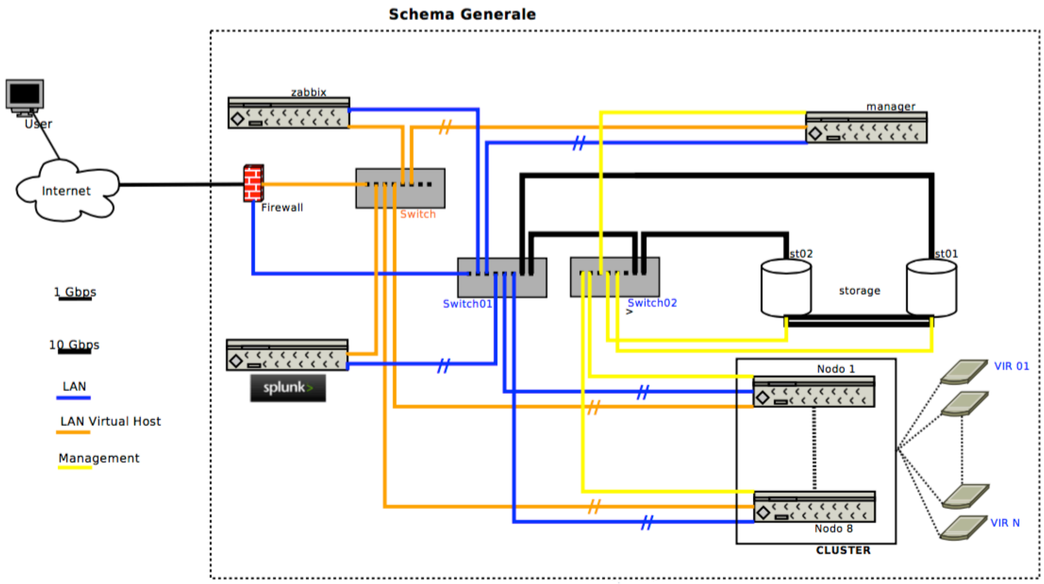


Figure 3 - The main IT infrastructure of ORBS

5.2 Rsyslog server configuration

We first set up a dedicated linux server for logs collection and Splunk's deploy. The minimal installation of the chosen linux distribution, CentOS v6, already includes the rsyslog package, therefore the system was simply installed as is.

The configuration of rsyslog was made modifying the file */etc/rsyslog.conf* as follow:

- Modules loading and activation of UDP, TCP and RELP protocols:

```
$ModLoad imudp.so
$UDPServerRun 514
$ModLoad imtcp.so
$InputTCPServerRun 10514
$ModLoad imrelp
$InputRELPServerRun 20514
```

- In order to give a well-defined organization to the logs coming from the various clients, the log files were organized within a tree structure, by using the *template* facility. Starting from the path */var/log/LogServer*, the logs are organized by source *host*, then by *date* splitted in *year*, *month* and *day*, and finally by the *application* type:

```
$template
Messages,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/%programname
%.log"

$template
Secure,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/secure"

$template
Maillog,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/maillog"

$template Cron,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/cron"

$template
Spooler,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/spooler"

$template
Boot,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/boot.log"

$template
Kern,"/var/log/LogServer/%hostname%/%$YEAR%/%$MONTH%/%$DAY%/kern.log"
```

- We also set up other parameters, including the type of format used to write the logs, which in this case remains the default, and the files and directories creation permissions:

```
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat

$FileCreateMode 0640
$DirCreateMode 0755
$Umask 0022
```

- Finally, we have specified the rules by which the logs are directed to the various templates based on their facility and severity:

```
#### RULES ####
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none ?Messages
# The authpriv file has restricted access.
authpriv.* ?Secure
# Log all the mail messages in one place.
mail.* -?Maillog
# Log cron stuff
cron.* ?Cron
# Everybody gets emergency messages
*.emerg *
# Save news errors of level crit and higher in a special file.
uucp,news.crit ?Spooler
# Save boot messages also to boot.log
local7.* ?Boot
# Log all kernel messages
kern.* ?Kern
```

5.3 Rsyslog client configuration

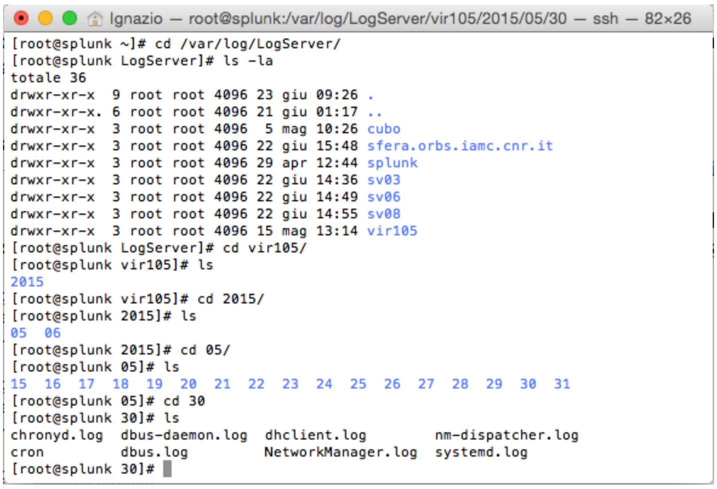
The eight virtualization nodes, the various hosted virtual machines, the two storages and the firewall represent the clients in our centralized log system.

The client-side configuration of the log system file is reported below:

```
#### RULES ####
kern.*                                              /var/log/kern.log
*.info;mail.none;authpriv.none;cron.none            /var/log/messages
authpriv.*                                          /var/log/secure
mail.*                                              /var/log/maillog
cron.*                                              /var/log/cron
*.emerg                                             :omusrmsg:*
uucp,news.crit                                      /var/log/spooler
local7.*                                            /var/log/boot.log

# ### begin forwarding rule ###
*.* @@ServerRsyslog:10514
*.* :omrelp: ServerRsyslog:20514
# ### end of the forwarding rule ##
```

Of course, even a Windows client would be able to send its logs to the rsyslog server thanks to some additional software tool, such as EventReporter, Eventlog to Syslog, Event Log Forwarder, Snare Agent, or others.

Figure 4 shows a terminal session on the log server. Here, we can see the log directory tree where the logs are saved, organized by host and date.



Ignazio — root@splunk:/var/log/LogServer/vir105/2015/05/30 — ssh — 82×26

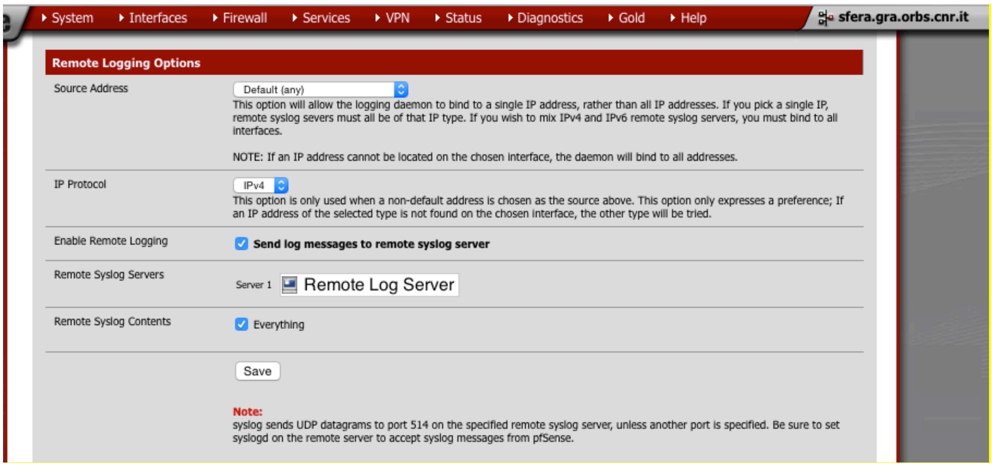
```
[root@splunk ~]# cd /var/log/LogServer/
[root@splunk LogServer]# ls -la
totale 36
drwxr-xr-x  9 root root 4096 23 giu 09:26 .
drwxr-xr-x. 6 root root 4096 21 giu 01:17 ..
drwxr-xr-x  3 root root 4096  5 mag 10:26 cubo
drwxr-xr-x  3 root root 4096 22 giu 15:48 sfera.orbs.iamc.cnr.it
drwxr-xr-x  3 root root 4096 29 apr 12:44 splunk
drwxr-xr-x  3 root root 4096 22 giu 14:36 sv03
drwxr-xr-x  3 root root 4096 22 giu 14:49 sv06
drwxr-xr-x  3 root root 4096 22 giu 14:55 sv08
drwxr-xr-x  3 root root 4096 15 mag 13:14 vir105
[root@splunk LogServer]# cd vir105/
[root@splunk vir105]# ls
2015
[root@splunk vir105]# cd 2015/
[root@splunk 2015]# ls
05  06
[root@splunk 2015]# cd 05/
[root@splunk 05]# ls
15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31
[root@splunk 05]# cd 30
[root@splunk 30]# ls
chronyd.log  dbus-daemon.log  dhclient.log        nm-dispatcher.log
cron         dbus.log         NetworkManager.log  systemd.log
[root@splunk 30]# 
```

Figure 4 – Log files directory structure

5.3.1 The firewall’s logs

The firewall, based on pfSense [14], natively supports remote syslog via UDP protocol on port 514.

For its activation, we simply set the IP address of the remote syslog server, as shown in Figure 5.



Remote Logging Options

Source Address: Default (any)

This option will allow the logging daemon to bind to a single IP address, rather than all IP addresses. If you pick a single IP, remote syslog severs must all be of that IP type. If you wish to mix IPv4 and IPv6 remote syslog servers, you must bind to all interfaces.

NOTE: If an IP address cannot be located on the chosen interface, the daemon will bind to all addresses.

IP Protocol: IPv4

This option is only used when a non-default address is chosen as the source above. This option only expresses a preference; If an IP address of the selected type is not found on the chosen interface, the other type will be tried.

Enable Remote Logging: ☑ **Send log messages to remote syslog server**

Remote Syslog Servers: Server 1 Remote Log Server

Remote Syslog Contents: ☑ Everything

Save

Note:
syslog sends UDP datagrams to port 514 on the specified remote syslog server, unless another port is specified. Be sure to set syslogd on the remote server to accept syslog messages from pfSense.

Figure 5 - Remote logging activation on the firewall

5.4 Splunk

The *Splunk-6.2.2-255606.i386.rpm* package was downloaded from the official site (http://www.splunk.com) and then installed.

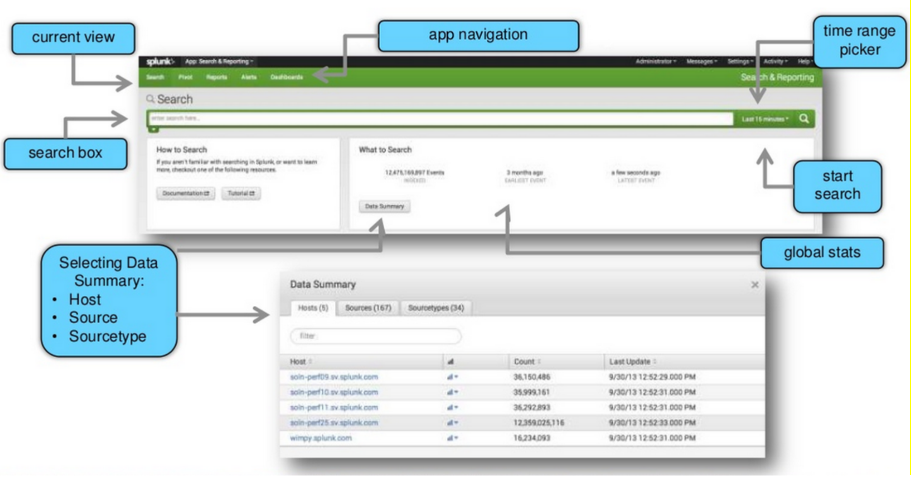
Appropriate entries were added to the log server's firewall configuration. In our case, as it is a CentOS distribution, the */etc/sysconfig/iptables* configuration file was modified:

```
# Rules for https (Splunk website)
-A INPUT -i eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -o eth0 -p tcp --sport 443 -m state --state NEW,ESTABLISHED -j ACCEPT

# Rules for Splunk
-A INPUT -i eth0 -p tcp --dport 8000 -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -o eth0 -p tcp --sport 8000 -m state --state NEW,ESTABLISHED -j ACCEPT

# Rules for RELP
-A INPUT -i eth1 -p tcp --dport 20514 -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -o eth1 -p tcp --sport 20514 -m state --state NEW,ESTABLISHED -j ACCEPT

# Rules for TCP logging
-A INPUT -i eth1 -p tcp --dport 10514 -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -o eth1 -p tcp --sport 10514 -m state --state NEW,ESTABLISHED -j ACCEPT

# Rules for UDP logging
-A INPUT -i eth1 -p udp --dport 514 -m state --state NEW,ESTABLISHED -j ACCEPT
-A OUTPUT -o eth1 -p udp --sport 514 -m state --state NEW,ESTABLISHED -j ACCEPT
```

After that, the *splunk* service was started with the command */opt/splunk/bin/splunk start*.

To start using the application, a browser pointing to the URL of the web interface of

Splunk is running is needed. Typically, such URL is *http://<servername>:8000*, but it can vary. After the sign-in, the typical initial screen is showed, as reported in Figure 6:



Data Summary

Hosts (5) | Sources (167) | Sourcetypes (34)

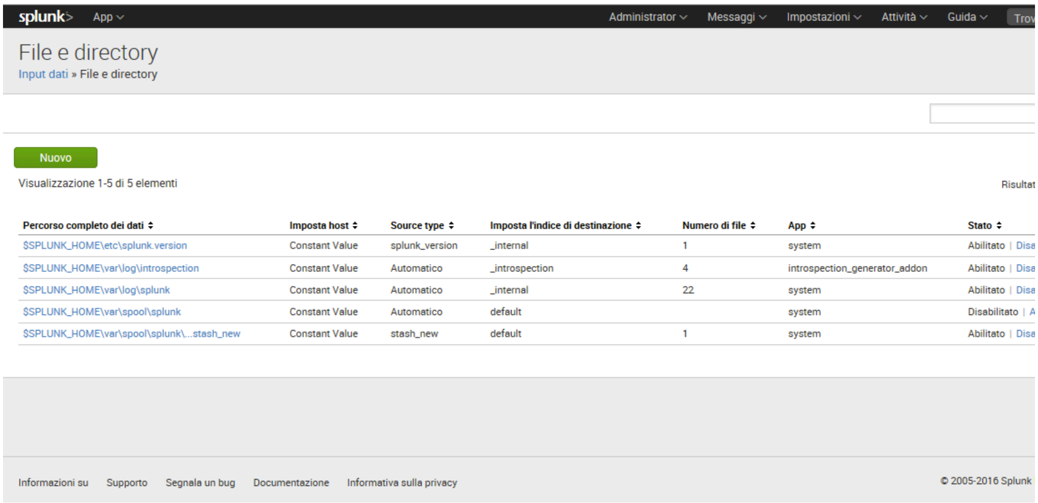
Host		Count	Last Update
soin-perf09.sv.splunk.com		36,150,486	9/30/13 12:52:29.000 PM
soin-perf10.sv.splunk.com		35,999,161	9/30/13 12:52:31.000 PM
soin-perf11.sv.splunk.com		36,292,893	9/30/13 12:52:31.000 PM
soin-perf25.sv.splunk.com		12,359,025,116	9/30/13 12:52:33.000 PM
wimpy.splunk.com		16,234,093	9/30/13 12:52:31.000 PM

Figure 6 - Home screen of Splunk (source: http://www.learnsplunk.com/splunk-search-tutorial.html)

The page is composed of the following sections:

- *Search box*: used to enter search keywords, e.g. username/error/event code for which logs are needed.
- *Time range picker*: used to select the time range needed to the search. Shorter the time range is, faster the searching will be.
- *Data summary*: it shows statistics of the searched logs, i.e. the number of results found after searching.
- *Global stats*: it shows the number of logs indexed by Splunk.

The first step is the configuration of the log. Starting from the menu Splunk → Settings → Input Data → File and Directory, the full directory path containing all the logs of various clients can be added by clicking the button New (Figure 4.5).



File e directory

Input dati » File e directory

Visualizzazione 1-5 di 5 elementi

Percorso completo dei dati	Imposta host	Source type	Imposta l'indice di destinazione	Numero di file	App	Stato
$SPLUNK_HOME\etc\splunk.version	Constant Value	splunk_version	_internal	1	system	Abilitato | Disa
$SPLUNK_HOME\var\log\introspection	Constant Value	Automatico	_introspection	4	introspection_generator_addon	Abilitato | Disa
$SPLUNK_HOME\var\log\splunk	Constant Value	Automatico	_internal	22	system	Abilitato | Disa
$SPLUNK_HOME\var\spool\splunk	Constant Value	Automatico	default		system	Disabilitato | A
$SPLUNK_HOME\var\spool\splunk\...stash_new	Constant Value	stash_new	default	1	system	Abilitato | Disa

Figure 7 - Files and directory configuration

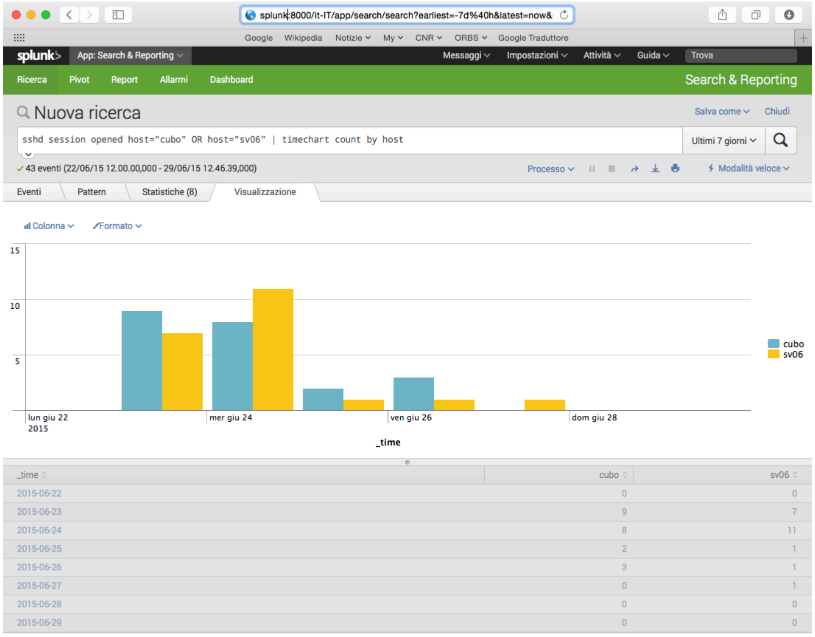
After that, we are able to search for terms and specific expressions, and we can also ise Boolean operators as well, to refine our search or to trace transactions across multiple systems.

The reporting and statistical controls allow to control the volumes of transaction data, to calculate metrics and to search for specific conditions within a variable period of time. The *Search Assistant* function offers suggestions during the typing, and a context-sensitive help, in order to use the potential of the Splunk search language to the utmost.

To highlight the ease of use and the power of Splunk, in order to know, for example, how many *ssh* sessions were successfully done in the last 7 days to two specific linux hosts in the infrastructure, we just need to go to the *Search & Reporting* section and type the following query:

```
sshd session opened host="cubo" OR host="sv06" | timechart count by host
```

This may help to give an overall view of any problems, such as an excessive number of failed login attempts. Figure 8 shows the results of the previous query.



```
sshd session opened host="cubo" OR host="sv06" | timechart count by host
```

✓ 43 eventi (22/06/15 12.00.00,000 - 29/06/15 12.46.39,000)

_time	cubo	sv06
2015-06-22	0	0
2015-06-23	9	7
2015-06-24	8	11
2015-06-25	2	1
2015-06-26	3	1
2015-06-27	0	1
2015-06-28	0	0
2015-06-29	0	0

Figure 8 - Sample query output

A variety of search commands can be used to extract information in different ways:

- *rex*: to extract a field using regular expressions in Perl;
- *extract*: to explicitly extract fields/values using predefined patterns;
- *multikv*: to extract fields/values of events on formatted multi-lines or tables;
- *spath*: to extract fields/values of events in XML format and Json;
- *kvform*: to extract fields/values based on predefined templates.

A such obtained graphical result can be saved as report. Finally, the reports can be loaded into one or more Dashboards, to have an immediate overview of all the aspects to monitor.

Specifically, we created a dashboard named “*Monitoraggio Infrastruttura*" (see Figure 9). It was designed to monitor the access to the hosts, to highlight the recent *ssh* accesses (in the last 7 days) performed on individual hosts and the failed authentications, distinguishing them by host destination and source IP address. Another panel highlights the number of access to the infrastructure from the virtual private network, distinguishing them by source IP address.

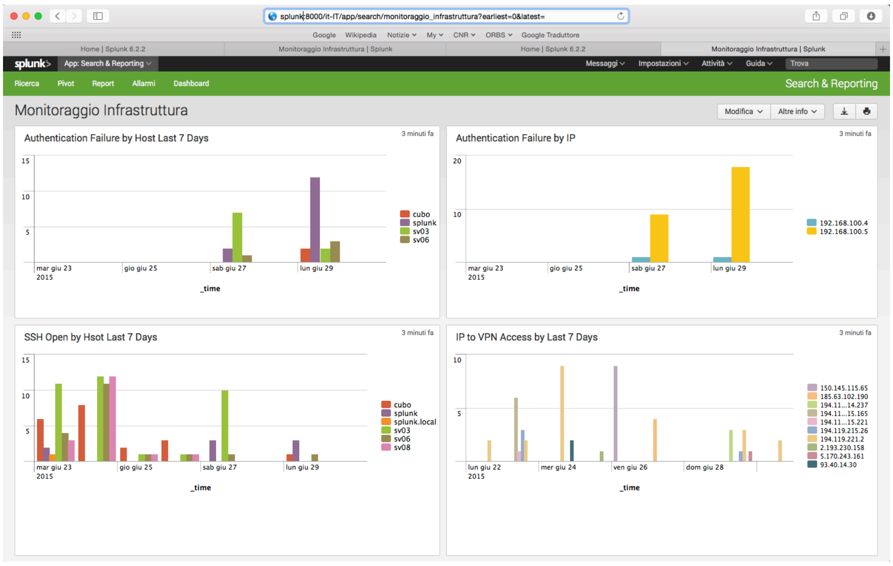


Figure 9 - The dashboard "Monitoraggio Infrastruttura"

After the installation of the free apps *Splunk for use with AMMAP* and *geoip*, we have created another query, in order to extract from firewall logs all the IP addresses of the hosts connecting to the provided public services. As shown in Figure 10, we obtain the map of IP spatial dislocation in the world.

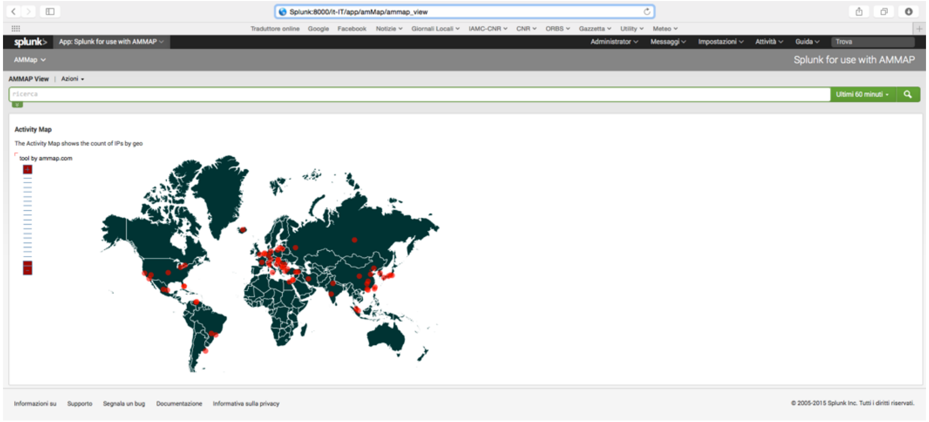


Figure 10 - Location of public services clients

6 References

[1] The Internet Society, “The BSD syslog Protocol – RFC 3164”, Aug. 2001, URL: https://www.ietf.org/rfc/rfc3164.txt

[2] The Internet Engineering Task Force (IETF), “The Syslog Protocol – RFC 5424”, Mar. 2009, URL: https://www.ietf.org/rfc/rfc5424.txt

[3] Network Time Foundation, “Network Time Protocol”, URL: http://www.ntp.org

[4] Adiscon GmbH, “RSYSLOG”, URL: http://www.rsyslog.com

[5] “Date and time format – ISO 8601”, URL: https://en.wikipedia.org/wiki/ISO_8601

[6] The Internet Society, “Generic Security Service Application Program Interface”, Jan. 2000, URL: https://tools.ietf.org/html/rfc2743

[7] The Internet Engineering Task Force (IETF), “The Transport Layer Security (TLS) Protocol”, Aug. 2008, URL: https://tools.ietf.org/html/rfc5246

[8] R. Gerhards, “RELP - The Reliable Event Logging Protocol”, Mar. 2008, URL: http://www.rsyslog.com/doc/relp.html

[9] BalaBit IT Security, “Syslog-NG”, URL: https://www.balabit.com/network-security/syslog-ng

[10] The Internet Society, “Reliable Delivery for syslog”, Nov. 2001, URL: https://www.ietf.org/rfc/rfc3195.txt

[11] J. Case, M. Fedor, M. Schoffstall, J. Davin, “A Simple Network Management Protocol (SNMP)”, May 1990, URL: https://tools.ietf.org/rfc/rfc1157.txt

[12] Splunk Inc., “Splunk”, URL: http://www.splunk.com

[13] Istituto Ambiente Marino Costiero, “Osservatorio Biodiversità Regione Sicilia”, URL: http://www.osservatoriobiodiversita.regione.sicilia.it/?page_id=733

[14] Electric Sheep Fencing LLC., “pfSense”, URL: https://www.pfsense.org