What's new in Search Guard 6: The configuration GUI

Jochen Kressin · 2018-01-22

Search Guard always had a compelling configuration mechanism: The configuration is stored in an Elasticsearch index, is hot-reloadable without any cluster downtime and secured by an admin TLS certificate. For accessing and changing configuration settings the *sgadmin* command line tool or the REST API were the tools of choice. Search Guard 6 now supplements this with an easy-to-use, Kibana based graphical configuration interface.

Role-based REST API access

The basis for the configuration GUI is the Search Guard REST management API. This API provides endpoints for creating, updating and deleting users, roles, mappings and action groups. To use the API with Search Guard 5 and below, you needed to send an admin TLS certificate in every request. Starting with Search Guard 6, you can now also configure role-based access to the API which was precondition for implementing the configuration GUI.

In *elasticsearch.yml*, configure all roles that should have access to the REST API like:

```
searchguard.restapi.roles_enabled: ["sg_all_access", ...]
```

In our example, we use the *sg*all*access* demo role, which the *admin* demo user maps to. But you can use any of your own roles here, changes to the actual role definition are not required. Simply list the roles, and you're good to go.

Note that you can also configure more fine-grained access control to the API, individually for each role. You can give a role the permissions to view internal users and create new users, but deny the permission to delete users. And you can also disable endpoints and methods globally. The configuration GUI has an auto-configuration feature which adapts the available functions to the user's permissions. In other words, if a user does not have permission to access the *ROLES* REST endpoint, the corresponding navigation entry in the GUI will not be displayed.

Configuring the role-based REST access is actually the only thing you need to do on Elasticsearch side, so we can directly head over to Kibana.

Installing the Kibana plugin

The configuration GUI is part of the Search Guard Kibana plugin. So first we download and install the Search Guard plugin that matches our Kibana version. Download the plugin, and install it with the *kibana-plugin* install command:

```
bin/kibana-plugin install file:///path/to/search-guard-kibana-plugin-<version>.zip
```

Then just follow the usual configuration steps and set up the Elasticsearch URL, the Kibana server user, the TLS verification mode and eventually also the HTTP header whitelist.

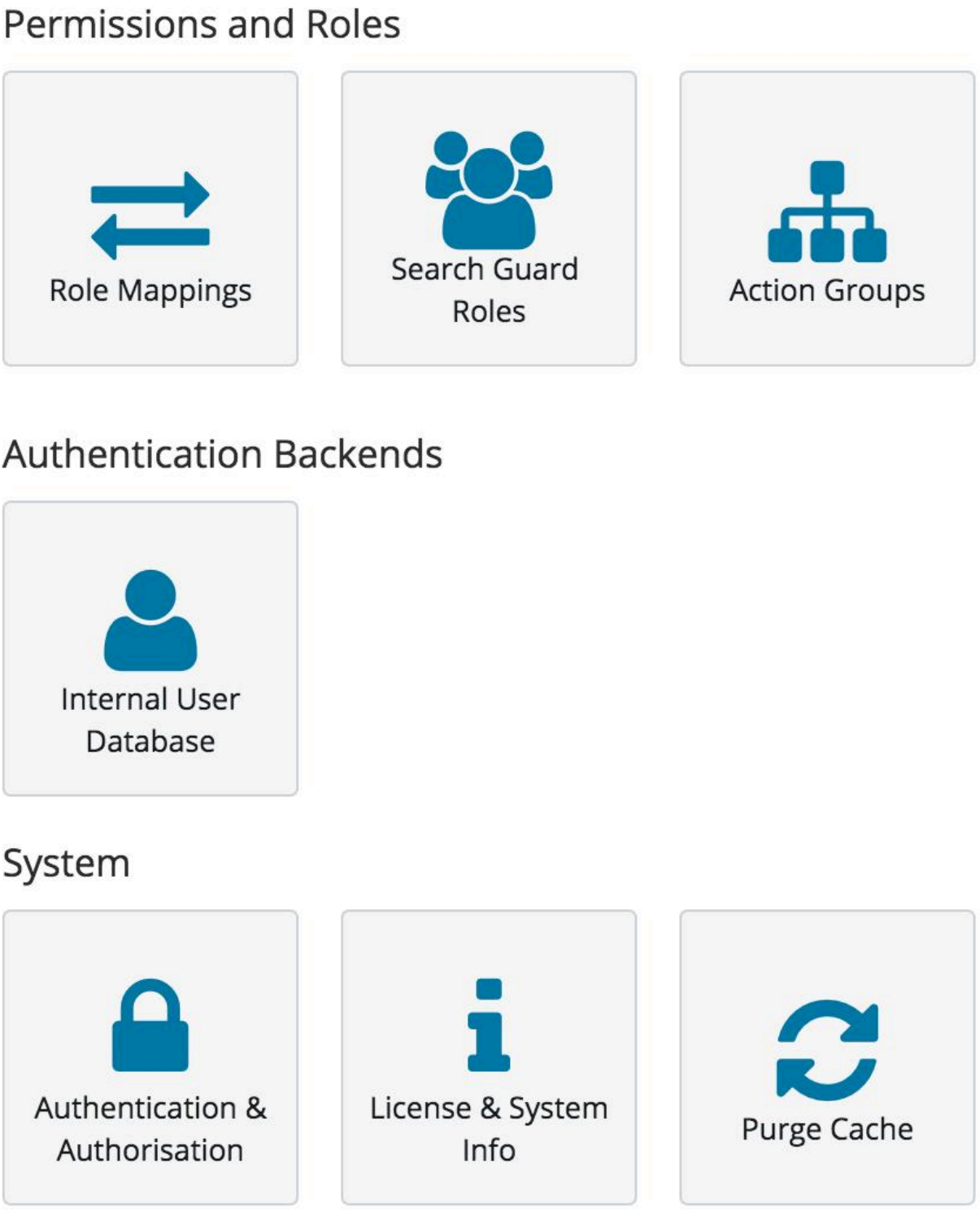
Configuring the GUI

Now we can add the relevant configuration keys for the GUI to *kibana.yml*. Wait … did we say auto-configuration? Yes, and in fact there is nothing more to configure in order to use the GUI. It automatically determines whether the REST API is installed, if the user has access to it, if there are any endpoints disabled and then adapts the user interface automatically. So, just fire up a browser, and log in with a user for which you configured REST API access in elasticsearch.yml.

Hint: If you use the Search Guard demo installation script, this configuration entry is added to elasticsearch.yml automatically. Just use the admin/admin demo user to log in to Kibana.

Main navigation entries

Once you logged in, you should see a new navigation entry in the left sidebar called "Search Guard". Clicking on it will take you directly to the main navigation points of the config GUI:



- Role Mappings
  - Configure how Search Guard will map users, backend roles and hosts to Search Guard roles
- Search Guard Roles
  - Configure Search Guard roles and define access permissions to indices, documents, and fields
- Action Groups

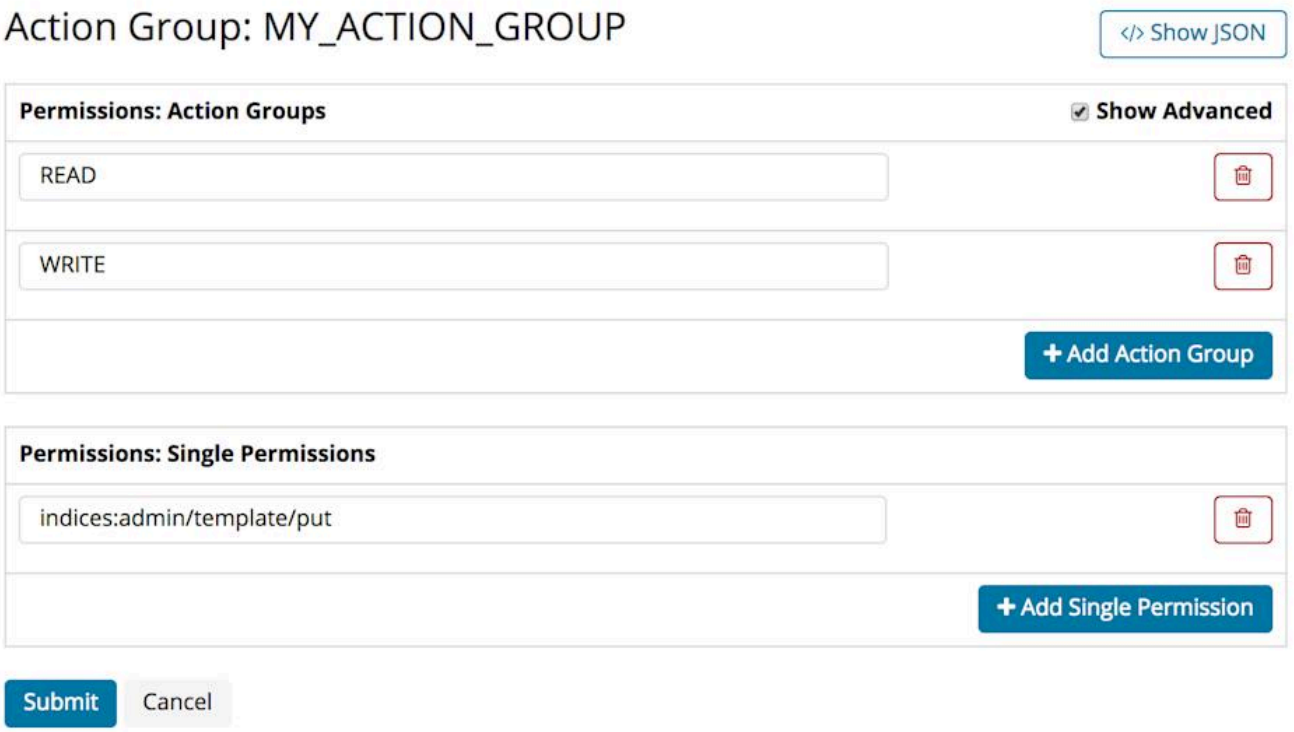
    - Configure action groups that group Elasticsearch action permissions under a telling name like READ, WRITE, CRUD etc.
- Internal User Database
    - If you don't use an external authentication system like Active Directory, LDAP, JWT or Kerberos you can also store users directly in the Search Guard internal user database.
- Authentication & Authorization
    - Display the currently active authentication and authorization configuration
- License & System Info
    - Display information about the currently active modules, the Search Guard license, and upload a new license at run-time.
- Purge Cache
    - Purge any Search Guard internal cache

Clicking on one of *Role Mappings, Search Guard Roles, Action Groups* or *Internal User Database* will first take you to an overview screen.

You can mark individual resources as *reserved*, which means they are read-only and cannot be edited. You might want to mark service users for Kibana and logstash as *reserved*, protecting them from accidental changes.

All overview screens provide a filter bar for searching for a particular resource name, an add button to create a new resource, and buttons for editing, cloning and deleting resources:

Clicking on a resource name or the edit button will then take you to the edit screen where you can change all aspects of the selected resource. For example, editing an Action Group looks like:



Once you save the resource, here: an Action Group, Search Guard will store the changes via the REST API, and the changes will take effect immediately. We also made sure to help you with all configuration tasks, for example by providing auto-complete for most of the available input fields, or by making it possible to validate a Document-Level security query directly against Elasticsearch.

Most of the screens are self-explanatory, but we will also publish a follow-up to this article explaining role configuration in detail.

Summary

In Search Guard 2 and 5 you either had to use sgadmin or the REST management API to change users, roles, mappings, and permissions. Search Guard 6 introduces a new Kibana-based configuration GUI which makes it really easy to configure all aspects of Search Guard. Access to the GUI is granted on a per-role basis. It has an auto-configuration feature, so there are no additional entries in *kibana.yml* necessary to configure in order to use the GUI.

This article was published on the Search Guard blog: https://search-guard.com/blog/configuration-gui/