Security for Elasticsearch and OpenSearch: Better Than Ever

Nils Bandener · 2022-03-08

In the past few months, we have been busy working on the next generation of Search Guard: Search Guard FLX. Right now, we are very close to the actual general availability release. As a preparation, we released a third "technology preview" of Search Guard FLX.

What Does This Mean?

The technology preview is meant to give you an impression of how Search Guard FLX will work. Also, we want to give you the chance to check it out and gather some experience. If you notice something which you think should work differently, please let us know! Right now is the best chance to get these things fixed and create a future-proof product.

However this also means that the tech preview is not meant for production, yet. There might be some bugs in it. Also, there might be some more breaking changes coming up.

The general release version is expected very soon, either end of March or early April.

Overview

Search Guard FLX brings big improvements in terms of both flexibility and user orientation. Highlights amongst the changes are:

Authentication / Authorization

Completely new approach for configuring authentication. The new configuration format is more coherent, more predictable and much more powerful. Simple setups require very little configuration; very complex

setups are possible by straight-forward configuration. If something goes wrong, Search Guard FLX provides extensive error messages and diagnostic information. This diagnostic information is not hidden away in logs, but is made easily accessible by special debug modes.

Administration

New administration tool sgctl which shall replace sgadmin. sgctl is stateful; that means, you can define connection profiles once and use these later. Thus, you don't have to specify all connection configuration on each invocation. The interface of sgctl is more streamlined and offers you improved configuration validation functionality.

Kibana / Dashboards

Completely new approach for logging into Dashboards/Kibana: The configuration can now also be changed without having to restart Kibana. Also, it is now possible to use several authentication modes at once. Thus, you can choose, for example, between password-based login and OIDC login while logging in. Finally, logged in users now get an actual server-side session. This fixes a number of issues Kibana login had so far.

Kibana / Dashboards

Improved way of handling unauthorized indices. You no longer have to worry that your wildcard query breaks because it might pick up an index you don't have permissions for.

Clear, Concise and Consistent: The New Configuration Format

Search Guard FLX brings a new configuration format for authentication: **sg_authc.yml**. It requires less boilerplate and clutter, while being more powerful and predictable.

In older versions of Search Guard, a basic configuration for username/password based authentication combined with JWT looked like this:

```
sg_config:
  dynamic:
      authc:
        basic_internal_auth_domain:
          http_enabled: true
          transport_enabled: false
          order: 4
          http_authenticator:
            type: basic
            challenge: true
          authentication_backend:
            type: intern
        jwt_auth_domain:
          http_enabled: true
          transport_enabled: false
          order: 0
          http_authenticator:
            type: jwt
            challenge: false
            config:
              signing_key:
"MIADuDCCAqCgAwIBAgIBATANBgkqhkiG9w0BAQsFADMS4wMSUwIwYDVQQLDBxFeGFtcGxlIERTRyBJbmMuIDEuMCBBtMR0wGw
YDVQ...."
              roles_key: "roles"
          authentication_backend:
            type: noop
```

Now it looks like this:

```
auth_domains:
- type: basic/internal_users_db
- type: jwt
  jwt.signing.rsa.certificate:
"MIADuDCCAqCgAwIBAgIBATANBgkqhkiG9w0BAQsFADMS4wMSUwIwYDVQQLDBxFeGFtcGxlIERTRyBJbmMuIDEuMCBBtMR0wGw
YDVQ...."
  user_mapping.roles.from_comma_separated_string: jwt.roles
```

Why is it so much simpler?

The *order* attribute is no longer used to define the order in which the authentication domains are evaluated. Now, it is simply the inherent order of the array. This makes it absolutely clear what the evaluation order is. You no longer have to wonder wether the auth domains are evaluated in ascending or descending order.

The *challenge* attribute is no longer necessary. Search Guard FLX is now capable to combine challenges of several auth domains when needed.

Authentication for the transport layer was always a niche topic. Most users don't need it. Thus, they should not be required to bother with this. Thus, for the rare case that transport layer authentication is necessary, the configuaration for this is done in a separate file.

Default values and config structures have been adapted to avoid lots of boiler-plate, which is not relevant for a human user.

Additionally, Search Guard FLX now offers first-class configuration variables. These allow you to easily manage and update secrets and other values which should be decoupled from the configuration.

This means, you can use the configuration variables to manage the JWT signing key. To upload the key, just use this command:

```
$ ./sgctl.sh add-var jwt_key -i /path/to/jwt_key.pem --encrypt
```

This will upload the file at */path/to/jwt_key.pem* to Search Guard FLX, which makes its content available as configuration variable. Then you can change your configuration to this:

```
auth_domains:
- type: basic/internal_users_db
- type: jwt
  jwt.signing.rsa.certificate: "#{var:jwt_key}"
  user_mapping.roles.from_comma_separated_string: jwt.roles
```

If you ever want to rotate your JWT key, you no longer need to modify the authentication configuration itself. Just do this:

```
$ ./sgctl.sh update-var jwt_key -i /path/to/jwt_key.pem --encrypt
```

This example only showed the configuration for JWT and user/password based authentication. However, also the configuration for LDAP, for anonymous auth, for proxy auth, etc. has been greatly improved. Just check out the documentation for more on these topics!

Take control with sgctl

Search Guard FLX also comes with a new administration command line tool, which allows you to modify the configuration and perform further administrative tasks: **sgctl**.

One of the greatest advantages of sgctl over sgadmin is that sgctl is able to remember connection settings. You have to create a connection profile only once initially with *sgctl connect*. This will store the connection profile in a local configuration file on your computer. Afterwards, sgctl will automatically pick up these settings for all other commands.

So, updating your Search Guard FLX configuration is now possible with just this command line:

```
./sgctl.sh update-config sg_authc.yml
```

That's easy to remember, right?

Also, if you make a mistake, *sgctl* brings you rich validation error messages which make errors easy to spot and understand. You no longer have to scroll through log files to identify the cause of an error:

```
./sgctl.sh update-config sg_roles_mapping.yml sg_authc.yml

Invalid config files:

sg_roles_mapping.yml:
  SGS_READALL.ips.0:
      Invalid value; expected: IP address or netmask in CIDR notation; got: abc
  SGS_READALL.search_uard_roles:
      Unsupported attribute

sg_authc.yml:
  auth_domains.0.enabled:
      Invalid value; expected: true or false; got: alse
  auth_domains.1.type:
      Unknown authentication frontend
```

sgctl also allows you to modify configuration without having to edit YAML files. Just use *sgctl set*. This switches on the authentication debug mode by setting the debug flag in sg_authc to true:

```
./sgctl.sh set authc debug --true
```

Also cool: sgctl is very lightweight. It comes as a self-contained, executable sh file. You just need to have a running cluster and a Java runtime enviroment installed. Then you are ready to go without further installation. You can use the same command both for Elasticsearch and for OpenSearch clusters protected by Search Guard FLX.

And that's not all; sgctl offers a number of very useful administration commands. Check out the docs.

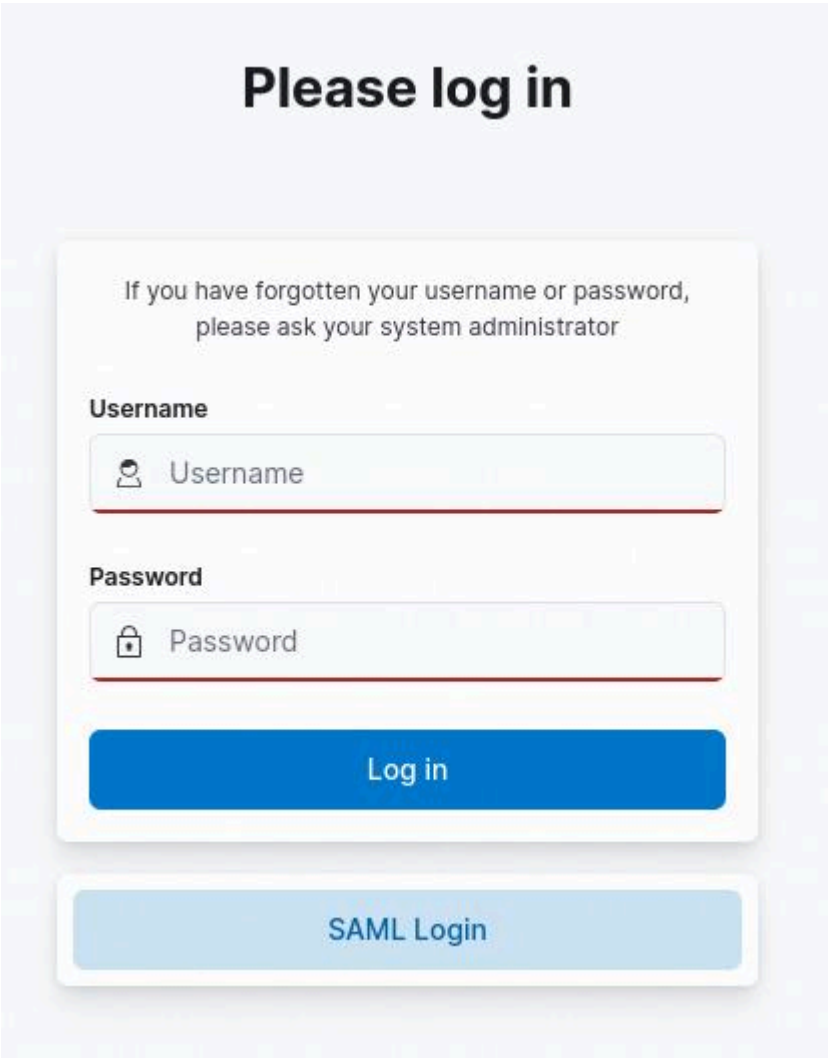
Authentication as you Need it for Kibana and OpenSearch Dashboards

The login mechanism for Kibana and OpenSearch Dashboards has also been greatly improved. Users now get an actual server-side session; This fixes a number of issues, such as:

- Issues with cookies exceeding the browser size limit.
- The “logout” menu item is able to invalidate the session. Thus, session cookies cannot be re-used any more.

Additionally, configuration of SSO using OIDC or SAML for Dashboards/Kibana no longer interfers with backend authentication configuration. Thus, you can now have challenging basic authentication on the backend while using OIDC or SAML for Dashboards/Kibana.

You can even use several authentication modes at the same time. The login screen then lets you choose the authentication mode you want:



Finally, Dashboards/Kibana authentication configuration can now be changed without having to restart the node.

To Wrap it up

This was only a very brief and high-level walk through of a number of cool new things which Search Guard FLX will bring.

Even though it is close to being finished, it is still a work-in-progress at the moment. Some pieces may be odd or not as you might need them. So, in order to make things perfectly suitable *for you*, please go ahead, grab the tech preview, read the docs and check out all the new stuff. If you notice anything which does not work as expected or which seems to be weird, do not hesitate to post in the forums or to file an issue in the main repo or the sgctl repo.

Now is the time to check it out!

This article was published on the Search Guard blog: https://search-guard.com/blog/security-for-elasticsearch-opensearch-better-than-ever/