An overview of the open source log management tool - Graylog

RAHUL SINGH[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: rahpsing@iu.edu*

March 27, 2017

Graylog is an open source log management tool that allows an organization to collect, organize and analyze large amounts of data from its network activity. It enhances the basic log management functionality by providing network traffic analysis, lucene syntax based search, drill-down analysis of data using field statistics and generates trigger actions based alert notifications. It integrates with other open source technologies to address a larger distributed system.

Keywords: Cloud, I524

https://github.com/rahpsing/sp17-i524/paper2/S17-IR-2036/report.pdf

INTRODUCTION

Graylog allows management of an organization's computing resources in a consistent way. It allows us to centrally collect and manage log messages of an organization's complete infrastructure. A user can perform search on terrabytes of log data to discover number of failed logins,find application errors across all servers or monitor the acivity of a suspicious user id. Graylog works on top of ElasticSearch and MongoDB to facilitate this high availability searching. It provides a lucene like query language, a processing pipeline for data transformation, alerting abilities and much more. Graylog enables organizations, at a fraction of the cost, to improve IT operations efficiency, security, and reduce the cost of IT [1].

ARCHITECTURE

Graylog is written in Java and uses a few key open source technologies like Elasticsearch and MongoDB. Additionally, for a larger setup Apache Kafka or Rabbit MQ could be integrated to implement queueing. A basic Graylog cluster consists of the following components:

Graylog server - It is the actual log processor system also responsible for implementing security. The Graylog server nodes shall be operated on the fastest CPU's available.

Graylog web UI - It is the Graylog web user interface where one can view histograms, dashboards and create alerts.

MongoDB - MongoDB stores Graylog configuration information and the non queried log messages. It prime purpose in the architecture is Metadata Management [2].

ElasticSearch - ElasticSearch is useful for storing actual log data and perform search operations on them. Elasticsearch nodes should have as much RAM as possible and the fastest disks linked to them. Messages are only stored on Elasticsearch nodes. If we have data loss on Elasticsearch, all messages are gone – except if the administrator has created backups of the indices.

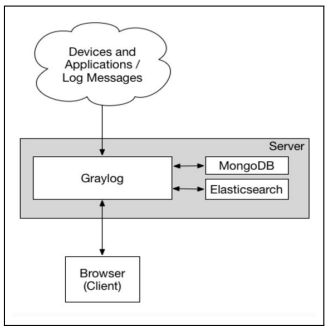

Fig. 1. [3] Graylog minimum architecture

GRAYLOG USE CASES

Computer And Network Security

The best way for intrusion detection is to monitor activity of all the devices in the network. Graylog allows a user to keep track of all failed logins, rejected network connections or exceptions

in the flow of the application. It also allows a user to integrate other Intrusion Detection Systems (IDS) to correlate detected activity with logs from all across the infrastructure [4].

Centralized IT management

Logging into every system and parsing the plain text log files to find meaningful data is an arduous task for any IT engineer. Graylog allows us to centrally collect all syslog and eventlog messages of the complete infrastructure thus allowing to solve production issues in real time. It does so by allowing the administrator to setup alerts for any trigger actions like performance degradation or exceptions.

Development and DevOps

Graylog allows on demand monitoring of distributed applications by giving tiered access to any developer in the organization to view system and application logs. It works on top of elastic search nodes to facilitate search operation on terrabytes of log data in a metter of milliseconds. In case of a customer operation resulting in an error, a developer shall need to search the logs with the customer id and locate the relevant logs to find the root cause of the problem.

GRAYLOG MODULES

Sending in log data

Graylog needs a log source to serve its purpose. The message input section is launched from the System -> Inputs section from the web interface (or the REST API) and can be configured without the need to restart any part of the system [5].

Graylog is able to accept and parse RFC 5424 and RFC 3164 compliant syslog messages and the Graylog Extended Log Format (GELF) [5]. For any devices on the network that do not publish RFC compliant syslog messages we need to make use of the plaintext messages. The Graylog Extended Log Format (GELF) is a log format that avoids the shortcomings of classic plain syslog by providing optional compression, fixed structure and a limits to payload length to 1024 bytes. Syslog is prefered for direct logging by machines in the network while GELF is suitable for logging from within the applications.

Search

As Graylog uses ElasticSearch to facilitate searching, the search syntax followed is very close to the Lucene syntax which is the underlying implementation of Elastic Search. By default, search is performed over all fields unless a specific field is specified in the search query. Search features like fuzzy, wildcard, range searches provided by Elastic Search API can be used to gain deeper insight to data. Graylog also provides a time frame selector which defines the time range over which the search shall be performed. The time selector could be relative, absolute or keyword based.<could extend this> Graylog also allows a user to save his searches to view it later. A user needs to save his search by a unique name and could load it later from the system under the saved search selector.

Graylog automatically constructs a histogram for the search results. The histogram depicts the concise number of messages received grouped by a certain time period that is adjustable. Based on a user's recent search queries,graylog also allows you to distinguish data that are not searched upon very often and thus can be archived on cost effective storage drives.

Log Streams

Graylog allows a user to create a set of rules to route messages into user defined categories. A user could create a stream called 'Database errors' and create a rule to direct all messages with the source attribute as 'database' to that stream. Thus, the stream 'Database errors' shall catch every error message from the system's database hosts. A message shall be routed into every stream that has the corresponding matching rule for the message. A message thus, can be part of many streams and not just one [6].

Alerts

Graylog alerts are periodical searches that can trigger some notifications when a defined condition is satisfied [7]. To get notified when more than 50 exceptions occur in the range of a minute, an alert can be created with the desired conditions. While defining alerts, a user can also specify the method of notification once the alert condition is met. Notifications can be obtained by an email or by an HTTP request to an endpoint in the system.

Dashboards

Graylog provides visualization through creation of dashboards that allows a user to build pre-defined views on his data to assemble all of his important data only a single click away [8]. Any search result or metric shall be added as a widget on the dashboard to observe trends in one single location. A user can also add search result metrics like result count, statistical values, field value charts and stack charts to the dashboard. These dashboards can also be shared with other users in the organization.

Full control and access through a REST API

Both configuration settings and log data are available through the Graylog REST API. The Graylog web interface uses Graylog Rest API internally to interact with the Graylog cluster. Graylog REST API could be used for automation or integrating Graylog into another system, such as monitoring or ticket systems [9]. Thus a network administrator can easily integrate Graylog into his evolving architecture and build reports and analysis.

Filtering messages

Graylog can use Drools to evaluate all incoming messages against a user defined rules file. To discard any message before its written to elastic search or to forward it to another system one can use Drools rules to perform custom filtering [10].

```
import org.graylog2.plugin.Message
import java.util.regex.Matcher
import java.util.regex.Pattern

rule "Blacklist all messages that start with 'firewall'"
  when
      m : Message( message matches "^firewall.*" )
  then
      System.out.println("DEBUG: Blacklisting message."); // Don't do this in production.
      m.setFilterOut(true);
end
```

Fig. 2. [10] Message filtering rule

COMPARISION WITH SPLUNK

Splunk is remarked as one of the best log management tools available and is Graylog's biggest competitor.

Table 1. Comparision of Graylog and Splunk

Parameter	Graylog	Splunk
Business Model	Opensource	Commercial software
Setup Time	Needs time	No time
Learning Curve	Difficult	Simple
Filetypes	syslog,gelf	Many
Security	Good	Good
Apps Supported	Low	Very high

CONCLUSION

Graylog provides an effective set of features to be adapted by any small to medium size organization. Alert notifications, sharing of dashboards and message filtering provide most features that any network administrator desires from a log management system. Being an open source tool it is cost effective compared to other log systems. However, it needs an environment to be setup before it can be operational. It has a steep learning curve with the responsibility of managing the MongoDB and ElasticSearch instances being completely managed by the user. Hence, the choice to choose Graylog as an organization's log management tool directly relies on the resources in terms of either time or money that it chooses to employ.

REFERENCES

[1] "Graylog," webpage, accessed : 03-20-2017. [Online]. Available: https://www.crunchbase.com/organization/graylog#/entity

[2] Severalnines, "High availibility log processing with graylog,mongodb and elastic search," webpage, Mar 2016, accessed : 03-19-2017. [Online]. Available: https://severalnines.com/blog/high-availability-log-processing-graylog-mongodb-and-elasticsearch

[3] "Architectural consideration - graylog 2.2.1 documentation," webpage, accessed : 03-19-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/architecture.html

[4] "Graylog | open source log management," webpage, accessed : 03-19-2017. [Online]. Available: https://www.graylog.org/

[5] "Sending in log data - graylog 2.2.1 documentation," webpage, accessed : 03-20-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/sending_data.html

[6] "Streams - graylog 2.2.1 documentation," webpage, accessed : 03-20-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/streams.html

[7] "Alerts - graylog 2.2.1 documentation," webpage, accessed : 03-20-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/streams/alerts.html

[8] "Dashboards - graylog 2.2.1 documentation," webpage, accessed : 03-20-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/dashboards.html

[9] "Graylog rest api - graylog 2.2.1 documentation," webpage, accessed : 03-21-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/configuration/rest_api.html

[10] "Blacklisting - graylog 2.2.1 documentation," webpage, accessed : 03-20-2017. [Online]. Available: http://docs.graylog.org/en/2.2/pages/blacklisting.html