Augtera The First Network AI for the Modern Enterprise

Solution Brief

LogAI for a New Era of Real-Time Log-based Network Operations

Introduction

Logs are a rich source of network telemetry information for network operations. While there are several tools that exist for historical analysis and querying of logs, they have not provided real-time anomaly detection capabilities that leverage the richness of text-based logs. This is increasingly relevant as operators want to find needles automatically and proactively in the growing haystack of log data. In addition, the volume, velocity, and variety of log messages calls for new approaches that generate structure and efficient analysis from unstructured text messages.

LogAI was developed to address Network Operations uses cases. To process in real-time, streaming log data, from any log, without latency or message drops. Specifically, to address the challenges of producing actionable insights from unstructured text data at such high volumes, where noise levels are so high, and the unknown unknowns are prevalent and undetected.

LogAI Flexible Data Ingestion

Syslog has long been the standard of network equipment vendors, however with the rise of cloud-based systems and message streaming technologies new logs formats, such as encoding logs in JSON are becoming more common. However, logs are generated by Cloud based systems as well as equipment vendors in many other non-standard formats, typically encoded as JSON. Further logs are collected, normalized, and distributed by operations teams in many other formats, typically also encoded as JSON. Kafka is an emerging message bus for many types of data within Network Operations environments.

LogAI supports all these scenarios today and at its core is agnostic to the format of the log messages when they are ingested, as all logs are normalized to a common internal format.

Image 1: Structure of a Syslog message, one of the log format supported by Augtera

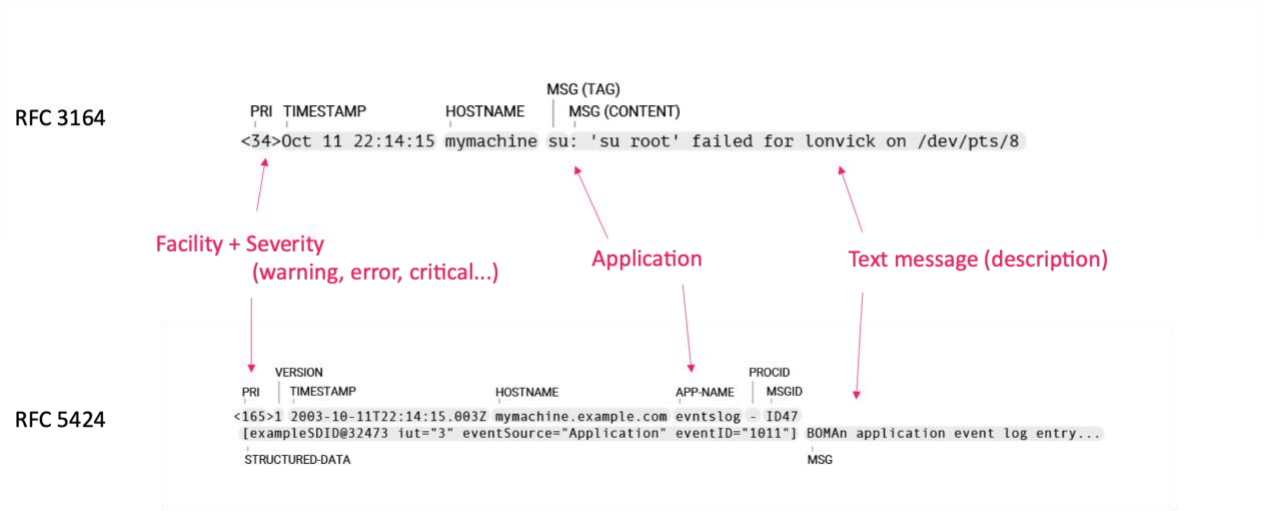


Image 2: Structure of a log message received in JSON format

Logs can be ingested directly from devices to Augtera platforms or forwarded by existing customer log tools such as Splunk or ElasticSearch.

Log Ingestion from third party Log collectors

Augtera Network AI can ingest logs forwarded by third party tools such as Splunk and ElasticSearch.

For example, Splunk forwarders can forward raw data to non-Splunk systems packaged in standard syslog. The syslog output processor sends RFC 3164-compliant events to a TCP/UDP-based server and port, making the payload of any non-compliant data RFC 3164-compliant. It is also possible to configure Splunk to send a subset of data to Augtera, for example only the data from hosts whose names begin with "sfo". A second option when working with large data sets is to send events from the Splunk platform directly to Kafka for ingestion.

Similarly, it is possible to forward log data from ElasticSearch using Logstash. You can create a Logstash pipeline by stringing together plugins (inputs, outputs, and filters) in order to process data. A very basic pipeline might contain only an input and an output to forward all logs from ElasticSearch to Augtera Network AI Platform. You can use two options as Output plugin in Logstash to forward logs to Augtera platform: syslog output plugin or Kafka output plugin.

Enabling AI/ML on Logs to Transform Network Operations

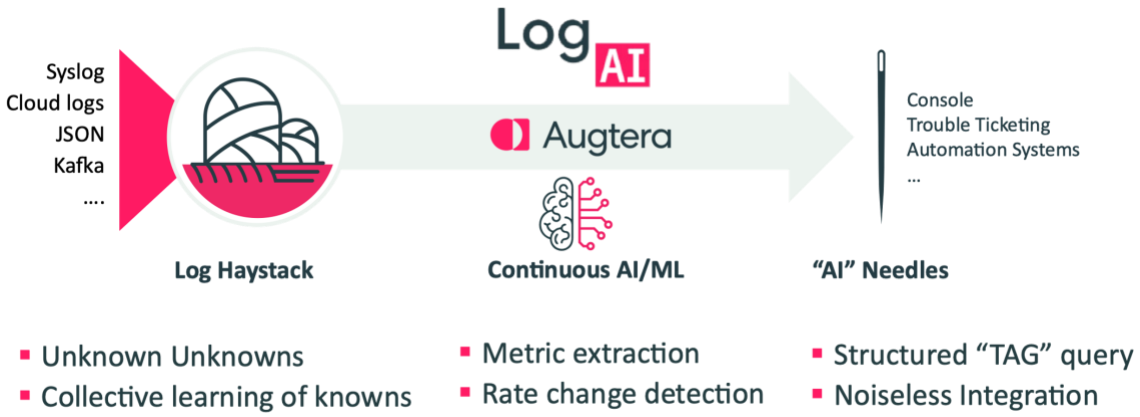
LogAI changes the log experience through high-performance, high-efficiency real-time AI/ML processing of streaming log messages from Syslog, or other non-standard format Logs such as Cloud Logs encoded as JSON and ingested using Kafka or Augtera APIs. The result is actionable and automated AI “needles” from the Log data haystack.

Network operations teams today either have no ability to process Logs in real-time or can only create limited rules to identify a set of known Log signatures. This leaves several gaps:

- Anomaly symptoms that have never been seen before i.e., the unknown unknowns
- Anomalies that are never acted on because they are lost in the noise
- Immediate implementation of new known Log signatures due to cumbersome rules-based approaches and/or lag time for internal software development and/or other inter-organization / process issues
- Extraction of metrics for anomaly detection & visualization
- Message rate-based anomaly detection with high fidelity and low noise
- Elimination of noise prior to signaling consoles, trouble ticket systems, and automation systems

Augtera Networks LogAI is a vastly different approach.

Image 3: LogAI Purpose-Built for Network Operations



Augtera Network AI has implemented three key automated capabilities to change the paradigm of Log observability:

- The first is to detect authentic unknown logs because they never happened either in the entire network, group of devices, at device level or any aggregation level set by the operator. These rare logs are called Zero Day Anomalies
- The second is to classify in real time logs known for their operational value based on the classifier library shipped by Augtera and constantly updated based on collective learning across the customer base.
- The third is to automatically detect any abnormal rate in log patterns, send alerts and point the root cause of these behavioral changes

ML-based Zero Day Anomaly detection

Zero Day Anomalies is a Natural Language Processing (NLP) based capability that detects new, rare, and unique log messages the first time they appear. So, if they are the symptom of a current or future incident, action can be taken immediately. The capability focuses on rare messages that have not been seen within a customer-defined period.

Determining what is a new or rare log message is not trivial though. There are so many nuances in log messages that simple text processing will lead to false results. To realize this capability, LogAI uses a high-performance, high-efficiency Natural Language Processing implementation, purpose-built by Augtera Networks. This is needed because semantic understanding is necessary to determine what is a new or rare log message.

The algorithms developed by Augtera are using NLP techniques such as tokenization, normalization, corpus analysis, stop words detection, statistical language modeling, bag of words, and similarity measures. The similarity measures technique in particular is one that is the result of significant proprietary R&D by Augtera and is used for determining the similarity of text strings to each other. This technique is designed for building online models from historical Big Data. These models are incrementally updated every few minutes and are leveraged for real-time prediction. These models can be built at different levels of aggregation based on metadata. For example a model can be built for all logs for a specific application and log severity in a data center, or across the entire network, or within a family of devices for a vendor or per device. These models can also be built for non-networking use cases.

Image 4: some NLP techniques relevant to Zero Day anomaly detection

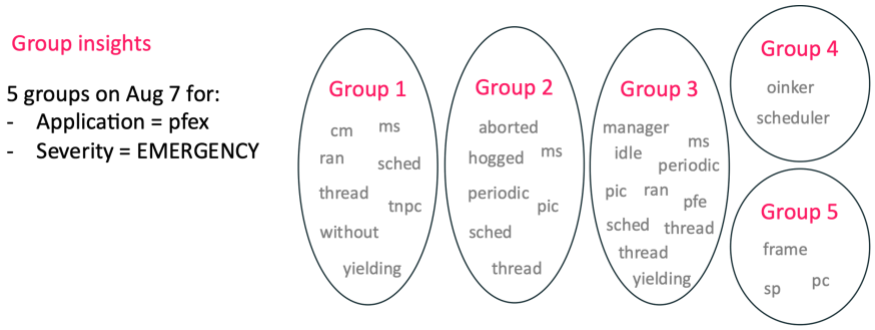
NLP Function	DESCRIPTION
Tokenization	Early step in NLP process. Splits longer strings into smaller pieces / tokens.
Normalization	Converting case, removing punctuation, etc.
Corpus	Collection of texts
Stop Words	Words filtered out because they contribute little to meaning “the, and, a….”
Statistical language modeling	Assigning probabilities to sequences of words
Bag of words	Omits grammar and word order
Similarity measures	Measure the similarity of strings

When a log message is ingested, it will be processed in real time following a workflow using the mentioned NLP techniques for semantic pattern analysis and using one or more of the models mentioned to predict similarity. As a result, this new message will either be categorized as belonging to an existing pattern or if it deviates “largely” from the existing semantic patterns, it will be considered as an outlier and flagged as an anomaly.
These algorithms have been proven in large scale production deployments to produce few and very high fidelity anomalies.

Let’s take an example from a production deployment with more than 10,000 switches generating about 4.5M syslogs per hour. In this configuration, we use Zero Day anomaly

detection for all combinations of log applications and severity levels for the entire network. As seen in the below figure after 21 days of learning, on August 7th, the model contains 5 different vocabulary groups fitting all past log messages issued by application “pfex” with severity “EMERGENCY”.

Image 5: Group insights for syslog messages from application “pfex” and severity=”EMERGENCY” in a 10K device network with 4.5 M slogs/hour



Suddenly, on Aug 8th at 11:09:17 precisely, a new log message was received from a single device in the network. After going through the real-time pipeline including the complete NLP workflow semantic analysis and using the appropriate model to predict similarity to existing log messages the platform cannot match “closely enough” the new log message with any existing group. As a result this rare loglog is flagged as Zero Day anomaly.

Image 6: Zero-day anomaly detection

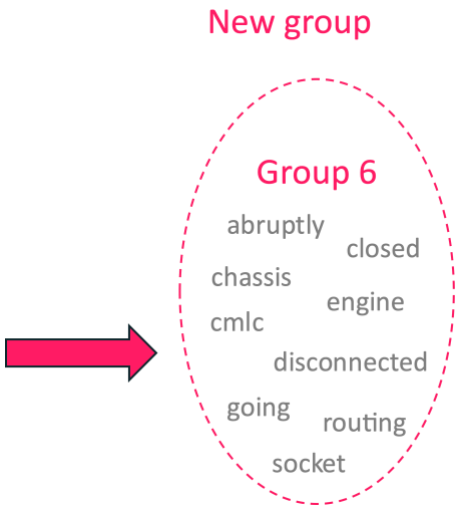
Aug 8: “Routing Engine disconnect“ rare syslog not matching a group

Timestamp	Application	Device Name	Severity	Description
Aug 08, 11:09:17AM - 11:09:18AM	pfex	[redacted]	EMERGENCY	CMLC: Going disconnected; Routing engine chassis socket closed abruptly

The anomaly can be notified in real time to operations, or even automatically ticketed in customer ticketing system such as Service Now in order to take immediate remediation action as needed.

However, the workflow doesn’t stop here. There are two more important steps.
The first is performed automatically by the NLP algorithm. It consists of creating a new semantic group for this message in order to maintain high precision and accuracy in finding future unknown messages.

Image 7: Automated new semantic group creation for application "pfex" and severity="EMERGENCY"



The second action is optional and requires operator action: The operator can choose to add a real time classifier to automatically recognize and act on this specific message in the future as it may no longer be detected as a Zero Day Anomaly. Log classification is explored in detail in next chapter.

Real Time Log Classification and Data/Metric Extraction

In the logs coming from the network, there are some that are very important for operations. Sometimes these messages carry critical information such as embedded metrics or metadata that needs to be extracted and leveraged. To address these needs, Augtera s platform supports a real time log classification engine that can be configured to detect known relevant operational logs and create structured events potentially enriched with data extraction.

Augtera platform provides hundreds of classifiers that are the result of large scale production deployment over the past several years, with frequent updates based on classifiers configured by operators across Augtera's customer base. This network effect represents a huge benefit to customers as they immediately leverage automated added-value classification for network operations without any effort. Of course, this global Augtera semantic knowledge database can be enriched and customized by any customer with its own experience.

As an example, a BGP session going down generates in some systems a log message with severity INFO which is completely diluted in the millions of logs. Therefore, as we know that identifying BGP session down is important, a classifier will capture this log to create an event

category, as we can see in following example with the classifier “clf47”, using a filter on severity and a REGEX matching expression on syslog text message.

Image 8: real time classifier “clf47” detecting BGP session down logs

```
536     "clf47": {
537       "membership": [
538         "syslog"
539       ],
540       "profile": "profile47",
541       "match": {
542         "expression": [
543           "EventSource == syslog",
544           "( (Severity == INFO) && (Description =~ \".*BGP.*ADJCHANGE.*neighbor.*Down.*\") )"
545         ]
546       }
547     },
```

In addition to the classification, the platform can enrich the newly created events with metadata and metrics coming from the original log text.
For example, the BGP session down log contains the peer IP address in the message text. The classification engine can be configured to extract that IP address and populate the Augtera dimension peerIP. This way operations will have full visibility and granularity on peering status.

Log messages sometimes also contain metrics. However, when embedded in an unstructured log message, these metrics are hard to extract, analyze and visualize with network operations tools.

Augtera enables the extraction of metrics from log messages. Metrics can then be used by the rest of the platform for anomaly detection, gray failure detection, incident root cause identification, and notification. Metric extraction can be used for any log message, regardless of where in the network the message pertains to. The extraction process is not specific to a set of messages.

Image 9: example of metric extraction from raw log messages

These dimensions (metadata or metrics) resulting from the classification can be leveraged with all Machine learning features such as the rate-based ML-anomaly detection, exactly like raw log data, as we will see in next chapter.

Log and Classified Log rate-based anomalies

Compared to threshold-based anomaly detection, machine learning anomaly detection reduces noise and enables proactive action.
Augtera's proprietary ML automatically learns the normal patterns for every Aggregated Object defined by default or following customized configuration. This includes traffic, discards, errors, optical tx/rx power, queues, CPU, route tables, flows, retransmits, etc. Similarly, the ML is also able to identify anomalies related to changes in rate of log patterns. This deep understanding of the network gives operators advance insight when the network begins to misbehave, not when it has reached a critical level. It also gives operators visibility to everything of operational relevance that has changed.

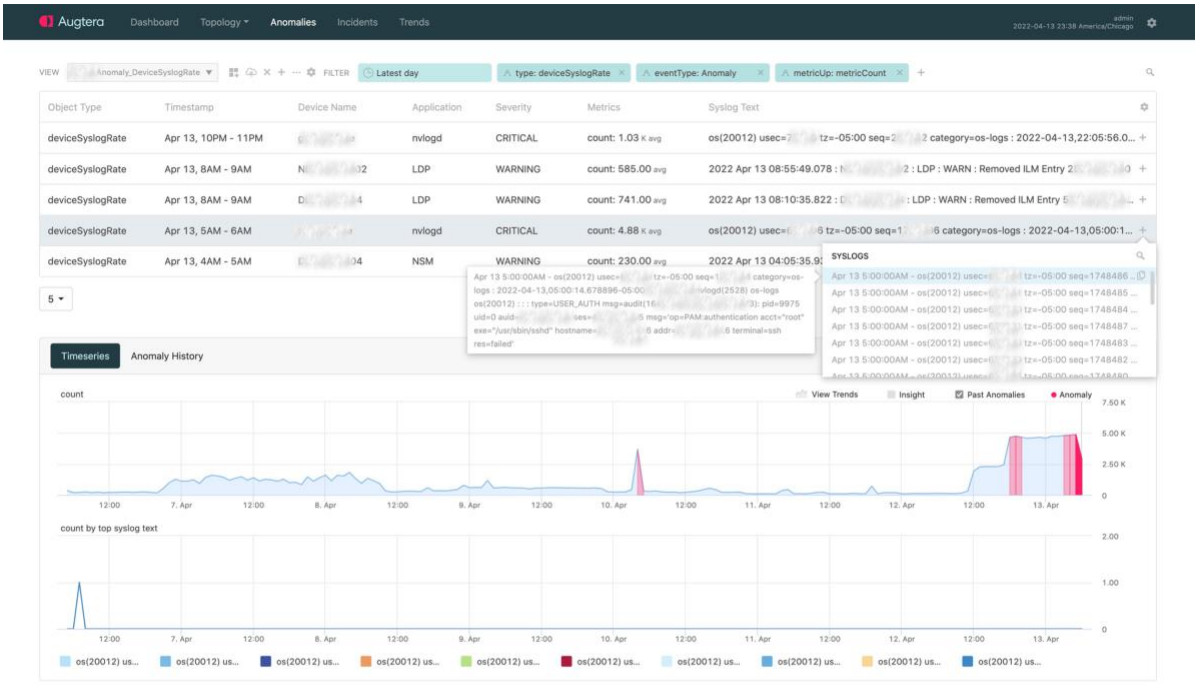
Aggregate Object defines how to group data for an instance of a ML model. Consider two different examples:

1. Aggregate Object deviceSyslog that groups all logs with same device, application and severity
2. Aggregate-Object networkSyslog that groups all logs with same application and severity across all devices

If there was an application “kernel” and severity “ERROR”, and 10 devices in the network, then for this application, severity combination, there will be 10 instances for deviceSyslog, while only 1 instance for networkSyslog

Augtera Network AI will build a distinct model for each instance to detect abnormal spikes of log rate.

Image 10: Example of rate-based anomaly detecting abnormal failed attempts to login root



The rate-based anomaly detection can be used similarly on classified logs. For example, ML can detect abnormal rate of BGP session down events created as a result of log classification.

These anomalies result in actionable events for operations. Let’s describe two example scenarios to illustrate the applicability of this revolutionary paradigm.

Incident Scenario 1:

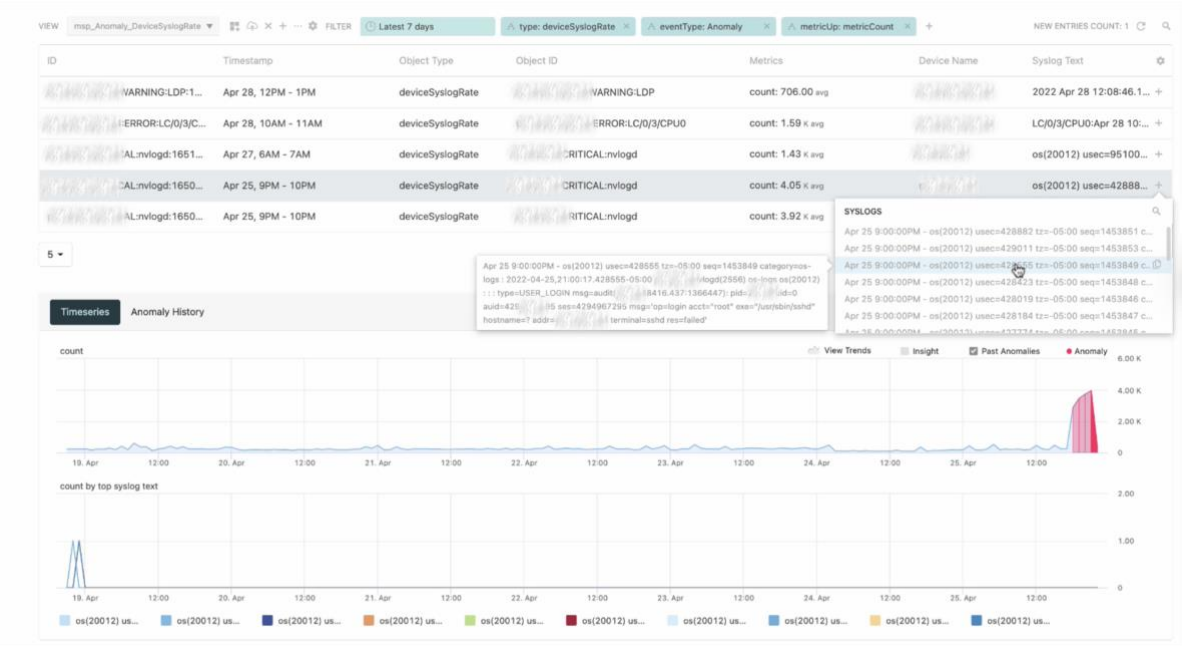
- Configuration change in a Firewall inadvertently causes Google DNS to be blocked
- Firewall classified log messages of blocked traffic show a dramatic increase in rate
- Augtera pro-actively detects the log rate change for this specific classified log with a ML-based anomaly

- Augtera analytics identifies the IP address that is being blocked from the log description.

Incident Scenario 2:

- A configuration change in the network opens up router SSH (port 22) to the external IP addresses
- Attackers using port scans find the open port and attempt to login
- An attacker does brute force login attempts across the network at the same time
- Attacks show up in logs, and Augtera detects rate change with a ML-based anomaly

Image 11: Production example of scenario 2 showing a rate-based anomaly applied to an Aggregate Object related to a specific device, application "nvlogd" and severity "CRITICAL"

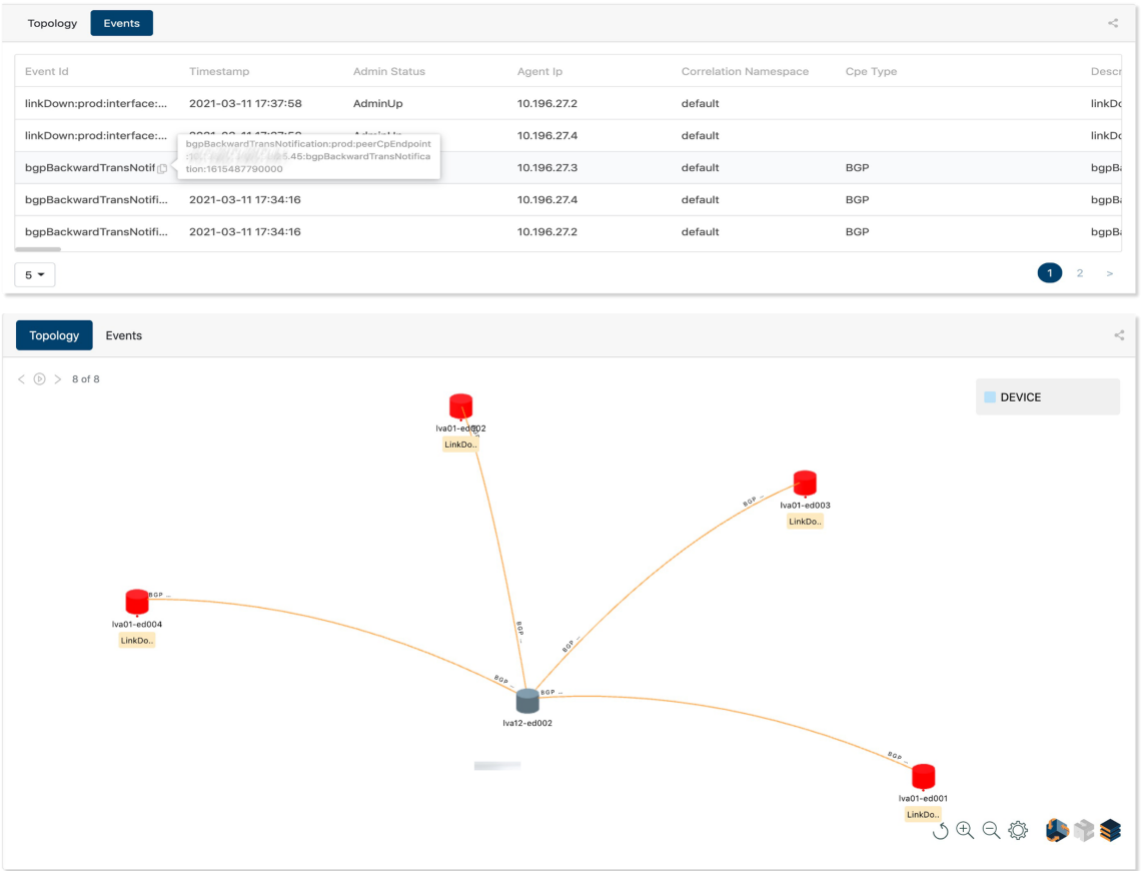


Network model-based Multi-layer Autocorrelation

Augtera Network AI correlates network events and Augtera anomalies across multiple data sources automatically using a purpose-built machine learning algorithm that is multi-layer network topology aware. This enables operators to proactively receive notifications of correlated network issues with high fidelity context, further reducing ticketing noise and enabling rapid root cause analysis and remediation.
The auto-correlation can mix together any kind of events such as classified logs and Zero Day anomalies that are described in this solution brief, as well as other events such as SNMP traps, or Augtera ML-based anomalies on hundreds of supported metrics.

Following example shows an incident created automatically by the correlation of 4 link down and bgp Backward Transition Notification events on four distinct devices, all connected using BGP sessions with a fifth “suspicious” device that remained silent.

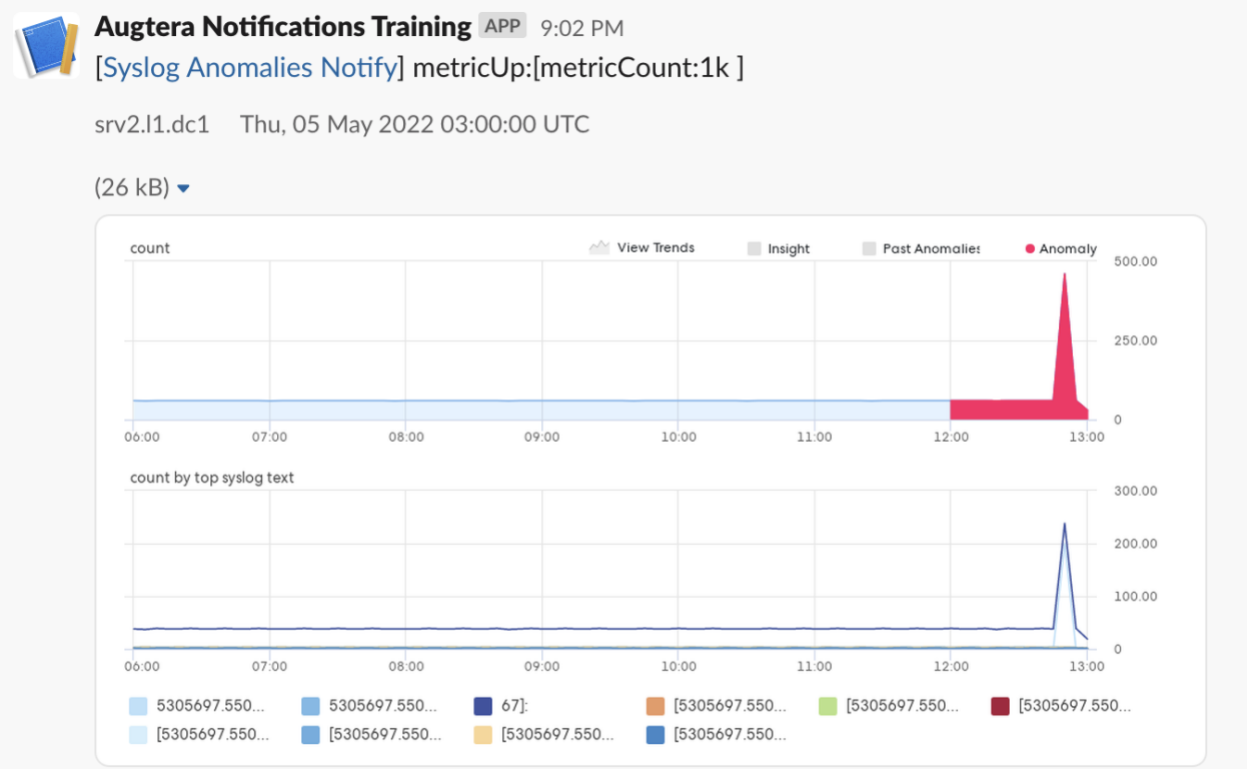
Image 12: Production example of one incident correlating link down and BGP flaps into one single incident



Automated Workflows and Real-Time Notification

Augtera Network AI implements simplified and flexible workflows for different teams driven by operator intent supporting ad-hoc analytics, notifications (slack, syslog, kafka), and automated ticketing integration (Service Now). Out of the box and custom Views with rich metadata aware filters (e.g., operator can choose to notify only certain types of anomalies on certain types of devices) are used to define what types of anomalies, events and auto-correlated incidents, should be notified to consoles, ticketing systems, and automation systems.
Integration with third-party tools such as Splunk or ElasticSearch can be done using a syslog message directly sent by Augtera as a notification method, or by using a kafka topic.

Image 13: Example of a log rate-based ML anomaly notified with a slack message



Conclusion

Logs contain a wealth of information that operators increasingly want integrated into real-time workflows. In addition, Network Operations teams can no longer be simply reactive to incidents, they must proactively detect, see potential incidents before they occur, and prevent those incidents from ever happening.

Augtera LogAI is already providing significant value to operators in production deployments. We have described earlier some examples from large scale deployment with 4.5M logs per hour. However small or medium networks can also benefit from these AI capabilities. For example, in a production network with less than 100K/hour logs, Zero Day Anomaly Detection has detected 17 unique and actionable logs among 12 millions logs. Most of the detected events uncovered issues that were not seen at all before impacting service and customer perception. Sometimes, this approach is complementary to existing tools by reducing Mean-Time-to-Detect (MTTD). In a recent live deployment, while benchmarking Augtera Network AI with an existing tool, one detected zero-day anomaly allowed the customer to open a ticket 45minutes before the legacy tool.

The Augtera Network AI platform, including LogAI, was developed to not only enable network operations teams to react faster, but to reduce the need to react by eliminating noise and preventing future incidents.

Related Links

- LogAI Solution
- Solution Brief: LogAI Integration with Splunk and Elastic Search
- Blog: Machine Learning Anomaly Detection – Beyond Thresholds
- Augtera Data Sheet