
ONE IDENTITY™

Simplifying log management, not just for security logs

Peter Czanik

ONE IDENTITY™

Overview

- Central logging
- Dedicated logging layer
- Using syslog-ng
- New directions: Kafka and OpenTelemetry

What is logging?

- Logging: recording events. For example:

```
Jan 14 11:38:48 linux-0jbu sshd[7716]: Accepted
publickey for root from 127.0.0.1 port 48806 ssh2
```

Why central logging?

Ease of use

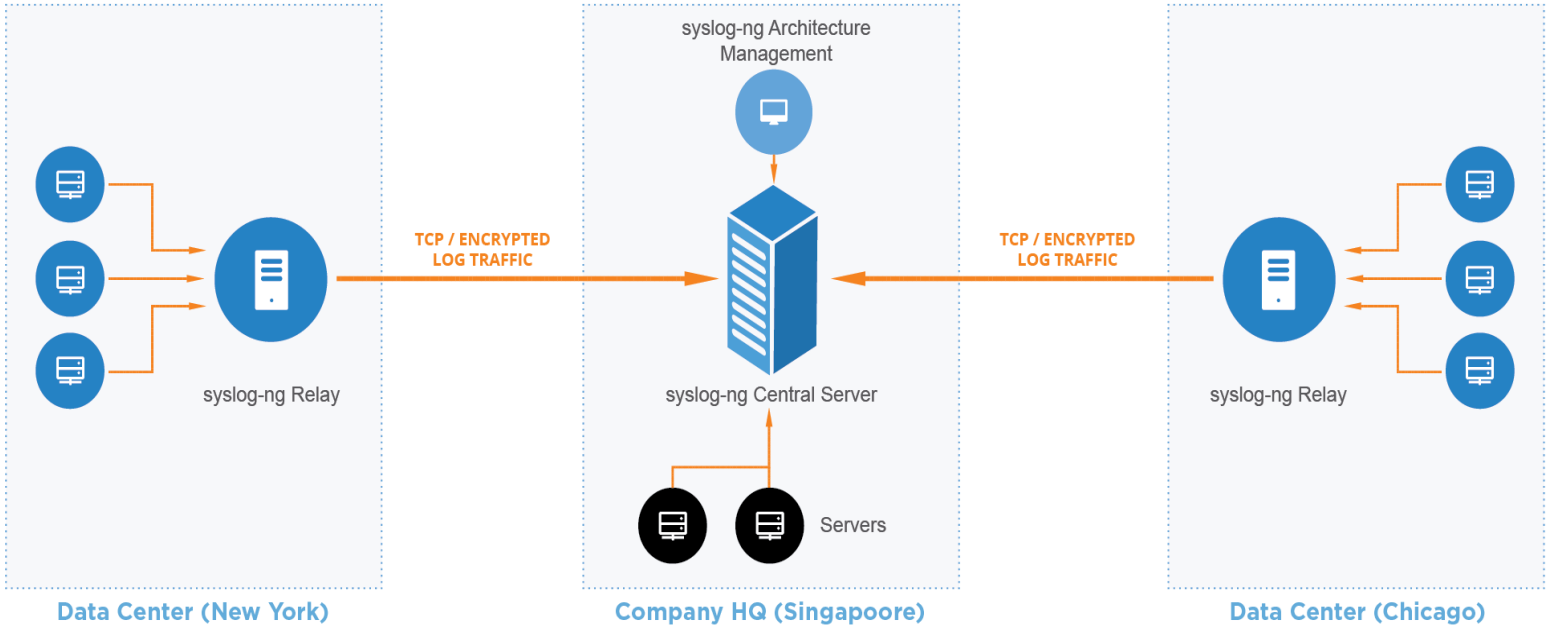
One place to check instead of many

Availability

Even if the sender machine is down

Security

Logs are available even if sender is compromised



Ease of use

- Checking logs on 2 hosts takes 2x more time.
  - Probably still easier than implementing central logging… ☺
- Checking logs on 10 hosts takes 10x more time.
  - Central logging saves a lot of time.
- Checking logs on 100 hosts takes 100x more time.
  - Security and operations *cannot* function without central logging.

Availability

- Logs are available even when sender host is down.
- You can start debugging without first bringing the host back to life.

Security

- Logs are often modified or deleted locally when a host is compromised.
- Logs collected centrally cannot be modified (or can only be modified later) by the attacker.
- Logs can be used to reconstruct what happened.

Compliance

- Logs are required everywhere, especially in health and finance.
- Central logging is mandated by various compliance regulations.
- Log retention sometimes even for years.

- Ease of use and availability for your own benefit, just as security.

Growing bigger

- More computers/services -> more logs
- Multiple tools to analyze logs
  - Each with its own log collector
- Long-term storage
  - Databases in analytics software use more resources

Dedicated log management layer

- A single tool to collect logs
  - Less computing resources
  - Less network resources
- Routes log messages to various analytics tools
  - Message parsing, enrichment, filtering
  - Each destination receives only what it really needs
- Archives logs for long-term storage (compliance)

Possible tools

In Linux distributions:

- syslog-ng
- Rsyslog

Other open source:

- NXLog
- Logstash
- Fluentd

What is syslog-ng?

- An enhanced logging daemon with a focus on portability and high-performance central log collection. Originally developed in C.

The four major roles of syslog-ng (and others)

- Collect
- Process
- Filter
- Store (or forward)

Role: Collect

Collect system and application logs together: contextual data for either side

- A wide variety of platform-specific sources:
  - /dev/log & Co., Journal, Sun streams
- Receive syslog messages over the network:
  - Legacy or BSD (RFC3164) and new (RFC5424) standard, UDP/TCP/TLS
- Logs or any kind of text data from applications:
  - Through files, sockets, pipes, application output, etc.

- Python source: Jolly Joker
  - HTTP server, Kafka source, etc.

Role: Process

- Classify, normalize, and structure logs with built-in parsers:
  - CSV parser, PatternDB, JSON parser, key=value parser
- Rewrite messages:
  - For example: anonymization
- Enrich data:
  - GeoIP
  - Additional fields based on message content
- Reformatting messages using templates:
  - Destination might need a specific format (ISO date, JSON, etc.)

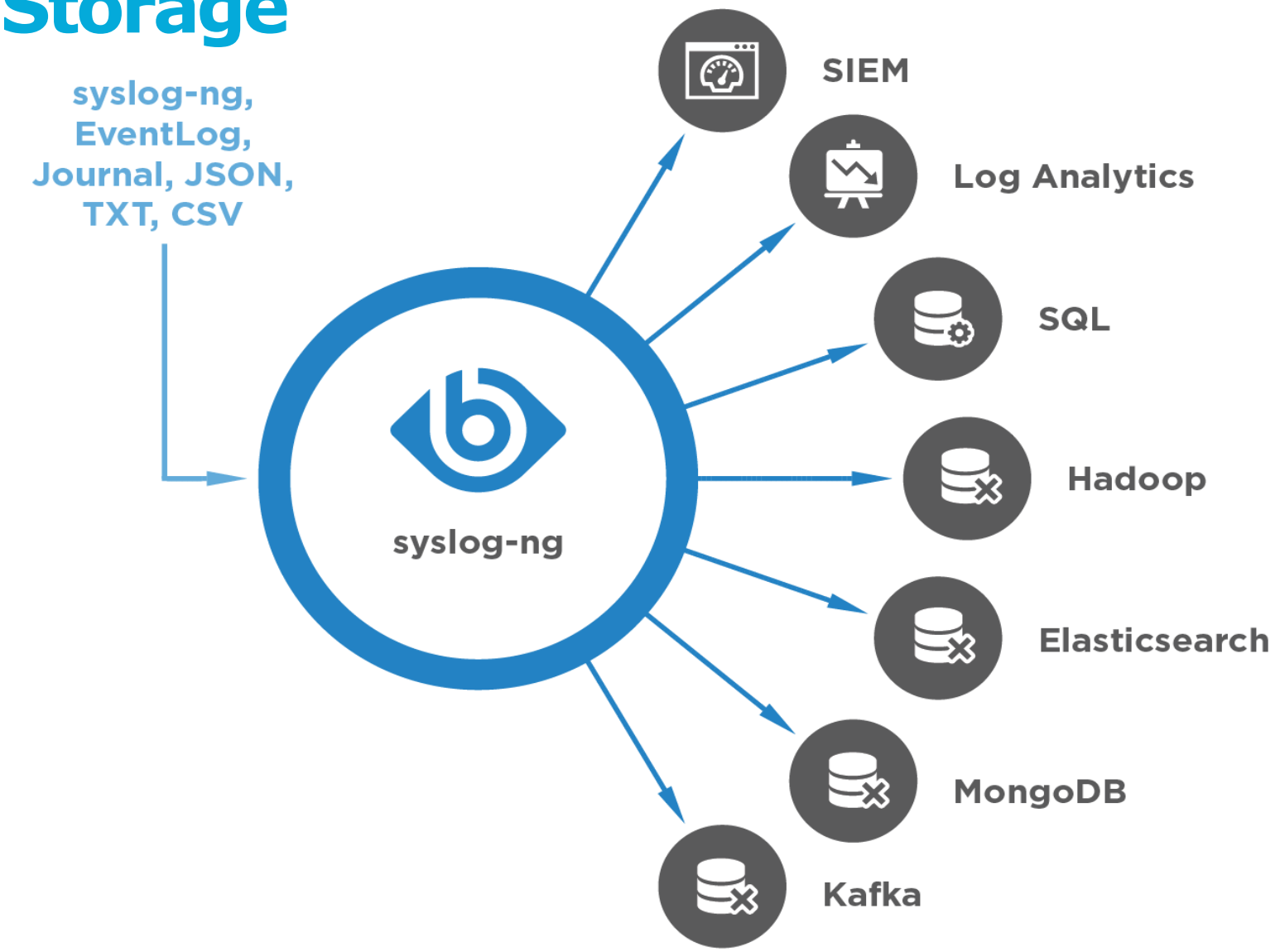
- Python parser:
  - All the above, enriching logs from databases, filtering

Role: Filtering

- Main uses:
  - Discarding surplus logs (not storing debug-level messages)
  - Message routing (login events to SIEM)

- Many possibilities:
  - Based on message content or parameters
  - Using comparisons, wildcards, regular expressions, and functions
  - Combining all of these with Boolean operators

Role: Storage



ONE IDENTITY™

Where is syslog-ng used?

- Available in most Linux distros and BSD variants
- All Kindle e-book readers
- BMW electric cars
- Airbus aircrafts
- Turris Omnia routers
- & many more...

Freeform log messages

- Most log messages follow the format of:
  date + hostname + text

```
Mar 11 13:37:56 linux-6965 sshd[4547]: Accepted
keyboard-interactive/pam for root from 127.0.0.1
port 46048 ssh2
```

- Text = English sentence with some variable parts
- Easy to read by a human
- Difficult to create alerts or reports

Solution: structured logging

- Events represented as name-value pairs. For example, an ssh login:

```
app=sshd user=root source_ip=192.168.123.45
```

- syslog-ng: name-value pairs inside
  - Date, facility, priority, program name, pid, etc.

- Parsers in syslog-ng can turn unstructured and some structured data (CSV, JSON) into name-value pairs

Message parsing

- Creates name-value pairs from interesting message parts
- Unstructured logs: PatternDB parser, regular expressions
- Structured logs: CSV, JSON, XML, key=value, etc.
- Combined parsers in SCL: sudo, Apache, Cisco, etc.

Why parsing is important

- Filtering
  - Routing / alerting
  - Discarding surplus logs
- Storage
  - Formatted differently
  - Save only part of the message

- Later (not in syslog-ng) in reporting, alerting, dashboards

Basic syslog-ng configuration

```
@version:3.23
@include "scl.conf"
source s_sys { system(); internal();};
destination d_mesg { file("/var/log/messages"); };
filter f_default { level(info..emerg) and not (facility(mail)); };
log { source(s_sys); filter(f_default); destination(d_mesg);};
```

Sample configuration

- Collects logs from Windows.
- Parses the XML-formatted logs.
- Uses PatternDB to find specific events and parse IP addresses from log messages.
- Sends alerts when there is a match.
- Saves all logs to Elasticsearch for easier search, dashboards and short-term storage.
- Saves all logs as text files for long-term storage: compliance.

Source and parsers

```
# source for Windows clients, RFC5424
source s_win {
  syslog(port(601));
};
# xml parser for Windows XML logs
parser p_xml {
  xml(prefix('winxml.'));
};
# patterndb parser for OI Network Agent connection logs
parser p_connect {
  db-parser(file("/opt/syslog-ng/etc/oina.pdb"));
};
```

ONE IDENTITY™

Elasticsearch destination

```
# Elasticsearch destination
destination d_elasticsearch {
  elasticsearch-http(
      index("syslog-ng")
      type("")
      user("elastic")
      password("Do2oFMbINS3hzfmR8uIV")
      url("https://172.16.167.182:9200/_bulk")
      template("$(format-json --scope rfc5424 --scope dot-nv-pairs
      --rekey .* --shift 1 --scope nv-pairs
      --exclude DATE @timestamp=${ISODATE})")
      tls(peer-verify(no))
  );
};
```

ONE IDENTITY™

File and Slack destination

```
# destination for Windows logs
destination d_fromwin {
  file("/var/log/fromwin");
  file("/var/log/fromwin.json" template("$(format-json --scope rfc5424 --scope dot-nv-pairs
        --rekey .* --shift 1 --scope nv-pairs)\n") );
};

# Slack destination
destination d_slack {
  slack(
    hook-url("https://hooks.slack.com/services/T06PxxxFE/B06xxxA/gywQkG8XXXmJOC")
    template("OI Network Agent connection at ${DATE} to ${oi.targethost}")
  );
};
```

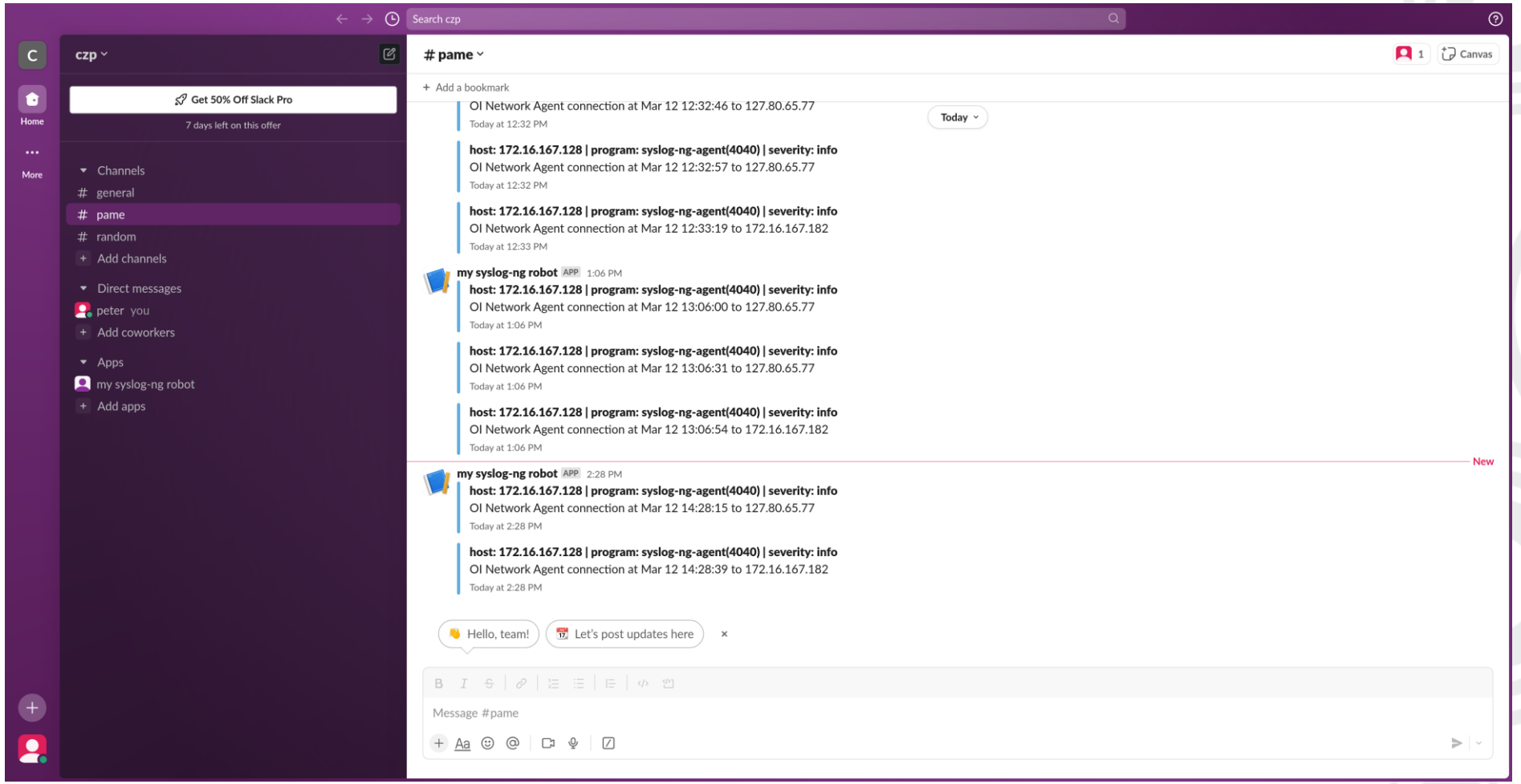
ONE IDENTITY™

Logpath: connecting building blocks together

```
# log path for Windows logs
log {
  source(s_win);
# syslog-ng-agent logs do not come in XML format
# only parse the rest
  if ("${PROGRAM}" ne "syslog-ng-agent") {
    parser(p_xml);
  };
  parser(p_connect);
# send alert if there is a connection through the OI Network Agent
# this macro is only created from connection logs
  if ("${oi.targethost}" ne "") {
    destination {file("/var/log/alert" template("${DATE} OI Network Agent connection to
${oi.targethost}\n"));};
    destination(d_slack);
  };
  destination(d_fromwin);
  destination(d_elasticsearch);
};
```

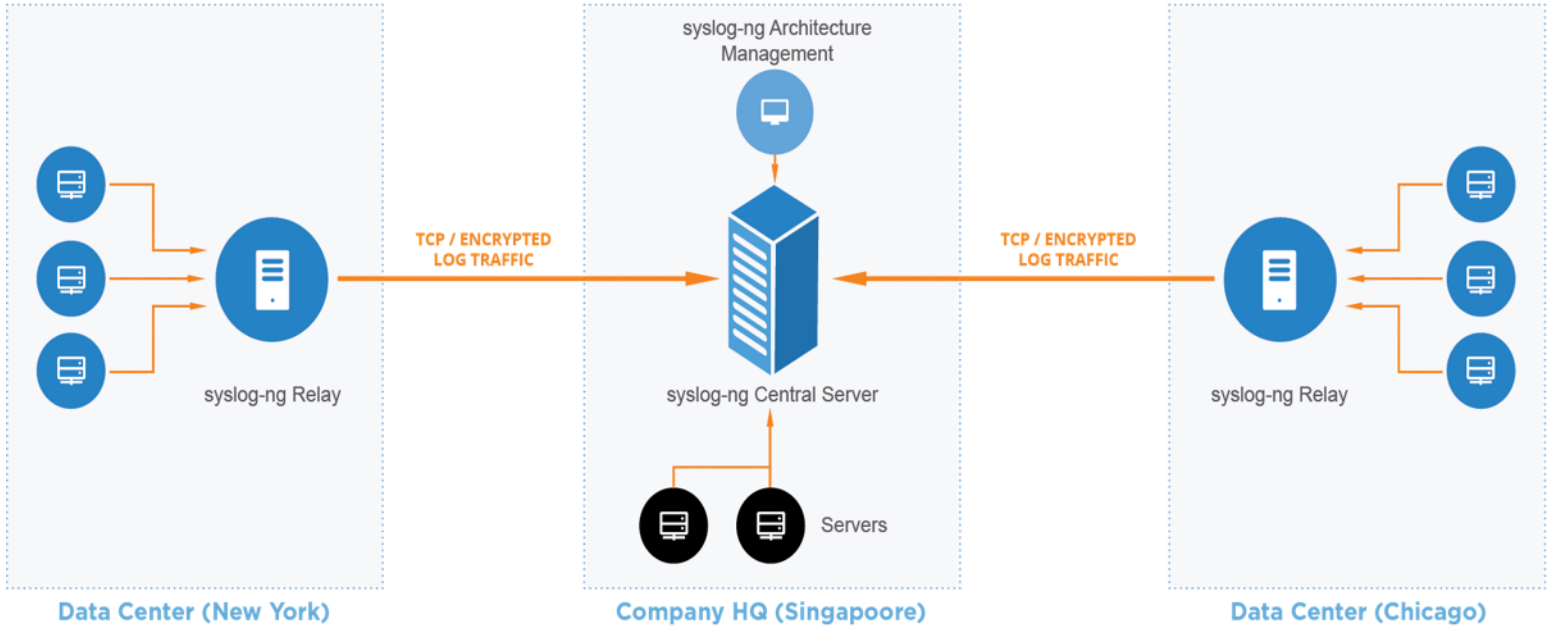
ONE IDENTITY™

Alerts from syslog-ng in Slack



How scaling works using syslog: relays

- Additional layer between syslog clients and server
- Must have for UDP logs
- Ensures that logs leave clients as soon as possible



ONE IDENTITY™

Why OpenTelemetry?

- Today, logs, traces and metrics are often collected separately.
- OpenTelemetry combines the three into a single protocol and application.
- An industry-standard, supported both by commercial vendors and open source projects.

OpenTelemetry in syslog-ng

- syslog-ng can:
  - Collect
  - Parse
  - Send

  OpenTelemetry data
- Scaling with relays, as with syslog

- Needs up-to-date dependencies: only available in recent Linux distributions

OpenTelemetry between syslog-ng instances

- syslog-ng-otlp() destination and source forwards logs with all related name-value pairs.
- Scales better than using the syslog protocol due to parallelization.
- Compression allows you to save bandwidth.

Kafka

- Popular data pipeline
- A single tool to move all data within an organization
- Syslog-ng can be used on both ends of a Kafka data pipeline

- Syslog-ng collects and puts logs into Kafka
- Scaling is within Kafka
- syslog-ng pulls logs from Kafka, processes them and forwards them to various destinations

Summary

- Central logging is useful for you, and is also a requirement in many industries.
- A dedicated log management layer can save resources and simplify your infrastructure.
- syslog-ng is one of the tools to implement central logging.
- Implementing OpenTelemetry can simplify your infrastructure even further.
- Kafka is often used as a data pipeline, syslog-ng can both read and write Kafka topics


ONE IDENTITY™