The 5-Minute Security Fix: How to Eliminate Your Elasticsearch's Biggest Vulnerabilities Today

Jochen Kressin · 2025-08-06

Is your Elasticsearch cluster a ticking time bomb? For many, the answer is an uncomfortable "yes." Default installations of Elasticsearch can leave critical data exposed, and these vulnerabilities are actively exploited. The good news? You can drastically improve your cluster's security in about five minutes using Search Guard. This post will walk you through the most common security gaps in Elasticsearch deployments, show you how devastating they can be with real-world examples, and then provide a quick-start guide to closing them. The scary part is that these exact vulnerabilities likely exist in your clusters right now.

The Uncomfortable Truth: Your Elasticsearch Cluster is Likely Exposed

When you first set up Elasticsearch, the primary focus is often on getting it running and ingesting data. Security, unfortunately, can become an afterthought. However, a default Elasticsearch installation often means:

- **No Authentication:** Anyone who can reach your cluster on port 9200 can read, write, or even delete your data. If your cluster is unintentionally exposed to the internet, this is an open invitation for disaster.
- **No Encryption (HTTP by Default):** Data exchanged between your clients and the Elasticsearch cluster, as well as between nodes in the cluster, travels in plain text. This means sensitive information can be intercepted by anyone snooping on your network.
- **No Granular Access Control:** Even if you have some network-level protection, once a user or application can connect, they often have full superuser-like access to all indices and operations.

These aren't theoretical problems; they are active attack vectors. You might think your firewall is enough, but misconfigurations happen, internal threats exist, and a defense-in-depth strategy is crucial for protecting valuable data stored in your Elasticsearch cluster.

Real-World Nightmares: Breaches Caused by These Exact Gaps

The consequences of unsecured Elasticsearch clusters are not hypothetical. Countless organizations have suffered significant data breaches due to these exact vulnerabilities.

The Massive Data Leak (2019 & Ongoing)

Numerous incidents have occurred where terabytes of sensitive customer data (names, addresses, contact information, financial details) were found exposed in unsecured Elasticsearch databases. In many cases, these clusters required no authentication, allowing anyone with the IP address to access the data. Cybercriminals actively scan for such exposed instances. One notable report in early 2020 detailed how a misconfigured Elasticsearch database exposed the records of over 200 million individuals.

Ransomware Attacks

Attackers have increasingly targeted unsecured Elasticsearch (and other NoSQL) databases. They access the unprotected cluster, export or delete the data, and leave a ransom note demanding payment for its return (which often never happens, even if paid). This is often automated, with bots continuously scanning for vulnerable servers. In 2017, a wave of such attacks hit thousands of Elasticsearch and MongoDB instances.

Cryptojacking

Unsecured clusters can also be hijacked for cryptocurrency mining. Attackers install mining software that consumes your server resources, degrading performance and increasing costs, all because they could gain unauthorized access.

These breaches almost always stem from the fundamental lack of authentication, encryption, and proper access control—gaps that Search Guard is designed to fill easily.

Your 5-Minute Fix with Search Guard: A Quick-Start Guide

Search Guard is a comprehensive security plugin for Elasticsearch and Kibana. It provides features like encryption, authentication, authorization, audit logging, and more. The best part? You can get started with essential protections very quickly and completely for free.

Here’s how you can address those major vulnerabilities in just a few steps:

Prerequisites

- You have an Elasticsearch cluster running.
- Search Guard must be installed on *every node* in your Elasticsearch cluster. Mixing secured and non-secured nodes is not supported and will lead to cluster instability.

Step 1: Disable Shard Allocation

Disable shard allocation to prevent Elasticsearch from attempting to redistribute shards. This can be done using either *sgctl* or *curl*.

1. Using sgctl

```
./sgctl.sh rest put _cluster/settings  \
    --json '{"persistent":{"cluster.routing.allocation.enable": "none"}}'
```

2. Using curl

```
curl -Ss -XPUT 'https://localhost:9200/_cluster/settings?pretty' \
-H 'Content-Type: application/json' -d'
{
  "persistent": {
    "cluster.routing.allocation.enable": "none"
  }
}'
```

Step 2: Stop Elasticsearch

Once shard allocation has been disabled, you can safely shut down the Elasticsearch cluster. Depending on how Elasticsearch was started, use one of the following methods:

1. If you are running Elasticsearch with systemd:

```
sudo systemctl stop elasticsearch.service
```

2. If you are running Elasticsearch with SysV init:

```
sudo -i service elasticsearch stop
```

3. If you are running Elasticsearch as a daemon:

```
kill $(cat pid)
```

Step 3: Install Search Guard Plugin

More information about the installation can be found here.

Before configuring Search Guard, you must install the plugin on your Elasticsearch nodes. Note that the Search Guard version must match your Elasticsearch installation's exact version. For example, if you are running Elasticsearch 8.17.5, then you need to install Search Guard for this version. Example:

```
sudo bin/elasticsearch-plugin install \
    https://maven.search-guard.com/search-guard-flx-release/com/floragunn/search-guard-flx-
elasticsearch-plugin/3.0.3-es-8.17.5/search-guard-flx-elasticsearch-plugin-3.0.3-es-8.17.5.zip
```

You can see a list of all available versions on the Search Guard Version Matrix page.

Step 4: Prepare Certificates

Search Guard requires certificates to secure your Elasticsearch cluster. You'll need certificates for both the transport and REST layers. These certificates can either be generated using OpenSSL or through your existing PKI infrastructure.

Transport Layer: Ensures encrypted communication between Elasticsearch nodes. *REST Layer(Optional):* Enables HTTPS for the REST API layer if required.

Step 5: Add TLS Configuration

Once you have your certificates ready, you need to configure TLS (Transport Layer Security) for both the transport and REST layers. Here's how you can do it in your `elasticsearch.yml` :

5.1 Transport Layer Encryption & Admin Certificate Configuration

This ensures secure communication between nodes by enabling TLS on the transport layer and specifying the admin certificate.

Add the following configuration to `elasticsearch.yml`

```
Search Guard.ssl.transport.pemcert_filepath: <path_to_node_certificate>
Search Guard.ssl.transport.pemkey_filepath: <path_to_node_certificate_key>
Search Guard.ssl.transport.pemtrustedcas_filepath: <path_to_root_ca>
Search Guard.ssl.transport.enforce_hostname_verification: true
Search Guard.ssl.transport.pemkey_password: <password>  # Optional, if your private key is
password-protected
Search Guard.authcz.admin_dn:
  - CN=kirk,OU=client,O=client,L=test,C=de
```

- **Search Guard.ssl.transport.pemcert_filepath:** Path to the node certificate.
- **Search Guard.ssl.transport.pemkey_filepath:** Path to the node certificate's private key.
- **Search Guard.ssl.transport.pemtrustedcas_filepath:** Path to the root certificate authority (CA) certificate.
- **Search Guard.ssl.transport.enforce*hostname*verification:** Enforces hostname verification for added security.
- **Search Guard.ssl.transport.pemkey_password:** Password for the private key, if it's protected (optional).
- **Search Guard.authcz.admin_dn:** Distinguished name (DN) for the admin certificate, used to configure and manage Search Guard.

5.2 REST Layer Encryption

If you want to enable HTTPS for the REST API, you'll need to configure the TLS settings for the HTTP layer. Add this to `elasticsearch.yml`

```
Search Guard.ssl.http.enabled: true
Search Guard.ssl.http.pemcert_filepath: <path_to_http_certificate>
Search Guard.ssl.http.pemkey_filepath: <path_to_http_certificate_key>
Search Guard.ssl.http.pemtrustedcas_filepath: <path_to_http_root_ca>
```

Step 6: Restart Elasticsearch Nodes

After configuring the certificates and the TLS settings, restart all Elasticsearch nodes to apply the changes. Ensure that every node has the correct certificate configuration.

Step 7: Re-enable Shard Allocation

Once the cluster has restarted, re-enable shard allocation to allow Elasticsearch to distribute the data across nodes. Use the following command to re-enable shard allocation:

```
./sgctl.sh rest put _cluster/settings \
    --json '{"persistent":{"cluster.routing.allocation.enable": "all"}}'
```

This simple configuration closes the gaping holes of unauthenticated access and unencrypted communication!

Secure Kibana with Search Guard

If you're using **Kibana**, you'll also need to configure it to work with your newly secured **Elasticsearch** cluster. **Search Guard** provides a plugin for **Kibana** that handles authentication. More information for the installation of Kibana can be found here.

Install the Search Guard Kibana Plugin:

Download the Search Guard **Kibana** plugin version that matches your **Kibana** version and install it:

```
bin/kibana-plugin install <link-or-path-to-search-guard-kibana-plugin.zip>
```

Configure `kibana.yml`:

Add the following to your `kibana.yml`:

```
elasticsearch.hosts: ["https://your-elasticsearch-host:9200"] # Use https
elasticsearch.ssl.verificationMode: none # Use 'full' or 'certificate' in production with proper
certs
elasticsearch.username: "kibanaserver" # Default user for Kibana internal server
elasticsearch.password: "kibanaserver" # Default password, change in production!
```

Note:
For production, `elasticsearch.ssl.verificationMode` should be set to `full` or `certificate`, and you should provide the `elasticsearch.ssl.certificateAuthorities` with the path to your CA certificate. The `kibanaserver` user and password should also be changed and configured with minimal necessary permissions in Search Guard.

Restart **Kibana**. You will now be prompted with a login screen when accessing **Kibana**.

Beyond the Basics: Essential Next Steps

What you've accomplished in these few minutes is a monumental step forward. However, security is a journey, not a destination. Your crucial next step is to configure **Role-Based Access Control (RBAC)**.

With RBAC, you can define:

- **Roles:** Collections of permissions (e.g., `read_only_logs`, `inventory_admin`).
- **Role Mappings:** Assign users or backend groups (like LDAP or Active Directory groups) to these roles.

This ensures users and applications only have access to the specific indices and operations they absolutely need. For example, an application writing logs should only have `write` access to log indices, not the ability to delete your customer data.

Dive into the Search Guard documentation to learn about configuring RBAC, setting up audit logging, integrating with LDAP/AD/OIDC, and other advanced features.

Conclusion: Take Control of Your Elasticsearch Security Now

You've seen how common, yet dangerous, default **Elasticsearch** vulnerabilities are and how real-world breaches have exploited these very weaknesses. More importantly, you've learned that a foundational level of security for your **Elasticsearch** and **Kibana** deployments can be achieved remarkably quickly with **Search Guard**. By enabling authentication and encryption, you've already eliminated the biggest, most common threats.

Don't wait to become another statistic. The security of your data is paramount.

Ready to take control? Download and try Search Guard yourself with our 60-day free trial. No credit card or email is required to get started! It's quick to install and easy to configure, allowing you to explore these features hands-on in your own environment. Once you've downloaded the trial, you can follow our comprehensive documentation to set up your first secure cluster.

Feel free to reach out on our community forum if you have questions or encounter any issues – our community and team are here to help you secure your **Elasticsearch** and **Kibana** stack.

This article was published on the Search Guard blog: https://search-guard.com/blog/the-5-minute-security-fix-how-to-eliminate-your-elasticsearchs-biggest/