Scalable Architectural Pattern for Integrating Syslog Servers with Splunk

Krishna Mohan Koyya

***Abstract:** An enterprise infrastructure consists of several devices. The devices emit event notifications representing their current state. The devices without storage such as printers and routers are configured to send the event notifications in the form of syslogs to one or more remote syslog servers over the network. Depending on the size and usage of the enterprise infrastructure, millions of syslogs may be emitted per second. These syslogs are used by the system administrators to detect and address the anomalies in the infrastructure. The system administrators often integrate the syslog servers with Log Analysis tools that offer aggregation, analytics, and visualisation capabilities. Splunk is one such popular tool that can be integrated with syslog servers. This paper proposes an architectural pattern for syslog servers that are to be integrated with Splunk for better performance, scalability and resilience.*

***Keywords:** Syslog, Syslog-ng, Splunk, Integration Patterns*

I. INTRODUCTION

Syslog was developed at University of Barkley, California, USA in the early 80s. Since then, it has been the standard logging solution on Unix-like systems. Diskless devices such as network routers and printers use syslog for logging the events on remote servers. The following are some of the circumstances in which devices send syslogs:

A printer emits a syslog when a print job is scheduled.

A printer emits a syslog when a print job is finished.

A VOIP-based phone emits a syslog when a call is placed.

A router emits a syslog when an interface fails to initialise.

The syslog protocol is documented in multiple RFCs.

A. RFC 3164

The IETF documented the RFC 3164 detailing the observed implementation of the syslog protocol across various devices. This is referred to as BSD Syslog Protocol [1].

Following is the summary of the RFC.

The devices send the syslogs using User Datagram Protocol (UDP) on port 514. The receiver does not send any acknowledgement upon receiving the syslogs.

Manuscript received on July 20, 2021.
Revised Manuscript received on July 26, 2021.
Manuscript published on 30 July, 2021.
* Correspondence Author

Krishna Mohan Koyya*, Assistant Professor, Department of Information Technology, Sasi Institute of Technology & Engineering, Tadepalligudem (Andhra Pradesh), India.

It is quite possible that some of the syslogs may be lost when the receiver is down. The receiver processes the collected syslogs, based on the application needs.

A receiver that forwards syslogs to another receiver is called a Relay whereas the receiver that processes the syslogs is called Syslog Server.

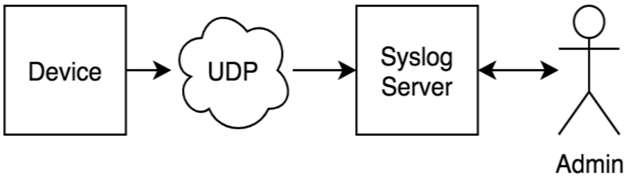

Fig 1: Connecting Devices and Syslog Server using BSD Syslog Protocol

The RFC also recommends a format for the syslogs.

The maximum length of the syslog should be limited to 1024 bytes. Every syslog should consist of three parts: HEADER, MSG and PRI. The HEADER should identify the hostname or IP Address of the device along with a timestamp. The MSG part should contain the actual text of the event. The PRI part should identify the computed priority of the syslog (it should be based on its severity and the facility of the device that generated the syslog).

Since RFC 3164 was not a standard but meant only for informational use, many vendors did not really implement the protocol as recommended in the RFC. Often, the formats of the syslogs from different vendors found to be incompatible with each other.

B. RFC 5424 and RFC 5426

Security has been one of the major concerns associated with BSD Syslog protocol. Since the syslogs carry critical state information of the devices, they attract man-in-the-middle attacks. For example, the protocol allows the attacker to intercept the syslogs on the wire between device and receiver, device and relay, relay and relay or relay and receiver. The relays and syslog servers are also vulnerable to DoS attacks.

Another issue associated with the BSD Syslog Protocol is that the format is not strictly enforced. Many devices emit the syslog events that are non-compliant with the suggested format.

The RFC 5424 addressed these two issues by proposing a 3-layered architecture. This protocol is often referred to as IETF Syslog Protocol [2].

The IETF Syslog Protocol recommends TLS (instead of UDP) to secure the infrastructure. TLS offers certificate based wire-level security.

*Retrieval Number: 100.1/ijrte.B63070710221
DOI: 10.35940/ijrte.B6307.0710221
Journal Website: www.ijrte.org*

*Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication
© Copyright: All rights reserved.*

Also, this protocol proposed a standard format for the syslogs with just two parts, namely HEADER and STRUCTURED DATA. As the name suggests, the STRUCTURED DATA part imposes a formal structure on the message.

However, since UDP was in wide-spread use, the RFC 5426 was released to propose best practices in using UDP in line with IETF Syslog Protocol [3].

C. Syslog-ng

The syslog-ng or Syslog Next Generation is a free and open source implementation of RFC 5424. It comes in two parts, namely client and server. The syslog-ng client runs on the devices whereas the syslog-ng server runs on a receiver. The client sends the syslogs to the receiver over a secure TLS connection. The syslog-ng also offers content-based filtering and flexible configuration options.

With the introduction of RFC 5424 along with a free implementation of it, it was expected that all the devices would move to the new protocol.

However, that has not been the case. Many devices still do not confirm with the standard protocol. Most of the syslog traffic is still on UDP and with a format that is not in conformance with the new RFCs.

Hence, the administrators of today are still dealing with the syslog systems that are built on top of plain UDP.
The syslog-ng supports UDP as well.

II. THE SYSLOG SERVER COMPONENTS

A Syslog Server consists of three components namely Collector, Parser and Processor as shown in the Fig 2.

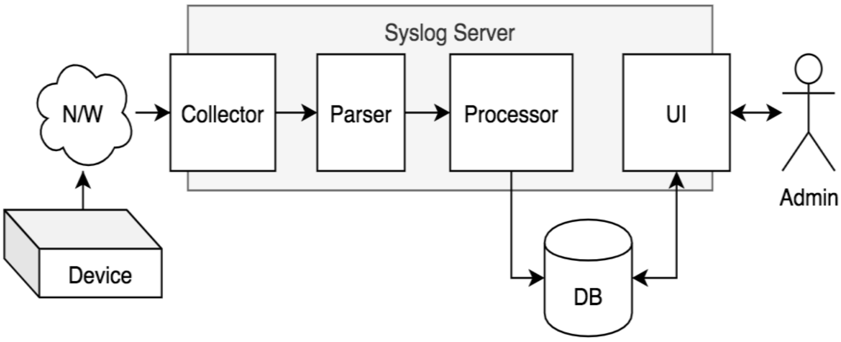

Fig 2: Syslog Server Components

The Collector component listens to the UDP/TCP/TLS port and collects the incoming syslogs in text format. The Parser component converts the received syslogs to the required data structures by identifying different fields such as severity, facility, timestamp, host name and other details. The Parser component also applies filter rules to weed out any irrelevant messages. Finally, the Processor component processes the syslogs. It usually includes persisting the syslogs and invoking any configured actions. Invoking a script and sending an email are some of the popular actions. Administrators define the filter rules and actions.

A simple syslog server may be designed to run all the three components in a single thread. However, since a lot of I/O is involved in processing the syslogs, it leads to poor performance. Because of the inherent nature of the UDP, a slow server may fail to collect all the syslogs.

A better alternative is to run each of the components in separate threads. In this design, the Collector keeps writing the received syslogs into a queue Q1 without waiting for further processing.

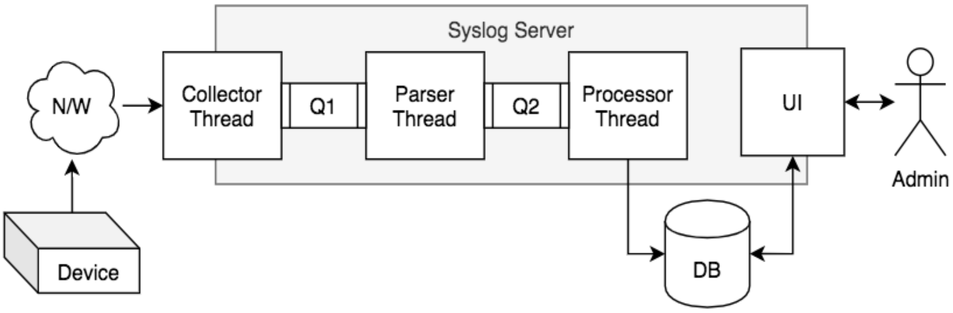

Fig 3: Multi-threaded Syslog Server

The Parser reads the raw messages from the Q1 and writes the parsed messages into another queue Q2.

Then the Processor reads the messages from the Q2 and processes them.

This multi-threaded design gives relatively better performance. It can be improved further by running multiple parsers and multiple processors in parallel, so that the system can scale well.

III. SPLUNK

Most of the contemporary infrastructure run a set of collaborative microservices on a distributed architecture.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication
© Copyright: All rights reserved.

It gives better performance, scalability, availability, modularity, agility and etc.,

However, a distributed system is very difficult to debug. It is because a single request involves multiple microservices on different nodes. In such a system, the logs from various nodes and microservices need to be collected at one single place in order to reconstruct the path of the request processing.

A. Splunk Enterprise Server

Splunk is popular a tool that aggregates logs from various sources [5]. It treats these logs as Machine Data. It indexes the machine data for efficient searching & reporting. It is very popular with IT Operation teams as it offers high quality dashboards with visualisations and analytics.

The data inputs that Splunk supports include files, directories, UDP, TCP, HTTP and scripts. Once configured, Splunk actively listens to these input sources for the machine data.

Splunk parses the machine data and discovers interesting data fields in the event as per the configured format. Administrator can configure different formats for different source types. The formats are normally supplied as regular expressions.

Splunk is also capable of invoking actions based on the received machine data.

B. Splunk Forwarder

It is a separate component of Splunk. As the name suggests, a forwarder collects machine data on a local machine and forwards them to a remote Splunk server over a secure connection. This is useful when the Splunk server cannot be hosted on a machine where the log files and directories are maintained.

There are two kinds of forwarders: Universal Forwarder and Heavy Forwarder. The Universal Forwarder is a light-weight forwarder that do not offer search capabilities whereas the Heavy Forwarder is a full-scale Splunk instance with search capabilities. Both the forwarders can pre-process the machine data in terms of tagging, compressing and etc.,

IV. INTEGRATING SYSLOG SERVER WITH SPLUNK

Many administrators prefer to integrate Syslog system with Splunk. It helps the administrator in reconstructing the scenarios with the help of syslogs and the events from other sources, from across the distributed system.

There are multiple ways in which Splunk and Syslog system can be integrated [4][6].

A. Splunk as the Syslog Server

Splunk supports UDP as a data input source. Thus, it is possible to use Splunk itself as the Syslog Server.

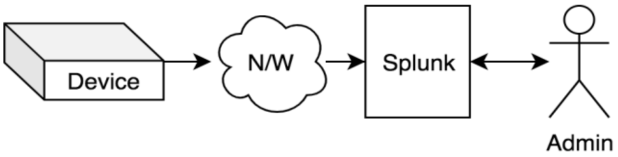

Fig 4: Splunk as the Syslog Server

In this architecture, the Splunk is installed on a server machine and configured to listen to the UDP port. The devices are configured to send the syslogs to the UDP port on the server.

This is the simplest setup. However, due to the inherent nature of UDP port on which syslogs are received, Splunk needs to run continuously to avoid message loss. In many enterprise environments, it might not be practical to run Splunk round the clock. Thus, this architecture is not recommended in production.

B. Co-hosting Splunk and Syslog Server

In this architecture, the syslog server and the Splunk run on the same server machine. The syslog server can be syslog-ng or any other custom solution.

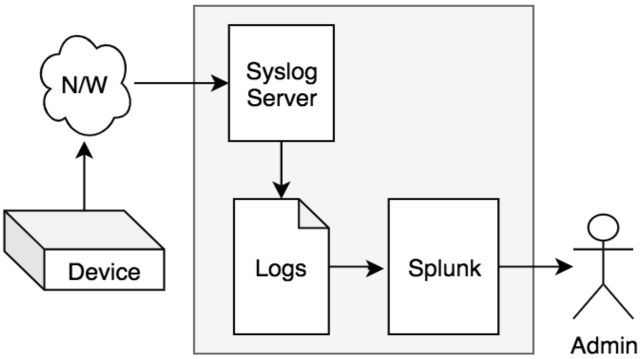

Fig 5: Co-hosting Splunk and Syslog Server

The syslog server is configured to listen to UDP port and the devices are configured to send the syslogs to the server over UDP. The syslog server is configured to write the received syslogs in to a directory on the disk. The Splunk is configured to listen to the directory.

This way, the Splunk is able to index the syslogs from the local directory. There is no message loss even if the Splunk is down.

C. Using Forwarders:

In this approach, the Universal Forwarder (F) component of Splunk is installed along with the standalone syslog server on “Machine-1” and the Splunk Enterprise Server is installed on “Machine-2”.

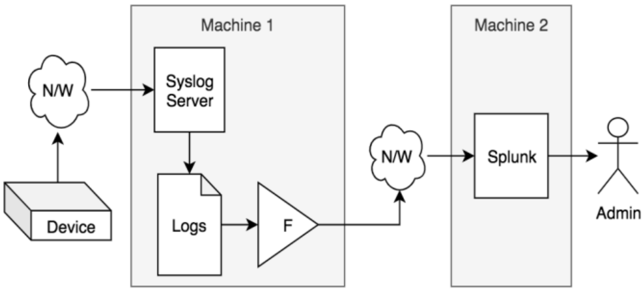

Fig 6: Integration using Splunk Forwarder

The devices are configured to send the syslogs to Machine-1 over UDP. The syslog server is configured to listen to UDP port.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication
© Copyright: All rights reserved.

The syslog server writes the received syslogs into a directory on the disk. The forwarder listens to the directory and sends them to the Splunk on Machine-2.

Since the syslogs are persisted on the disk, this setup assures fault tolerance.

V. THE PROPOSED ARCHITECTURE

Partitioning different types of data to separate indexes is one of the best practices of Splunk indexing [7]. This practice assures optimal reporting and action capabilities.

The following are the limitations of the architecture that is detailed in section IV.

The processor component of syslog server writes all the syslogs into a single directory.

The forwarder tags the syslogs from a directory to a single source type and forwards them to a single index.

Maintaining all the syslogs from across the infrastructure in one index is not a best practice.

The following is the aim of the proposed architecture:

The processor component of syslog server writes syslogs to different directories based on the configurable rules.

The forwarder forwards the syslogs from different directories to different indexes.

The Fig 7 details the proposed architecture.

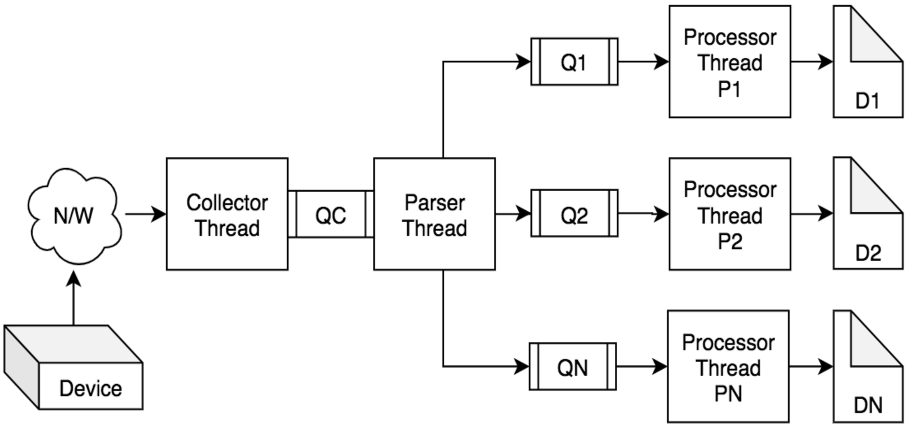

Fig 7: Proposed Architecture

Collector: It receives syslogs from multiple devices and writes to QC.

Parser: It parses the syslogs, applies the filter rules and writes them to different queues Q1 … QN. Each queue is meant for a different index on the Splunk. Administrator defines the rules. Multiple threads may run to scale up the parsing and filtering.

Processor: There will be N number of processors for N number of queues. Each processor reads the syslogs from the associated queue and writes them into corresponding directory. For instance, the processor P1 reads syslogs from queue Q1 and writes them to directory D1.

Forwarder: Each of the directories is configured as a Data Input Source for the forwarder with appropriate source type. Forwarder reads the logs from the directories and sends them to corresponding indexes. For instance, forwarder reads the logs from directory D1 and sends them to index I1.

This architecture avoids conflicts among the processor components during the I/O operations and thereby ensures smooth and scalable processing of the syslogs.

The Splunk is able to process the syslogs at much faster rate since multiple indexes are segregated on the lines of source types.

VI. CONCLUSION

This paper recounts the history of syslog protocol and patterns of its adaptation. The architecture and functionality of Splunk are presented. A case for integrating Splunk and Syslog system is established along with the popular patterns of integration. An architectural pattern is proposed to integrate Syslog servers with Splunk, to speed up detection and correction of anomalies, along with high scalability and resilience.

REFERENCES

1. C. Lonvick, “The BSD Syslog Protocol”, RFC 3164
2. R. Gerhards, “The Syslog Protocol”, RFC 5424
3. A. Okmianski, “Transmission of Syslog Messages over UDP”, RFC 5426
4. “Forwarding log messages to Splunk from syslog-ng™” (https://www.syslog-ng.com/documents/syslog-ng-with-splunk-use-cases-datasheet-132872.pdf)
5. Ashish Kumar, Tulsiram Yadav, “Advanced Splunk”, Packt Publishers
6. “Using Syslog-ng with Splunk” (https://www.splunk.com/en_us/blog/tips-and-tricks/using-syslog-ng-with-splunk.html)
7. “Managing Indexers and Cluster of Indexers”, (https://docs.splunk.com/Documentation/Splunk/8.2.1/Indexer/Aboutindexesandindexers)

AUTHORS PROFILE

Krishna Mohan Koyya, I hold B.E in Electronics & Communications Engineering and M.Tech in Computer Science and Engineering from Andhra University, Visakhapatnam. I worked for Hewlett-Packard, Wipro Technologies and Cisco Systems as Project Lead between 1998 and 2006. Later, I worked as the CEO at Sudhari IT Solutions, Bengaluru and as Assistant Professor in the Department of Information Technology at Sasi Institute of Technology & Engineering, Tadepalligudem. I floated my own technology consulting firm Glarimy Technology Services in 2008 and have been the Principal Consultant since then. Scalable Software Design and Architecture has been my area of interest.

Retrieval Number: 100.1/ijrte.B63070710221
DOI: 10.35940/ijrte.B6307.0710221
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication
© Copyright: All rights reserved.