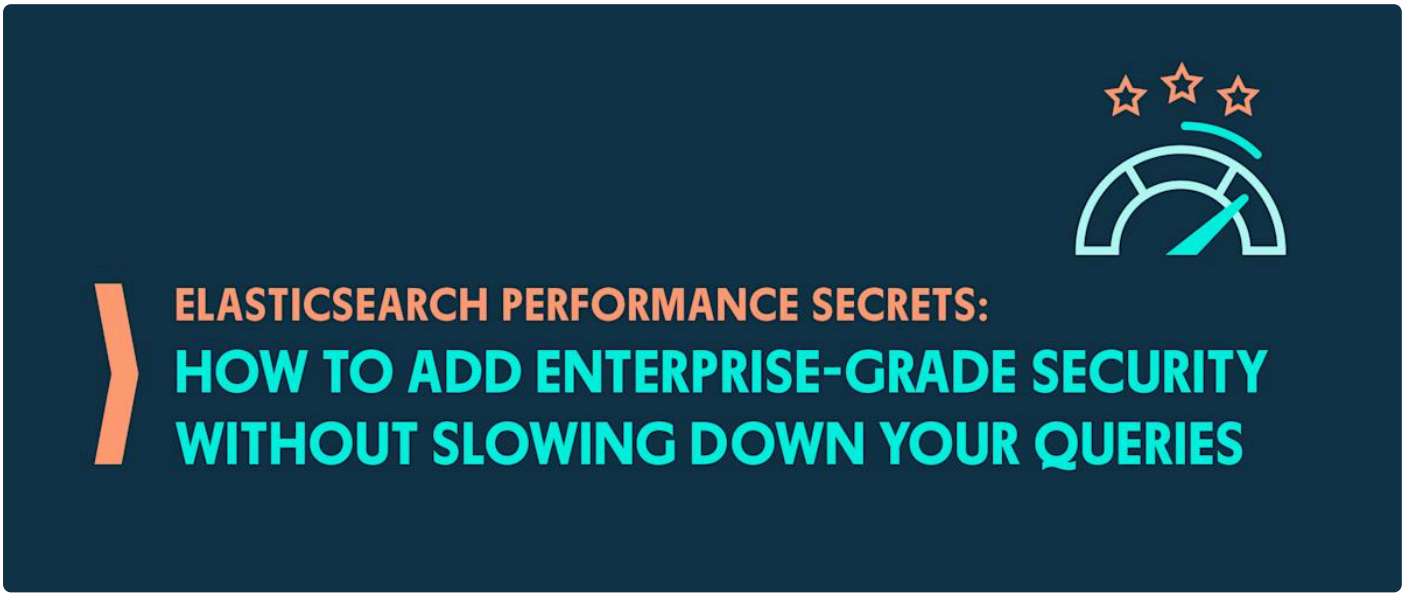
Elasticsearch Performance Secrets: How to Add Enterprise-Grade Security Without Slowing Down Your Queries

Daniel Gleim · 2026-05-12



There is a persistent assumption in many engineering teams that securing an Elasticsearch cluster means accepting slower queries. The logic seems intuitive: every additional layer of processing costs time. Search Guard FLX is designed to prove that assumption wrong.

Non-FLX releases of Search Guard no longer receive new features — whenever possible, Search Guard FLX should be the choice. This post is written for teams running Search Guard FLX and assumes FLX 3.0 or higher throughout. The performance characteristics of each security layer are well understood, and with the right configuration, the impact on query latency is minimal. Understanding what each layer does and which configuration options matter is the difference between a secured cluster that performs well and one that does not.

WHAT SEARCH GUARD ADDS TO YOUR STACK

Search Guard adds several layers to Elasticsearch's request processing: TLS encryption on the transport and REST layer, authentication (LDAP, JWT, SAML, Kerberos and others), RBAC-based authorization, and optionally Document-Level Security (DLS) and Field-Level Security (FLS). Each has a different performance profile, and each can be tuned.

For the vast majority of deployments, the performance impact of a properly configured Search Guard installation is negligible. The sections below explain why and what to watch out for.

TLS: SECURITY AND PERFORMANCE IN ONE CONFIGURATION STEP

For the REST layer, TLS is optional, while it is mandatory for the transport layer. In practice, both layers should be encrypted in any production environment. `searchguard.ssl.http.enabled` controls whether TLS is enabled on the REST layer. If enabled, only HTTPS is allowed. The default is `true` since FLX 4.0.0.

FLX 4.0.0 blocks outdated and insecure SSL/TLS protocols entirely. Use of insecure ciphers and protocols such as SSLv3, TLS 1.0, TLS 1.1, and weak export ciphers is no longer permitted. Administrators will receive a clear error message if they attempt to enable deprecated protocols in the configuration. This means Search Guard actively enforces modern, efficient cryptography out of the box.

You can limit the allowed ciphers and TLS protocols for both the REST and transport layer. For example, restricting connections to strong ciphers only and limiting TLS versions to the most recent ones. If this is not set, ciphers and TLS versions are negotiated automatically, which in some cases can lead to a weaker cipher suite being used. Explicit configuration via `searchguard.ssl.http.enabled_ciphers`, `searchguard.ssl.http.enabled_protocols`, `searchguard.ssl.transport.enabled_ciphers` and `searchguard.ssl.transport.enabled_protocols` both hardens security and ensures optimal cipher selection — two goals achieved in a single step.

For operators migrating from older versions: support for OpenSSL was removed from Search Guard some time ago. If you have any settings in `elasticsearch.yml` mentioning OpenSSL, remove these before starting FLX. Search Guard FLX 4.0.0 now defaults to the Java Cryptography Extensions (JCE) instead of the previously used Bouncy Castle library. In most common deployments this requires no action, but testing is recommended if obscure or legacy cipher formats are in use.

AUTHENTICATION CACHING: MINIMAL ROUNDTRIPS BY DEFAULT

One of the reasons Search Guard has minimal impact on authentication performance is its built-in user cache. Search Guard uses a cache to store the user information of authenticated users, avoiding the need to query authentication backends for each request.

The cache is enabled by default. Cache contents expire two minutes after they have been written, and the cache is limited to 1000 entries. For teams migrating from legacy Search Guard: before FLX, Search Guard would keep users cached for one hour by default, and the cache size was unlimited. The FLX defaults strike a better balance - fast authentication with prompt propagation of role changes.

Cache settings can be adjusted in `sg_authc.yml` using `user_cache.expire_after_write` (defaults to `2m`) and `user_cache.max_size` (defaults to 1000 entries). Because `sg_authc.yml` is live-reloadable, any tuning can be applied at runtime without a cluster restart. The defaults work well for most deployments. The main thing to avoid is disabling the cache entirely, which removes this performance advantage for LDAP and the Internal User Database.

LDAP ROLE RESOLUTION: BUILT FOR EFFICIENCY IN FLX

LDAP with nested role resolution is one of the areas where Search Guard FLX delivers a concrete performance improvement over its predecessor. While the old LDAP implementation required at least one network roundtrip per role, the new FLX implementation batches requests so that the number of

roundtrips is not higher than the depth of the search tree. For users with many roles, this dramatically reduces LDAP traffic.

Recursive group search is opt-in: `ldap.group_search.recursive.enabled` defaults to `false`. When enabled, `ldap.group_search.recursive.enabled_for` can be set to a regular expression pattern to allow recursion only for DNs which match the specified pattern — a simple way to limit recursive lookups to the groups that actually require them.

By default, the results of group searches are cached, with TTL and maximum size configurable via `ldap.group_search.cache.expire_after_write` (defaults to `2m`) and `ldap.group_search.cache.max_size` (defaults to 1000 entries). This group search cache operates independently from the user cache, giving LDAP-heavy environments an additional tuning lever.

The connection pool rounds out the picture: `ldap.idp.connection_pool.min_size` defaults to 3, and `ldap.idp.connection_pool.max_size` defaults to 10. These values can be tuned to match actual concurrency patterns in high-load environments.

PRIVILEGE EVALUATION: CONSTANT TIME, REGARDLESS OF SCALE

Search Guard FLX fundamentally changed how authorization works under the hood. In most cases, privileges can now be evaluated in constant time. Earlier versions of Search Guard had linear or even quadratic complexity, depending on the number of roles and indices. This is achieved by a number of different techniques: the mapping from backend roles to Search Guard roles is performed using trie data structures, and index and action name patterns are resolved in advance against all indices in the cluster.

This means that as your cluster grows — more roles, more indices, more users — authorization overhead does not grow with it. It is one of the most significant architectural improvements in FLX, and it happens transparently.

DOCUMENT-LEVEL SECURITY: POWERFUL AND EFFICIENT

DLS gives you fine-grained control over which documents each user can access without duplicating indices or managing separate clusters per user group. A DLS query can be as simple or complex as necessary, using the full range of Elasticsearch's query DSL. Think of the DLS query as an additional query executed on top of your original one. In practice, straightforward DLS queries add negligible overhead to query execution.

Search Guard FLX ships with a new DLS/FLS implementation specifically designed for efficiency. The new implementation provides better efficiency and functionality compared to the legacy implementation. It can be activated by creating or editing `sg_authz_dlsfls.yml` and adding `use_impl: flx`, a one-line configuration change that is worth making on any fully migrated FLX cluster.

AUDIT LOGGING: COMPLIANCE WITHOUT CLUSTER IMPACT

Search Guard makes it straightforward to meet audit requirements for GDPR, HIPAA, PCI, SOX and similar frameworks. Audit logging is disabled by default. To enable it, configure `searchguard.audit.type` in `elasticsearch.yml`.

Once enabled, the design ensures minimal cluster impact: all events are logged asynchronously, so the overall performance of your cluster will only be affected minimally. Search Guard uses a fixed thread pool with a default size of 10, and the maximum queue length per thread defaults to 100,000.

The default category configuration is already optimized for most use cases: the Audit Log module logs all events in all categories, but excludes `AUTHENTICATED` and `GRANTED_PRIVILEGES` by default, as these represent successful requests. These exclusions are intentional, enabling them would log every valid request in the cluster, which in most cases adds volume without adding security value. The defaults give you full coverage of security-relevant events with minimal noise.

ROLE AND PERMISSION DESIGN: SCALE WITHOUT COMPLEXITY

Search Guard FLX 3.0 introduced a cleaner, more performant way to manage permissions at scale. For improved performance, it is recommended to apply permissions on data stream and alias level, instead of directly on indices. Support for aliases and data streams was added in Search Guard FLX 3.0.

For environments with large numbers of time-series indices behind a single alias, this approach simplifies role definitions significantly and reduces the number of index pattern evaluations per request. It is one of those configuration decisions that simultaneously improves security manageability and cluster performance.

SUMMARY

A properly configured Search Guard FLX deployment adds enterprise-grade security to your Elasticsearch cluster with minimal impact on query performance. The key decisions that keep it that way:

- Configure TLS cipher suites and protocols explicitly — this both hardens security and ensures optimal cipher selection
- Keep the authentication cache enabled at its defaults — it is what makes LDAP and the Internal User Database fast
- Enable LDAP recursive group search only where needed, and use `ldap.group_search.recursive.enabled_for` to scope it precisely
- Activate the FLX DLS/FLS implementation via `sg_authz_dlsfls.yml` once all nodes are fully migrated
- Use the default audit logging category configuration — it already excludes high-volume non-security events
- Apply permissions at alias and data stream level (FLX 3.0+) for both better performance and simpler role management

Security and performance are not a trade-off. With Search Guard FLX, they are the same goal.

This article was published on the Search Guard blog: https://search-guard.com/blog/elasticsearch-performance-secrets-how-to-add-enterprise-grade-security/