An introduction to TLS certificates

Cliff Staley · 2018-10-08

Secured inter-node communication is a must-have when protecting Elasticsearch clusters. We have explained that in details in the article Security in distributed Systems. At Search Guard we apply TLS which assures message integrity and encrypts data transferred through the network based on symmetric cryptography. TLS handshake is a phase in which a client and a server agree on encryption methods and their parameters. Agreement is met based on PKI infrastructure, which uses certificates to check identity of both sides that take part in a handshake.

In this article we explain what TLS certificates are and how do they work.

Certificate content

We assume You are familiar with our search-guard.com website wich uses HTTPS protocol. Once You are on the website, it is reasonable to assure that website You view in a browser truly belongs to us. You can check it by clicking on a HTTPS padlock to view a certificate.

Distinguished Name

The first thing You can check is an issuer of the certificate. The issuer is identified by a DN (distinguished name). DN can contain following fields:

- **CN** - Common Name
- **OU** - Organizational Unit
- **O** - Organisation
- **L** - Locality, usually a city name
- **ST** - State
- **C** - Country

Fields are not mandatory and barely all of them are used. The issuer of the Search Guard website's certificate is:

- Country (C): BE
- Organisation (O): GlobalSign nv-sa
- Common Name (CN): AlphaSSL CA - SHA256 - G2

Distinguished name's structure is also used to identify a certificate's subject, in this case our company.

Subject Alternative Name (SAN)

A certificate can be only used within a specified scope, which can be defined with:

- specified IP addresses,
- hostnames,
- domain names
- wildcards.

This value is set within Subject Alternative Name (SAN) section of a certificate. In case of search-guard.com it is:

- DNS Name: *.search-guard.com
- DNS Name: search-guard.com

This means the certificate is valid for search-guard.com domain and all its subdomains. So if you get the same certificate from a website non-search-guard.com, it will not be valid anymore and padlock in a browser will be open.

Chain of trust

We already know who has issued the certificate of a website. The interesting question is: *"Can we trust the issuer?"*. To answer that question we need to check issuer of the issuer. Then we check its issuer and move further recursively. We stop once we reach an issuer, that we all trust. This happens when we reach Root Certificate Authority (Root CA).

TLS relies on a chain of trust which starts at Root CA that everybody can trust. TLS always checks if all certificates within a chain are trusted. If not, You can see a padlock open within Your browser, which means a certificate is not trusted.

Chain of trust can be also viewed in a browser and in case of search-guard.com, it is:

```
* GlobalSign Root CA
         * AlphaSSLA CA - 256 - G2
           * *.search-guard.com
```

where *.search-guard.com* is a common name of a subject within a certificate.

Self Signed Certificates

When we access search-guard.com website, we need a third party that confirms website identity. This is necessary because, unless You do not live in Berlin, it would have been tough visit our office and check the certificate personally.

However it is still possible to come to our office. Here You could see us generating a certificate and signing it with our private key. In this case, You would not need to ask Certificate Authorities (CA) because You would know that:

- The certificate is signed with our private key.
- The certificate was created by us.

You could also take the certificate home with You. As a result, you could trust it.

Visiting our office to get a certificate may sound silly. In fact, this is how self-signed certificates work and how You can make use of them with Search Guard.

Self-signed certificate is a certificate that is signed by a private key of the same identity. This assures that when You connect to a server and send data encrypted with server's public key, only the private key used to sign a certificate is able to decrypt data.

Differences of self-signed certificates

Technically, self-signed certificates are exactly the same as those issued by third party authorities. There are some differences worth mentioning:

- Some CAs are able to verify the identity of a person to whom a certificate has been issued. Distinguished Name can contain such information. It is not possible with self-signed certificates but it is not the case when securing inter-node communication in Elastic Search clusters.
- Relying on CA introduces another possible attack vector - an attack on CA.
- Self-signed certificates are free and are faster to deploy.
- They can be also easily customized. For instance, a larger encryption key can be applied.

To sum up, the potential drawbacks have no impact in our use case. On the other hand, the advantages allow You to deploy TLS faster and at lower cost.

Generating certificates

Certificates and keys can be generated with **openssl** and **keytool** - the two popular tools for generating cryptographic keys. Although they have many features in common, some are unique.

Keytool allows to manage Java KeyStores, which are Java containers for authorisation certificates. Keystore files have .jks extension.

OpenSSL works well with standard formats (PEM/CER/CRT/PKCS) but cannot manipulate Java keystore files.

This is why in many cases we need both tools:

- Generate keys and certificates with **openssl**,

- Import generated data with **keytool** into Java keystore.

Non self-signed certificates need to be issued by a third party - Certificate Authority (CA). To achieve that, Certificate Signing Request (CSR) needs to be sent from an applicant (later certificate subject) into CA. This is not necessary for self-signed certificates, as CSR is done locally.

Generating keys and certificates is not easy. One needs to know several encryption algorithms to pick up the best, decide on a key size, and many other parameters. Fortunately this has been already done by us: for both PoC purpose and production environment.

If You are conducting a proof of concept, You can use our TLS Certificate Generator and certificates will be sent to You by email. This should not be used for production environment as the keys are sent via internet.

We provide You with ready-to-go scripts (see “Using the example PKI scripts” section) that generate all cryptographic data to configure TLS.

Summary

No matter how good You try to protect Your Elasticsearch, it is worth nothing once communication between cluster nodes is not secured. At Search Guard we understand it deeply and have put a lot of effort to make TLS work and secure our clients’ data. The decision has been made based on our extensive experience and compliance regulations (GDPR, HIPAA, PCI, SOX) that our clients need to meet. Meeting them requires transmission security and integrity controls. TLS is the best on the market technology to do so.

Where to go next:

- Read our Introduction to TLS post
- Generate your own TLS certificates by using our TLS Tool
- Read about the Search Guard TLS configuration options in our documentation
- If you have problems configuring TLS, read our TLS troubleshooting guide or ask for help on our forum.

Image: shutterstock / Shawn Hempel

This article was published on the Search Guard blog: https://search-guard.com/blog/elasticsearch-tls-certificates/
Search Guard is Security and Alerting for Elasticsearch and Kibana — search-guard.com · Start a free trial