SEARCH GUARD

AUDIT LOGGING

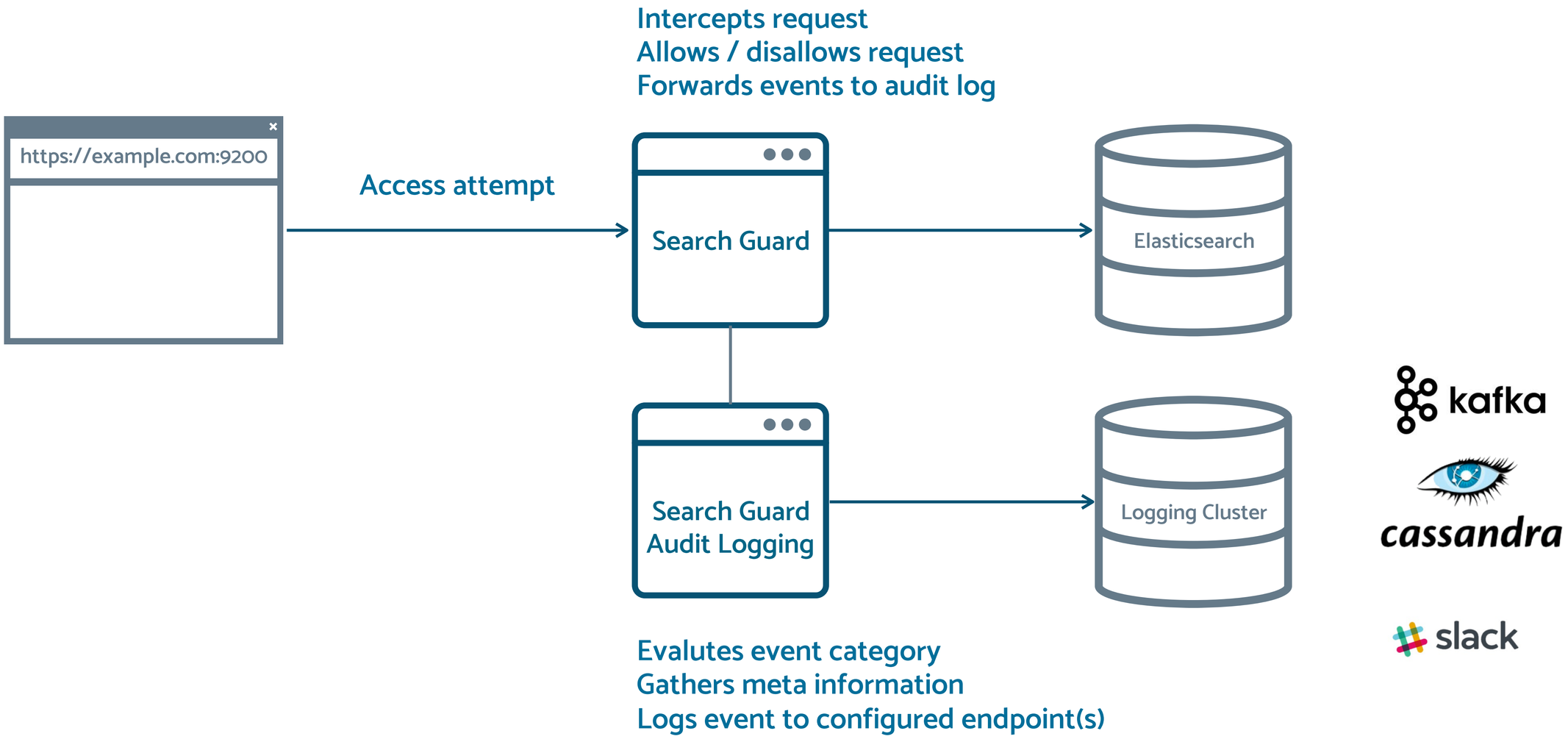
Search Guard

01.

WHAT IS AUDITLOGGING

- **Track irregular, security-related events, e.g.**
  - → failed log in attempts
  - → missing privileges
  - → spoofed HTTP headers
- **Track regular events**
  - → Login events
  - → Executed actions like searches or aggregations
- **Can be logged to different output channels**
  - → Elasticsearch, Kafka, Cassandra, Webhooks, etc.

EXAMPLE



02.

EVENT CATEGORIES

- **FAILED_LOGIN**

→ The user credentials of a request could not be validated.

→ Most likely because the user does not exist or the password is incorrect

- **AUTHENTICATED**

→ The provided user credentials were authenticated successfully

- **MISSING_PRIVILEGES**

→ The user is authenticated but lacks the respective privileges for the requested action

- **GRANTED_PRIVILEGES**

→ Represents a successfully executed action, e.g., a search request

03.

EVENT CATEGORIES

- **SSL_EXCEPTION**

→ An attempt was made to access Elasticsearch without a valid SSL/TLS certificate.

- **BAD_HEADERS**

→ An attempt was made to spoof a request with Search Guard internal headers

- **SG_INDEX_ATTEMPT**

→ An attempt was made to access the Search Guard internal user- and privileges index without a valid admin TLS certificate

04.

EVENT CONTENT

- **Category**

→ FAILED_LOGIN, MISSING_PRIVILEGS

- **Username**

→ if available for the logged category

- **Metadata**

→ Timestamp, remote IP, node name, cluster name, layer (REST/transport), etc.

- **Request type**

→ E.g. SearchRequest, GetIndexRequest, GetMappingsRequest ...

05.

EXTENDED LOGGING

- **Indices affected by the request**
  - → Enable by: searchguard.audit.resolve_indices: true
  - → Indices need to be resolved before the event is logged
  - → Slight performance overhead
- **Request body**
  - → E.g., executed query, indexed document
  - → Enable by: searchguard.audit.log_request_body: true
  - → Increases event size

EXAMPLE

audit_cluster_name: ”searchguard_demo”,

...

audit_transport_request_type: ”SearchRequest”,

audit_category: ”MISSING_PRIVILEGES”,

audit_request_origin: ”REST”,

...

audit_request_body: ”{”query”:{”match”:{”Designation”:

{”auto_generate_synonyms_phrase_query”:true,”query”:”CEO”,”zero_terms_query”:”NONE”,”fuzzy_transpositions

”:true,”boost”:1.0,”prefix_length”:0,”operator”:”OR”,”lenient”:false,”max_expansions”:50}}}}”,

...

audit_request_layer: ”TRANSPORT”,

audit_request_effective_user: “jdoe”,

audit_trace_resolved_indices: [

”humanresources”

]

06.

EVENT STORAGE

- **Log events are stored asynchronously**

→ Minimizes performance impact

→ Thread pool can be optimized

- **Log events can come from any node**

→ Only centralized storage is useful

- **Search Guard ships with pre-defined storage endpoints**

→ Own implementations also possible

07.

PRE-DEFINED STORAGE ENDPOINTS

- **Events can be shipped to one or more storage endpoints**
- **Internal Elasticsearch**

→ Stores events on the same cluster they have been generated on

- **External Elasticsearch**

→ Stores events on a remote Elasticsearch cluster

- **Webhooks**

→ Any system that supports GET or POST webhooks

- **log4j**

→ Kafka, Cassandra, SNMP, Mail, etc.

08.

DATA CONSISTENCY

- **Security events must be tamper-proof**

→ Security events must not be changed once they are written

- **Immutable indices**

→ Any index can be marked “immutable”

→ “write-once / read many”

→ Documents can be indexed, but not changed afterward

→ Immutable indices cannot be deleted or closed

→ Ideal for storing audit events

09.

CONFIGURATION

- **Audit logging is configured statically, not dynamically**

→ in elasticsearch.yml

- **Due to security considerations**

→ Audit logging must not be disabled by any unauthorized user

→ Configuration must not be changed by any unauthorized user

→ Logged events and data must be predictable and consistent over time

- **Any change to the audit logging configuration requires a cluster restart**

EXAMPLE

Internal:

searchguard.audit.type: internal_elasticsearch

searchguard.audit.config.index: auditlog

External:

searchguard.audit.type: external_elasticsearch

searchguard.audit.config.http_endpoints: [‘es1.example.com’,’es2.example.com’]”

searchguard.audit.config.index: auditlog

searchguard.audit.config.username: auditloguser

searchguard.audit.config.password: auditlogpassword

searchguard.audit.config.enable_ssl: true

Webhooks:

searchguard.audit.config.webhook.url: “https://siem.example.com/ingest”

searchguard.audit.config.webhook.format: JSON

10.

ADDITIONAL RESOURCES

- **Configuring Search Guard audit logging**

→ https://docs.search-guard.com/latest/audit-logging-compliance

- **Configuring storage endpoints**

→ https://docs.search-guard.com/latest/audit-logging-storage

- **Audit events field reference**

→ https://docs.search-guard.com/latest/audit-logging-reference

SEARCH GUARD

SEND US A MESSAGE:

info@search-guard.com

Search Guard

floragunn GmbH
Tempelhofer Ufer 16
D-10963 Berlin, Germany

E-Mail: info@search-guard.com
Web: search-guard.com

Managing Directors: Claudia Kressin, Jochen Kressin
Registergericht: Amtsgericht Charlottenburg
Registernummer: HRB 147010 B
E-Mail: info@floragunn.com