Solving "Unable to Get Local Issuer Certificate" Errors with cURL

Jochen Kressin · 2024-10-29

As a developer or DevOps professional working with Elasticsearch and Kibana, you might frequently use cURL to interact with the Elasticsearch REST API.

One common error you may encounter is:

```
curl: (60) ssl certificate problem: \
  unable to get local issuer certificate
```

This error can be perplexing, especially when you need to ensure secure communication with your Elasticsearch cluster. In this article, we'll delve into what this error means, its connection to the TLS chain of trust, and how to resolve it effectively.

Understanding the Error

The error message `"ssl certificate problem: unable to get local issuer certificate"` indicates that cURL is unable to verify the SSL/TLS certificate presented by the Elasticsearch server. Essentially, this means cURL cannot find a trusted certificate in the local certificate store that can authenticate the server's certificate. This verification failure is a security feature to ensure that you only connect to trusted servers.

The TLS Chain of Trust

To understand why this error occurs, it's important to grasp the concept of the TLS (Transport Layer Security) chain of trust. The TLS chain of trust is a sequence of certificates, starting from the server's certificate, passing through intermediate certificates, and ending at a trusted root certificate. This

hierarchy is designed to establish a secure and trusted connection between the client and server. Here's how it works:

1. **Server Certificate**: Issued to the Elasticsearch server by a Certificate Authority (CA). It is presented to the client during the SSL/TLS handshake.
2. **Intermediate Certificates**: These certificates bridge the server certificate and the root certificate. They are issued by a CA and can issue other certificates.
3. **Root Certificate**: The top-most certificate in the chain, self-signed by a trusted CA. Root certificates are pre-installed in the operating system's certificate store or browser.

For a secure connection, cURL must validate the entire chain. If any link in this chain is missing or invalid, the verification fails, leading to the error.

Causes and Solutions

Several issues can trigger this error when connecting to the Elasticsearch REST API. Here are some common causes and their solutions:

Missing Intermediate Certificates

Cause: The Elasticsearch server might not be sending the necessary intermediate certificates to complete the chain of trust.

Solution: Check your Elasticsearch server's SSL/TLS configuration to ensure that it includes the full certificate chain. You can use tools like SSL Labs' SSL Test to verify the server's configuration. Make sure all intermediate certificates are included in your Elasticsearch configuration files.

Outdated or Incomplete CA Certificates Store

Cause: Your local CA certificates store might be outdated or incomplete, lacking the certificates needed to validate the chain. This is only relevant if you use certificates that are ultimately signe by a well-known Root CA. Companies usually maintain their own Root CAs.

Solution: Update your CA certificates store. For Linux users, updating the `ca-certificates` package is usually the solution. For Debian-based systems, run:

```
sudo apt-get update
sudo apt-get install ca-certificates
```

On macOS, use Homebrew to update the certificates:

```
brew update
brew install curl-ca-bundle
```

Custom CA Certificates

Cause: You might be using custom or private CA certificates that are not recognized by the default CA store. This is usually the case for companies.

Solution: Tell cURL explicitly to use a specific CA certificate file or directory. You can do this using the `--cacert` option to specify a file or the `--capath` option to specify a directory. For example:

```
curl --cacert /path/to/custom-ca.pem \
  https://your-elasticsearch-domain.com
```

Insecure Bypass (Not Recommended)

Cause: In a development or testing environment, you might not have the necessary certificates.

Solution: Although not recommended for production environments, you can bypass certificate validation using the `-k` or `--insecure` option. This should only be used when security is not a concern, such as in a controlled development environment:

```
curl -k https://your-elasticsearch-domain.com
```

Conclusion

Encountering the `"ssl certificate problem: unable to get local issuer certificate"` error when using cURL with the Elasticsearch REST API highlights the importance of a correctly configured and complete TLS chain of trust. By ensuring your Elasticsearch server sends the full certificate chain, keeping your local CA store updated, and properly handling custom CA certificates, you can resolve this error effectively. Understanding these principles helps maintain secure and reliable communication with your Elasticsearch cluster, reinforcing your application's overall security posture.

Where to go Next

- Read our article on TLS troubleshooting
- Browse all our articles tagged with TLS
- If you need to create TLS certificates for your Elasticsearch cluster, check out our offline TLS Tool

---

This article was published on the Search Guard blog: https://search-guard.com/blog/elasticsearch-curl-unable-to-get-local-issuer-certificate/