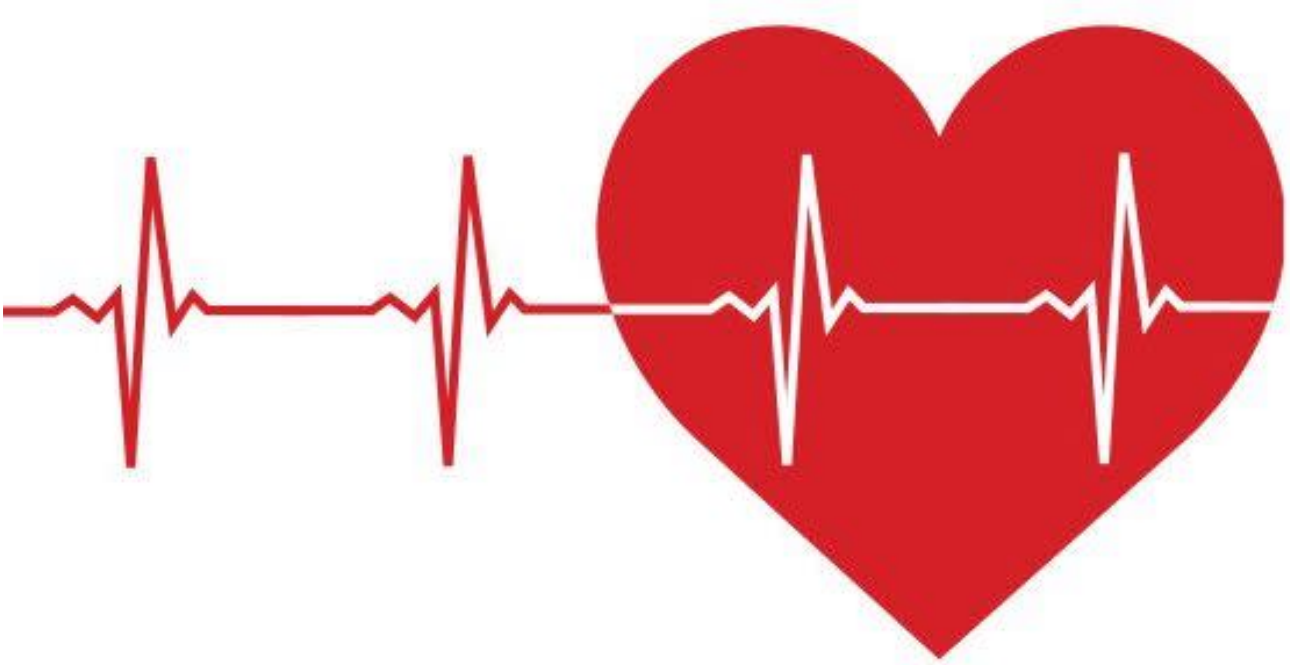
Log ingestion with Beats, Logstash and Search Guard

Paulo Melo · 2019-03-20



Elasticsearch and its ecosystem provide a great stack for ingesting, storing and searching logs. Search Guard adds important safeguards to protect log data over its lifetime. It can be also easily integrated with all tools from the log analytics ecosystem. In this article we present a demo of Search Guard & Beats & logstash integration.

Beats

Beats is a *"lightweight data shipper"* that serves as an agent on a server where logs are produced. Ingesting logs with Beats into Elasticsearch can be done in two ways:

1. We can define beats output to store data in Elasticsearch, which means we transport data directly from Beats agent to a cluster.
2. We can export logs from Beats into Logstash, and then into Elasticsearch.

Both pipelines are supported by Search Guard and we start with the first approach. We will install filebeat and configure a log input from a local file.

We install a fresh demo version of Elasticsearch and Kibana, both with Search Guard plugins enabled. The default demo configuration already contains a user *logstash* (with a password *logstash*), and a *sg_logstash* role assigned. The user has been granted permissions on indices *logstash-** and *beat*. We do not need to apply any changes on Elasticsearch cluster.

We add the following entries into filebeat.yml that will fetch log lines from a local file */var/log/searchguard_test.log*:

```
filebeat.inputs:

- type: log
  enabled: true
  paths:
    - /var/log/searchguard_test.log
  json.keys_under_root: true
  json.add_error_key: true
  json.message_key: log
```

We also define an output to Elasticsearch over *HTTPS* protocol with user and password provided:

```
setup.template.name: "filebeat"
setup.template.pattern: "filebeat-*"

output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["localhost:9200"]
  index: "beats-%{+yyyy.MM.dd}"
  protocol: "https"
  username: "logstash"
  password: "logstash"
```

As a demo, we can echo some JSON log into searchguard_test.log:

```
echo "{ \"level\": \"info\", \"message\": \"SearchGuard beat log sample\"}" >>
/var/log/searchguard_test.log
```

After a moment, the log will be available in Elasticsearch and Kibana. We can use a demo user *kibanaro* (password: *kibanaro*) to create an index pattern and browse incoming logs:

Logstash

A downside of our previous approach is that in order to do modify our log pipeline, we need to modify the configuration on all Beats servers. Additionally, all the Beats servers store username and password to access Elasticsearch which can lead to security concerns.

That is why it is so popular to use Beats & logstash together:

- Use beats to capture logs on servers and pass them into logstash.
- Use logstash to output the logs into Elasticsearch.

We will now modify our previous example to make it work this way. We modify *filebeat.yml*:

- Remove ElasticSearch output from previous sections.
- Add logstash output of the form:

```
output.logstash:
  hosts: ["localhost:5044"]
```

We download logstash and create or modify sample pipeline. We can alter *logstash-sample.conf* like:

```
input {
  beats {
    port => 5044
  }
}

output {
  elasticsearch {
    hosts => ["https://localhost:9200"]
    index => "logstash-%{[@metadata][version]}-%{+YYYY.MM.dd}"
    user => "logstash"
    password => "logstash"
    ssl => true
    cacert => "path-to-root-CA/config/root-ca.pem"
    ssl_certificate_verification => true
  }
}
```

We specify the root-ca certificate that has been generated for the demo configuration. The file should be located within config directory of Elasticsearch.

We start logstash with a given pipeline definition:

```
/bin/logstash -f logstash-sample.conf
```

We can now fire some logs:

```
echo "{ \"level\": \"info\", \"message\": \"SearchGuard beats & logstash log sample\"}" >>
/var/log/searchguard_test.log
```

and check if they appear in Kibana. Note that we write into different indices and a separate index pattern needs to be created.

t	level	⊕ ⊖ ◫ ✱	info
t	log.file.path	⊕ ⊖ ◫ ✱	/var/log/searchguard_test.log
t	message	⊕ ⊖ ◫ ✱	SearchGuard beats & logstash log sample

Summary

In this post we have described two possible log ingestion setups:

- Beats → ElasticSearch
- Beats → Logstash → Elasticsearch

We explained differences between the approaches and we have shown that Search Guard is ready for both of them. Feel free to choose the one that suits your needs best.

Where to go next

- Read more about logstash configuration in our documentation
- As an alternative to logstash, learn how to use fluentd with Search Guard

Image: shutterstock / memej

This article was published on the Search Guard blog: https://search-guard.com/blog/beats-logstash-searchguard/
Search Guard is Security and Alerting for Elasticsearch and Kibana — search-guard.com · Start a free trial