Managing TLS certificates with Certificate Revocation Lists

Jochen Kressin · 2020-07-16

To implement enterprise-grade security for Elasticsearch, Search Guard uses TLS a lot. TLS makes sure your traffic cannot be sniffed or tampered with, and only trusted nodes can join the cluster. But what if a TLS certificate or private key is lost or compromised? Certificate revocation is here for the rescue!

What is certificate revocation?

A Certificate Revocation List (CRL) is

“a list of digital certificates that have been revoked by the issuing Certificate Authority (CA) before their scheduled expiration date and should no longer be trusted. CRLs are a type of blacklist and are used by various endpoints, including Web browsers, to verify whether a certificate is valid and trustworthy.” (https://searchsecurity.techtarget.com/definition/Certificate-Revocation-List)

You can use a CRL to either permanently or temporarily revoke certificates. There are two different states for a revoked certificate:

Revoked: A certificate is permanently revoked and should no longer be trusted. This status is irreversible. It is used for example, if a certificate was issued by mistake, or a private key has been compromised.

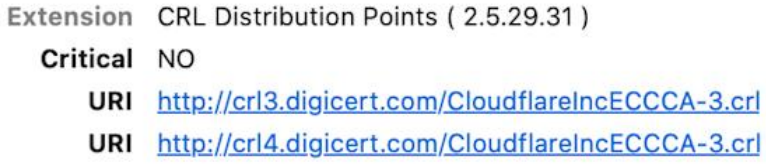
Hold: A certificate is temporarily revoked and should not be trusted at the moment. This status is reversible. It is used, for example, if a user thinks a private key has been compromised, but it turns out this is not the case.

In other words, a root CA can revoke a certificate at any time, rendering it invalid. The question is - how does a consumer of a certificate (in our example Elasticsearch / Search Guard) know what certificates are revoked?

There are two approaches: Certificate Revocation Lists (CRL) and the Online Certificate Status Protocol (OCSP). OCSP has largely superseded CRLs. However, Search Guard supports both approaches.

Certificate Revocation Lists

Each CA is required to keep track of revoked certificates. After the CA has revoked a certificate, it adds it's serial number to the CA's list of revoked certificates. This list is publicly available. The URL can be found in the *CRL Distribution Points* section of the certificate:



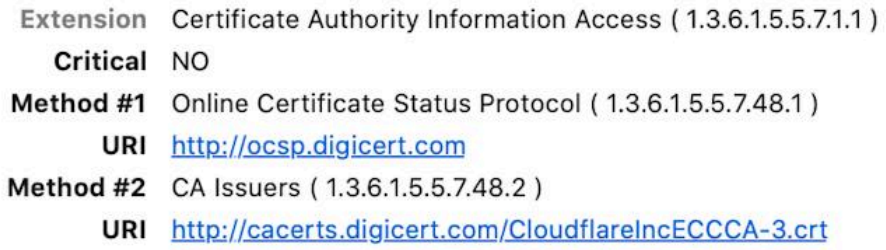
The problem with this approach is that the CRL tends to become very large over time because CAs constantly add new certificates to it. Also, to check the validity of one certificate, you need to always download the complete list. Not really an optimal approach. Let's look at OCSP as an alternative.

Online Certificate Status Protocol

"The Online Certificate Status Protocol (OCSP) is an Internet protocol used for obtaining the revocation status of an X.509 digital certificate" (https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol)

How does it work? In short, instead of downloading a potentially large CRL file periodically, you would query the CA's OCSP server to check the validity of one or more certificates.

The certificates serial number is transmitted to the OSCP server, and the validation result is based on the server's response. As with CRLs, the OCSP endpoint is part of each certificate extensions section:



Communication is usually done via HTTP in a request/response style, so the OCSP server is sometimes called the *OSCP responder*.

On the pro side, OCSP transmits less data than the CRL approach, and puts less burden on the network and on the client that parses the response. On the con side, by querying an OCSP responder for the validity of a certificate, information is transmitted to the responder that a particular network host used a particular certificate at a particular time. Since encryption is not mandatory with OCSP, other parties may sniff this information.

Managing TLS certificates for Elasticsearch with CRL and OCSP

Search Guard uses TLS for securing Elasticsearch traffic on the REST and on the transport layer, and supports both CRL and OCSP. By default this feature is switched off, so let's enable it by adding the following line to elasticsearch.yml:

```
searchguard.ssl.http.crl.validate: true
```

That's it. If a certificate contains either an entry for CRL or OCSP, Search Guard will now check for the validity of the certificate. By default, both CRL and OCSP endpoints are enabled. If a certificate contains both, OCSP is preferred.

Of course, the CRL feature comes with many options that you can use to fine-tune the behavior. For example:

```
searchguard.ssl.http.crl.disable_crldp: true
Disable CRL completely. Default is false.
```

```
searchguard.ssl.http.crl.disable_ocsp: true
Disable OCSP completely. Default is false.
```

```
searchguard.ssl.http.crl.prefer_crlfile_over_ocsp: true
If set to true, CRL is preferred over OCSP. Default is false.
```

Intermediate certificates

By default, Search Guard would only validate leaf certificates. If you are using intermediate certificates and want to verify their validity too, add the following line to elasticsearch.yml:

```
searchguard.ssl.http.crl.check_only_end_entities: false
```

Of course, this comes with a slight performance overhead but adds an additional layer of security.

Summary

Any CA can revoke any issued certificate at any time. Revoked certificates can have either the *revoked* or *hold* state, and are deemed invalid. This is useful if a certificate was issued by mistake or a private key was compromised.

To check if a certificate is revoked, you can either use Certificate Revocation Lists (CRL) or the Online Certificate Status Protocol (OCSP). The URL endpoints of the CRL or OCSP are contained in the

certificate itself.

When running Search Guard for Elasticsearch you can use both CRLs or OCSP to check if a node certificate, admin certificate, or client certificate has been revoked. If a certificate has been revoked, it is deemed invalid.

If your PKI supports it, we recommend enabling this feature for additional security.

Where to go next

- Visit our documentation for a deep dive into certificate revocation

Image: shutterstock / Yilong Xiao

This article was published on the Search Guard blog: https://search-guard.com/blog/elasticsearch-tls-certificate-revocation/