How Immutable Indices help you to stay GDPR compliant

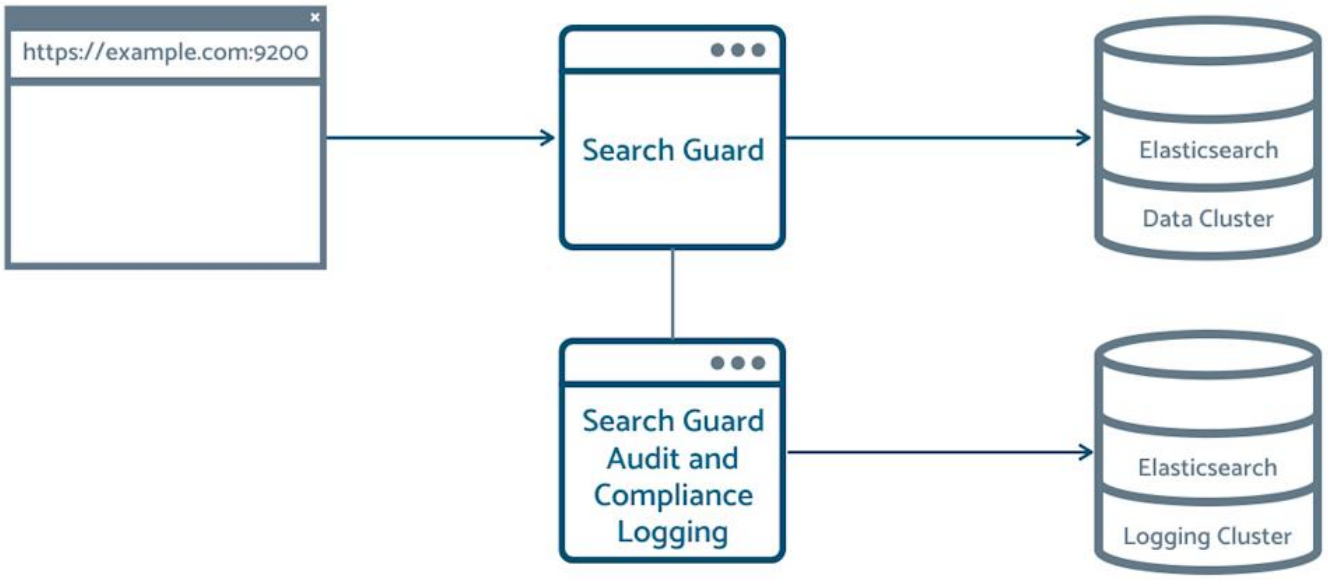
Jochen Kressin · 2018-04-16

Search Guard is more than capable of monitoring and recording security and compliance related events. You can, for example, track the complete lifecycle of a document, including creation, changes, and deletion. Search Guard can ship these compliance events to multiple endpoints, like SIEM systems, Kafka, Cassandra and, of course, Elasticsearch. To ensure the integrity of this compliance audit trail, Search Guard provides you with the new Immutable Indices feature: Once data is written to an immutable index, any changes to it is automatically prevented. This makes sure that no one tampers with the event data once recorded. Stay calm on your next compliance audit!

Both the Audit Logging module and the new Read- and Write History can produce a lot of valuable security and compliance relevant events. By using both, you are on a good way to meet most compliance regulations, like GDPR, PCI, SOX or ISO.

Using a dedicated Elasticsearch logging cluster

Generating an audit trail is fine, but the logical next question is where to store these events. It's good practice to store them in an external system, like SIEMonster, Kafka, Cassandra or the like. But why introduce another system when you already have the perfect solution in place - Elasticsearch. In fact, many users set up a dedicated Elasticsearch logging cluster which stores all security and compliance events.



The generated events are very valuable, especially in case of a data breach, a compliance audit or whenever a customer exercises his or her information rights under GDPR. So you have to make sure no one can change the events after they have been recorded.

> How do you allow a user to write data, but prevent any changes once the data has been written?

What options do you have? Of course, you can protect the event index with Search Guard's powerful role-based access control features, and maybe supplement that with Document- and Field-Level security: Set up one or more roles, and grant only read access to the event index. But you will run into a chicken-and-egg problem: In order to write audit events to Elasticsearch, you need at least one user/role with write access to the index. But this account can potentially also be used to change any existing data in that index. So how do you allow a user to write data, but prohibit any changes once the data has been written?

Introducing Immutable Indices

The Search Guard Immutable Indices feature comes to the rescue. Immutable indices, sometimes also called *write-once indices* allow the creation of new documents, but once created, they cannot be changed anymore. It's like burning a non-rewritable CD or DVD back in the days: You can burn them once and read them as many times you want. But once the data is recorded, changes are not possible.

> Immutable Indices allow the creation of new documents, but once created, prevent any changes to them.

Making an index immutable is very simple. You only need to list all indices you would like to mark immutable in *elasticsearch.yml* and you are done:

```
searchguard.compliance.immutable_indices:
 - auditlog
```

Don't forget to restart the node(s) for the changes to take effect!

We can now create a Search Guard role that has *CRUD* access to this index in *sg_roles.yml*, and map a user *auditlog* to the role in *sgrolesmapping.yml*:

```
sg_roles.yml
sg_auditlog:
  indices:
    'auditlog':
      '*':
        - CRUD
```

sg*roles*mapping.yml

```
sg_auditlog:
  users:
    - auditlog
```

Now we create a new document in the immutable *auditlog* index:

```
curl -u auditlog:password \
  -H 'Content-Type: application/json' \
  -XPUT "https://example.com:9200/auditlog/_doc/1" \
  -d '{"field1": "value1", "field2":"value2"}'
```

As expected, the call succeeds and the document is created. Let's now try to change it:

```
curl -u auditlog:password \
  -H 'Content-Type: application/json' \
  -XPOST "https://example.com:9200/auditlog/_doc/1/_update?pretty" \
  -d '{"doc":{"field1": "newvalue1", "field2":"newvalue2"}}'
```

As per role definition you would expect that this call also succeeds. We have write permissions, right? This is where the immutable index protection kicks in an prevents the changes from being executed:

```
{
 "error": {
   "root_cause": [{
     "type": "security_exception",
     "reason": "Index is immutable"
   }],
   "type": "security_exception",
   "reason": "Index is immutable"
 },
 "status": 403
}
```

And we get the same result if we try to delete the document:

```
curl -u auditlog:password \
 -XDELETE "https://example.com:9200/auditlog/_doc/1"
```

This of course applies to any Elasticsearch action that would change data, including bulk requests or delete by query for example.

By making the index immutable we are now protected from any changes to the data in it. Not quite. There are other things an attacker or an inattentive employee can do:

- Deleting the index
- Restoring the index from a snapshot
- Creating an index alias and trying to change data via the alias name
- Closing and opening the index
- Reindexing

Search Guard has your back also covered in these situations. Once an index is marked immutable, none of the actions mentioned above are possible anymore. Any attempt will result in a security exception.

What about maintenance?

Merely forbidding any changes to an immutable index is not practical. Your index would grow forever, without any chance for you to do anything about it. How can you maintain the index then?

> Index maintenance for immutable indices is only possible with an admin TLS certificate

Search Guard never had any default users or passwords, and that means there is no default root or admin user. Instead, we rely on the industry standard TLS to give a user elevated privileges to the cluster: To perform any sensitive operation, like changing roles and permissions, you need to provide an admin TLS certificate to Search Guard. Maintaining immutable indices are no exception to this rule: Search Guard protects immutable indices against changes performed by any regular role, but will allow index maintenance if you provide an admin TLS certificate.

Summary

Immutable Indices, introduced with the Compliance Edition of Search Guard, solve the chicken and egg problem when storing audit and compliance events in Elasticsearch. Once data has been written to an immutable index, it cannot be changed anymore, even if a Search Guard role has write access for this index. This ensures integrity of your data and makes sure that audit and compliance events are unchangeable once written. In addition, your index is protected from any operation that changes or copies it, like deletion, reindex or restore. Immutable indices are a perfect feature to meet compliance regulations like GDPR, SOX, PCI or ISO.

Where to next

- Download the Search Guard Compliance Edition and try immutable indices for yourself
- Learn about the Search Guard Compliance features, including Immutable Indices

- Help us in making Search Guard even better by taking part in our compliance survey

Image: shutterstock / gotphotos