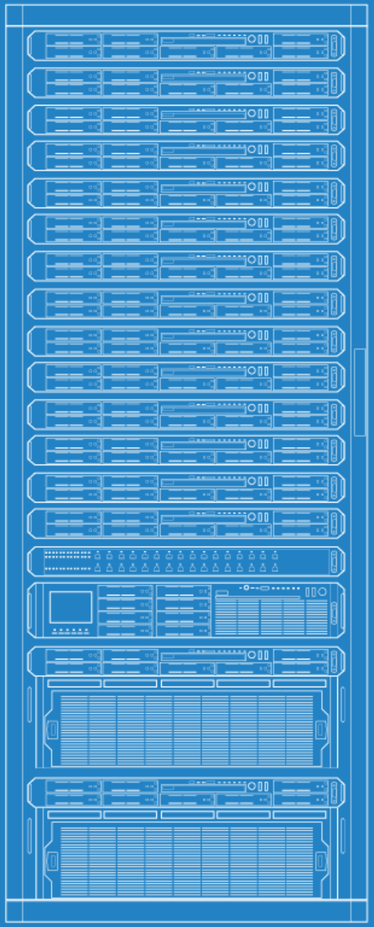
SCALING YOUR LOGGING INFRASTRUCTURE USING SYSLOG-NG

FOSDEM 2017
Peter Czanik / Balabit



ABOUT ME

- Peter Czanik from Hungary
- Community Manager at Balabit: syslog-ng upstream
- syslog-ng packaging, support, advocacy

Balabit is an IT security company with development HQ in Budapest, Hungary

Over 200 employees: the majority are engineers

syslog-ng

Logging

Recording events, such as:

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

syslog-ng

Enhanced logging daemon with a focus on high-performance central log collection.

WHY CENTRAL LOGGING?

EASE OF USE

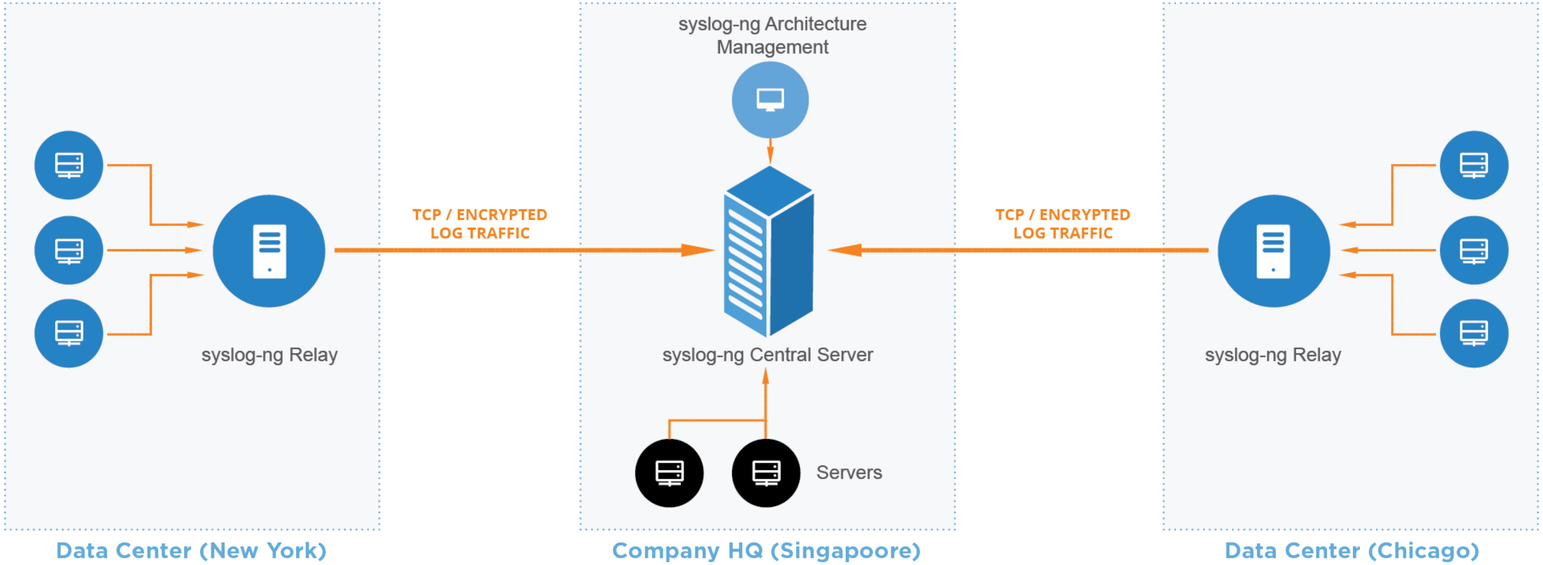
one place to check
instead of many

AVAILABILITY

even if the sender
machine is down

SECURITY

logs are available even
if sender machine
is compromised



BALABIT

MAIN SYSLOG-NG ROLES

collector

processor

filter

storage
(or forwarder)

ROLE: DATA COLLECTOR

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 RFC5424, UDP/TCP/TLS

Logs or any kind of data from applications:

- Through files, sockets, pipes, etc.
- Application output

ROLE: PROCESSING

Classify, normalize and structure logs with built-in parsers:

- CSV-parser, DB-parser (PatternDB), JSON parser, key=value parser and more to come

Rewrite messages:

- For example anonymization

Reformatting messages using templates:

- Destination might need a specific format (ISO date, JSON, etc.)

Enrich data:

- GeoIP
- Additional fields based on message content

ROLE: DATA FILTERING

Main uses:

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

Many possibilities:

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

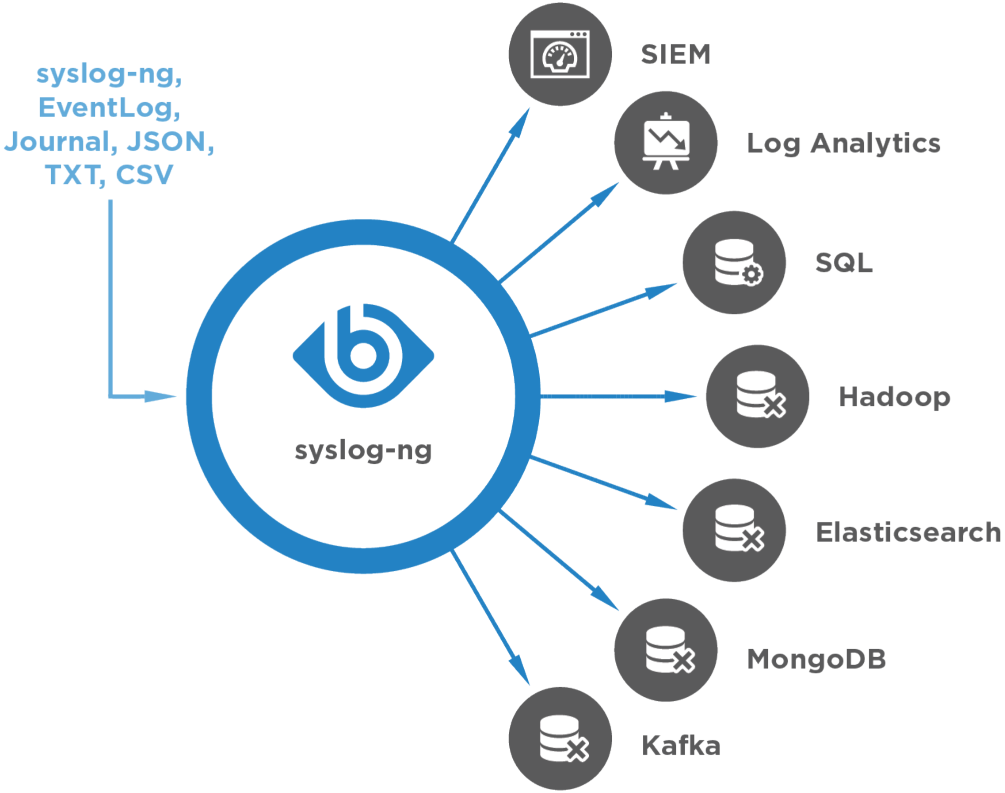
ROLE: DESTINATIONS

“TRADITIONAL”

- File, network, TLS, SQL, etc.

“BIG DATA”

- Distributed file systems:
  - Hadoop
- NoSQL databases:
  - MongoDB
  - Elasticsearch
- Messaging systems:
  - Kafka



FREE-FORM LOG MESSAGES

Most log messages are: 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 process them with scripts

SOLUTION: STRUCTURED LOGGING

- Events represented as name-value pairs

- Example: an ssh login:

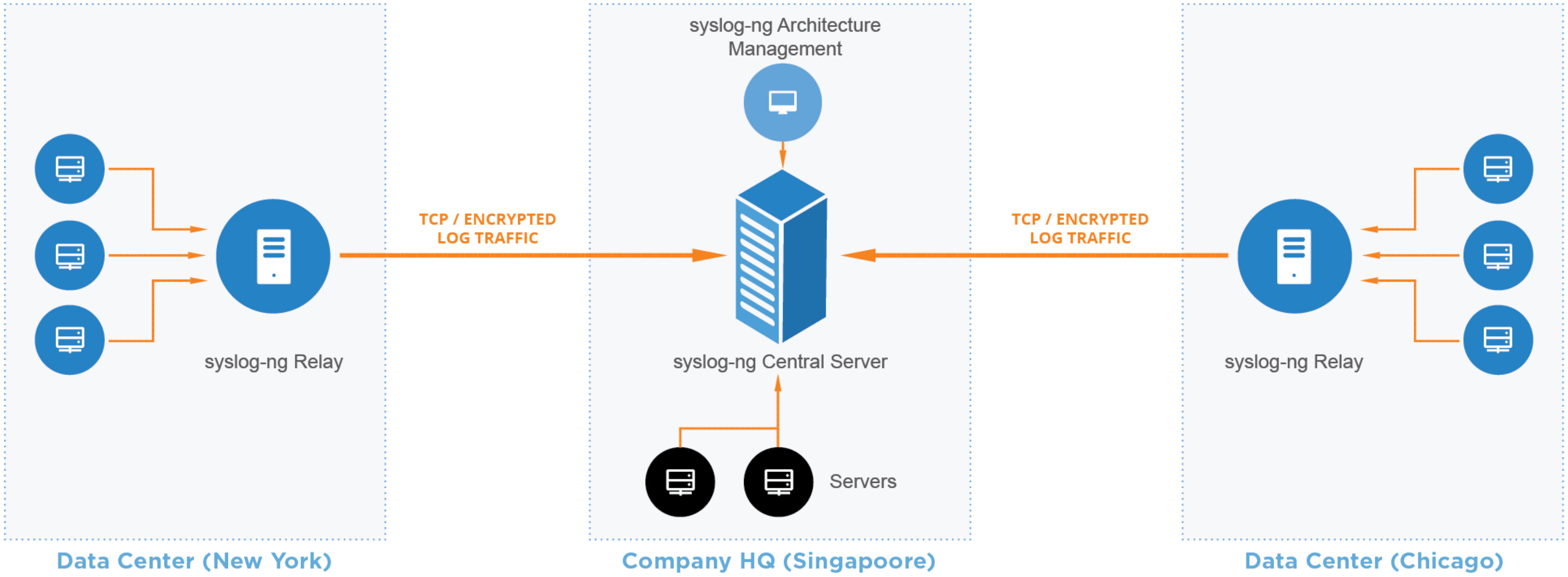
  app=sshd user=root source_ip=192.168.123.45

- 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

SCALING SYSLOG-NG

- Client – Relay – Server instead of Client – Server
- Distribute some of the processing to Client/Relay



BALABIT

LOG ROUTING

- Based on filtering
- Send the right logs to the right places
- Message parsing can increase accuracy
  - E-mail on root logins

- Can optimize SIEM / log analyzer tools
  - Only relevant messages: cheaper licensing
  - Throttling: evening out peaks

WHAT IS NEW IN SYSLOG-NG 3.8

- Disk-based buffering
- Grouping-by(): correlation independent of patterndb
- Parsers written in Rust
- Elasticsearch 2.x support
- Curl (HTTP) destination
- Performance improvements
- Many more :-)

SYSLOG-NG BENEFITS FOR LARGE ENVIRONMENTS

High-performance reliable log collection

Simplified architecture

Single application for both syslog and application data

Easier-to-use data

Parsed and presented in a ready-to-use format

Lower load on destinations

Efficient message filtering and routing

JOINING THE COMMUNITY

- syslog-ng: http://syslog-ng.org/
- Source on GitHub: https://github.com/balabit/syslog-ng
- Mailing list: https://lists.balabit.hu/pipermail/syslog-ng/
- IRC: #syslog-ng on freenode

QUESTIONS?

My blog: https://www.balabit.com/blog/author/peterczanik/

My e-mail: peter.czanik@balabit.com

Twitter: https://twitter.com/PCzanik

SAMPLE XML

```
•   <?xml version='1.0' encoding='UTF-8'?>
•   <patterndb version='3' pub_date='2010-07-13'>
•     <ruleset name='opensshd' id='2448293e-6d1c-412c-a418-a80025639511'>
•       <pattern>sshd</pattern>
•       <rules>
•         <rule provider="patterndb" id="4dd5a329-da83-4876-a431-ddcb59c2858c" class="system">
•           <patterns>
•             <pattern>Accepted @ESTRING:usracct.authmethod: @for @ESTRING:usracct.username: @from @ESTRING:usracct.device: @port @ESTRING:: @@ANYSTRING:usracct.service@</pattern>
•           </patterns>
•           <examples>
•             <example>
•              <test_message program="sshd">Accepted password for bazsi from 127.0.0.1 port 48650 ssh2</test_message>
•              <test_values>
•               <test_value name="usracct.username">bazsi</test_value>
•               <test_value name="usracct.authmethod">password</test_value>
•               <test_value name="usracct.device">127.0.0.1</test_value>
•               <test_value name="usracct.service">ssh2</test_value>
•              </test_values>
•             </example>
•           </examples>
•           <values>
•             <value name="usracct.type">login</value>
•             <value name="usracct.sessionid">$PID</value>
•             <value name="usracct.application">$PROGRAM</value>
•             <value name="secevt.verdict">ACCEPT</value>
•           </values>
•       </rule>
```