Helpful Hint: Assign roles based on IPs or hostnames

Jochen Kressin · 2020-11-19

One of the lesser-known features of Search Guard is the ability to use the hostname or IP address to map roles to a request. The most common use case is to allow full access to all Elasticsearch data if the request was sent directly from a node in the cluster.

Mapping users to roles

In a typical Elasticsearch / Search Guard setup, you create a couple of roles that define access permissions and then map users to these roles.

The two most common attributes to assign roles to users are

- Username
- Backend roles, for example, LDAP groups or JWT claims

A typical role mapping might look like:

```
sg_read_write:
  users:
    - janedoe
    - johndoe
  backend_roles:
    - 'cn=ldaprole,ou=groups,dc=example,dc=com'
```

This definition assigns the role *sg_read_write* to users with username *janedoe* or *johndoe*, and to users that belong to the LDAP group *cn=ldaprole,ou=groups,dc=example,dc=com*.

The role *sg_read_write* defines what permissions these users have when accessing your Elasticsearch cluster.

Using hostnames and IPs for role mappings

In addition to *username* and *backend_roles*, you can also use the callers hostname or IP address to assign roles.

When is that useful? In essence, this feature allows you to assign roles dynamically, depending on where the request comes from.

For example, you may want to grant a user only a limited set of permissions if the cluster is accessed from outside your network. When accessed from inside, the user should have broader permissions.

Or, and this is probably the most prominent use case, grant a user full access to all indices and data if the request comes from a node in your cluster itself. While users should have only limited access, you want to grant DevOps full permissions if someone uses SSH to connect a node for troubleshooting.

Role mapping definition

To do that, you can set up a simple Search Guard role without any access limitations:

```
sg_all_access:
  cluster_permissions:
    - *
  index_permissions:
    - index_patterns:
        - "*"
      allowed_actions:
        - "*"
```

Then, configure a role mapping that assigns all requests from 127.0.0.1 to this role:

```
sg_all_access:
  hosts:
    - "127.0.0.1"
```

Advanced: Hostname lookup

Search Guard provides three different modes to resolve the actual hostname against the configured hosts mapping.

- **ip-only Match**: IP addresses only. Default.
- **ip-hostname Match**: IP addresses and hostnames
- **ip-hostname-lookup**: Match IP addresses and hostnames, and perform a reverse hostname lookup

This can be configured in sg_config.yml:

```
searchguard
  dynamic
    hosts_resolver_mode: <mode>
```

Where to go next

- Read our documentation on how to use role mappings to assign roles
- If you have any question, hop over to our forum and ask us

Image: Shutterstock / FlashMovie

This article was published on the Search Guard blog: https://search-guard.com/blog/elasticsearch-assign-role-ip-hostname/