TLS encryption and mutual authentication using syslog-ng Open Source Edition

December 09, 2014

Table of Contents

Collecting log messages is an essential part of managing, maintaining, and troubleshooting IT systems. Since your log messages can contain all kinds of sensitive information, you should make sure that they are kept safe. The easiest way to protect the log messages as they are transferred from your clients to your logserver is to authenticate and encrypt the connection between the client and the server.

This tutorial shows you step-by-step how to create the certificates required to authenticate your server and your clients, and how to configure syslog-ng Open Source Edition (syslog-ng OSE) to send your log messages in an encrypted connection. Installing syslog-ng OSE is not covered, but *downloading it* for your platform and installing it should be easy.

The tutorial is organized as follows:

- *Section 1, Creating self-signed certificates (p. 4)* describes how to create the required certificates to encrypt and authenticate the connection between your logserver and your clients. Actually, you can use this part of the tutorial even if you do not use syslog-ng OSE, as it is independent from the logging application you use.
- *Section 2, Configuring syslog-ng OSE (p. 9)* describes how to configure syslog-ng OSE on your clients and your logserver.
- *Procedure 3, Testing what you have done (p. 10)* gives you tips on how to test your configuration to make sure it is really working.

1. Creating self-signed certificates

TLS-encryption uses certificates to authenticate the server, and in case of mutual authentication, the client as well. The following sections show you how to create the required certificates.

To use mutual authentication in syslog-ng OSE, certificates are required. There are several commercial certificate authorities (CAs) who can help you, but the process costs both money and time (waiting until the submitted certificate is signed). This guide demonstrates how to create your very own Certificate Authority (CA) for creating self-signed certificates. It does not cover all the details, for example, changing expiration dates, only the minimally required steps to be able to use mutual authentication in syslog-ng OSE.

There are handy tools, such as CA.pl, which can make certificate creation and signing easier, but they are not available on all platforms, even if it is part of the OpenSSL software suite. On the other hand, the OpenSSL command line tool is available on all Linux distributions and BSD variants, so this tool will be used in the guide.

1.1. Procedure – Creating a CA

Purpose:

To create a CA, complete the following steps:

Steps:

Step 1. Create an empty directory and navigate into that directory:

```
mkdir CA

cd CA
```

Step 2. Create a few directories and give starting values to some support files:

```
mkdir certs crl newcerts private

echo "01" > serial

cp /dev/null index.txt
```

Step 3. Copy `openssl.conf` to the current directory. Depending on your distributions, the source directory might be different, so check the list of files in the OpenSSL package before copying:

```
cp /etc/ssl/openssl.cnf openssl.cnf
```

Step 4. Edit `openssl.conf` in the current directory:

```
vi openssl.cnf
```

Step 5. Search for the following part and replace `./DemoCA` with a single dot:

```
[ CA_default ]

dir             = ./demoCA              # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
```

Change it to:

```
[ CA_default ]

dir             = .                     # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
```

Step 6. As a last step, generate the certificate for the CA:
`openssl req -new -x509 -keyout private/cakey.pem -out cacert.pem -days 365 -config openssl.cnf`

The following will be displayed. Answer the questions as in the example:

```
Generating a 1024 bit RSA private key
.++++++
.....................++++++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HU
State or Province Name (full name) [Some-State]:Budapest
Locality Name (eg, city) []:Budapest
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BalaBit
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:Peter Czanik
Email Address []:czanik@balabit.com
```

1.2. Procedure – Creating a server certificate

Purpose:

To create a server certificate, complete the following steps:

Steps:

Step 1. The next step is to create and sign a certificate for your syslog-ng OSE server. The common name should contain the FQDN or IP address of your server, and the e-mail address should be left blank.
`openssl req -nodes -new -x509 -keyout serverkey.pem -out serverreq.pem -days 365 -config openssl.cnf`

Step 2. The following will be displayed. Answer the questions as in the example:

```
Generating a 1024 bit RSA private key
.....................................++++++
.++++++
writing new private key to 'serverkey.pem'
```

```
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HU
State or Province Name (full name) [Some-State]:Budapest
Locality Name (eg, city) []:Budapest
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BalaBit
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:172.16.177.147
Email Address []:
czanik@linux-modi:~/CA> openssl x509 -x509toreq -in serverreq.pem -signkey
serverkey.pem -out tmp.pem
Getting request Private Key
Generating certificate request
czanik@linux-modi:~/CA> openssl ca -config openssl.cnf -policy policy_anything
 -out servercert.pem -infiles tmp.pem
Using configuration from openssl.cnf
Enter pass phrase for ./private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jun 25 10:27:39 2014 GMT
            Not After : Jun 25 10:27:39 2015 GMT
        Subject:
            countryName               = HU
            stateOrProvinceName       = Budapest
            localityName              = Budapest
            organizationName          = BalaBit
            commonName                = 172.16.177.147
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
               55:4E:B1:47:33:CF:0C:83:5F:29:64:9B:E9:99:77:DF:0E:72:52:76

            X509v3 Authority Key Identifier:

keyid:D1:FF:ED:B4:0B:66:E6:45:EE:70:4F:DC:6C:C5:34:48:42:38:E9:38

Certificate is to be certified until Jun 25 10:27:39 2015 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
```

```
Write out database with 1 new entries
    Data Base Updated
```

Step 3. Enter the following:
`rm tmp.pem`

1.3. Procedure – Creating a client certificate

Purpose:

To create a client certificate, complete the following steps:

Steps:

Step 1. The steps for the client(s) are very similar, only the file names and the embedded common name (host identifier: FQDN or IP address) are different. If you have multiple clients, make sure that each has the right host identifier.
`openssl req -nodes -new -x509 -keyout clientkey.pem -out clientreq.pem -days 365 -config openssl.cnf`

Step 2. The following will be displayed. Answer the questions as in the example:

```
Generating a 1024 bit RSA private key
.....................................................................++++++
.............................++++++
writing new private key to 'clientkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:HU
State or Province Name (full name) [Some-State]:Budapest
Locality Name (eg, city) []:Budapest
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BalaBit
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:172.16.177.129
Email Address []:

czanik@linux-modi:~/CA> openssl x509 -x509toreq -in clientreq.pem -signkey
clientkey.pem -out tmp.pem
Getting request Private Key
Generating certificate request
czanik@linux-modi:~/CA> openssl ca -config openssl.cnf -policy policy_anything
 -out clientcert.pem -infiles tmp.pem
Using configuration from openssl.cnf
Enter pass phrase for ./private/cakey.pem:
Check that the request matches the signature
Signature ok
```

```
Certificate Details:
        Serial Number: 2 (0x2)
        Validity
            Not Before: Jun 25 10:28:49 2014 GMT
            Not After : Jun 25 10:28:49 2015 GMT
        Subject:
            countryName               = HU
            stateOrProvinceName       = Budapest
            localityName              = Budapest
            organizationName          = BalaBit
            commonName                = 172.16.177.129
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                91:D9:99:95:F2:0D:22:BF:72:95:56:9A:C0:DF:A3:07:5C:E2:3F:63

            X509v3 Authority Key Identifier:

keyid:D1:FF:ED:B4:0B:66:E6:45:EE:70:4F:DC:6C:C5:34:48:42:38:E9:38

Certificate is to be certified until Jun 25 10:28:49 2015 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Step 3. Enter the following:

```
rm tmp.pem
```

2. Configuring syslog-ng OSE

Once you are ready with generating CA, server and client certificates, copy them to the respective machines and configure syslog-ng OSE to use them. In theory, the CA and other certificates could be placed anywhere in the file system. In practice, server applications, such as syslog-ng OSE are often protected by AppArmor, SELinux or other mechanisms, therefore it is recommended to create sub-directories where the `syslog-ng.conf` resides. This way syslog-ng OSE can read them without modifying the related access rules.

2.1. Procedure – Configuring the syslog-ng OSE server

Purpose:

In the following example `syslog-ng.conf` is under `/usr/local/etc/syslog-ng`, but it could be `/opt/syslog-ng/etc/`, `/etc/syslog-ng/` or any other directory in your system, so adopt the configuration example accordingly.

Steps:

Step 1. As a first step, create two new directories under the syslog-ng OSE configuration directory:
`mkdir cert.d ca.d`

Step 2. Copy `serverkey.pem` and `servercert.pem` to `cert.d`. Copy `cacert.pem` to `ca.d` and issue the following command on the certificate:
`openssl x509 -noout -hash -in cacert.pem`

The result is a hash (for example 6d2962a8), a series of alphanumeric characters based on the Distinguished Name of the certificate.

Step 3. Issue the following command to create a symbolic link to the certificate that uses the hash returned by the previous command and the `.0` suffix.
`ln -s cacert.pem 6d2962a8.0`

Step 4. Adopt the following configuration example to your `syslog-ng.conf` by changing the IP and port parameters and directories to your local environment. In the log statement replace “`d_local`” with an actual log destination name in your configuration (for example the one that refers to `/var/log/messages`).

```
source demo_tls_source {
tcp(ip(0.0.0.0) port(6514)
tls( key_file("/usr/local/etc/syslog-ng/cert.d/serverkey.pem")
cert_file("/usr/local/etc/syslog-ng/cert.d/servercert.pem")
ca_dir("/usr/local/etc/syslog-ng/ca.d"))
); };

log { source(demo_tls_source); destination(d_local); };
```

Step 5. Finally, restart syslog-ng OSE for the configuration changes to take effect.

2.2. Procedure – Configuring syslog-ng OSE clients

Purpose:

Configuring the client side is similar to the server, the difference is in the configuration part. In the following example `syslog-ng.conf` is under `/etc/syslog-ng`, but it could be `/opt/syslog-ng/etc/`, `/usr/local/etc/syslog-ng/` or any other directory on your system, so adopt the configuration example accordingly.

Steps:

Step 1. As a first step, create two new directories under the syslog-ng OSE configuration directory:
`mkdir cert.d ca.d`

Step 2. Copy `serverkey.pem` and `servercert.pem` to `cert.d`. Copy `cacert.pem` to `ca.d` and issue the following command on the certificate:
`openssl x509 -noout -hash -in cacert.pem`

The result is a hash (for example 6d2962a8), a series of alphanumeric characters based on the Distinguished Name of the certificate.

Step 3. Issue the following command to create a symbolic link to the certificate that uses the hash returned by the previous command and the `.0` suffix.
`ln -s cacert.pem 6d2962a8.0`

Step 4. Adopt the following configuration example to your `syslog-ng.conf` by changing the IP and port parameters and directories to your local environment. In the log statement replace "`src`" with an actual log source name in your configuration.

```
destination demo_tls_destination {
tcp("172.16.177.147" port(6514)
tls( ca_dir("/etc/syslog-ng/ca.d")
key_file("/etc/syslog-ng/cert.d/clientkey.pem")
cert_file("/etc/syslog-ng/cert.d/clientcert.pem") )
); };

log { source(src); destination(demo_tls_destination); };
```

Step 5. Finally, restart syslog-ng OSE for the configuration changes to take effect.

3. Procedure – Testing what you have done

Purpose:

After configuring syslog-ng OSE, test if everything works as expected.

Steps:

Step 1. On the client side, enter the following command:
`logger "This is a test message"`

Step 2. On the server side, `tail` the file, where logs from the network are arriving. You should see something similar in case of the above test message:

```
tail -f /var/log/messages | grep test
```

```
Jun 26 19:12:06 172.16.177.129 root: This is a test message
```

If you cannot see it, check the log file, where the internal messages of syslog-ng are stored, both on the server and the client side. The most common causes of the problem are the following:

- There is no trace of connection at all (internal logs show connection attempts), there is a network / firewall problem, or incorrectly configured destination or listening IP.
- With new certificates an incorrectly configured clock can already cause problems. Check if all of your systems have the same time / time zone.
- Make sure, that the Common Name is set to the correct FQDN or IP address. If you use FQDN, make sure, that your DNS server works correctly.
- Do not include an e-mail address in the client and server certificates.

4. Summary

This tutorial has shown you how to encrypt and authenticate the connection between your clients and your logserver.

- If you have run into problems, or need help, leave a comment, or post your problem on the *syslog-ng mailing list*.
- If you would like to know more about syslog-ng OSE, visit the *syslog-ng project page*, or check the *documentation*.