



syslog-ng Premium Edition 7.0.26

Mutual authentication using TLS

Copyright 2021 One Identity LLC.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC .

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (<http://www.OneIdentity.com>) for regional and international office information.

Patents

One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <http://www.OneIdentity.com/legal/patents.aspx>.

Trademarks

One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at www.OneIdentity.com/legal. All other trademarks are the property of their respective owners.

Legend

 **WARNING: A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.**

 **CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.**

syslog-ng PE Mutual authentication using TLS
Updated - 05 July 2021, 02:23
Version - 7.0.26

Contents

Introduction	4
Creating self-signed certificates	5
Creating a CA	5
Creating a server certificate	7
Creating a client certificate	8
Configuring syslog-ng PE	11
Configuring the syslog-ng PE server	11
Configuring syslog-ng PE clients	12
Testing what you have done	14
Summary	15
About us	16
Contacting us	16
Technical support resources	16

Introduction

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 Premium Edition (syslog-ng PE) to send your log messages in an encrypted connection. Installing syslog-ng PE is not covered, for details, see "[Installing syslog-ng PE](#)" in the [Administration Guide](#).

The tutorial is organized as follows:

- [Creating self-signed certificates](#) 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 PE, as it is independent from the logging application you use.
- [Configuring syslog-ng PE](#) describes how to configure syslog-ng PE on your clients and your logserver.
- [Testing what you have done](#) gives you tips on how to test your configuration to make sure it is really working.

NOTE: Some of the command line examples in this tutorial are quite long. You might need to scroll the example to read the whole example.

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 PE, 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 PE.

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.

Creating a CA

This section describes how to create a CA.

NOTE: Some of the command line examples in this section are quite long. You might need to scroll the example to read the whole example.

To create a CA

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

```
mkdir CA
cd CA
```

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
```

3. Copy `openssl.cnf` 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
```

4. Edit `openssl.cnf` in the current directory:

```
vi openssl.cnf
```

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
```

6. As a last step, generate the certificate for the CA:

```
openssl req -new -x509 -keyout private/cakey.pem -newkey rsa:4096 -out
cacert.pem -days 365 -config openssl.cnf
```

NOTE: By including and customizing the `-newkey rsa:<key size>` element in your command line, you can set the key size that is compliant with your organization policy.

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]:Mycompany
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:Example
Email Address []:example@mycompany.com
```

Creating a server certificate

This section describes how to create a server certificate.

NOTE: Some of the command line examples in this section are quite long. You might need to scroll the example to read the whole example.

To create a server certificate

1. The next step is to create and sign a certificate for your syslog-ng PE server. The common name should contain the FQDN or IP address of your server, and the email address should be left blank.

```
openssl req -nodes -new -x509 -keyout serverkey.pem -newkey rsa:4096 -out
serverreq.pem -days 365 -config openssl.cnf
```

NOTE: By including and customizing the `-newkey rsa:<key size>` element in your command line, you can set the key size that is compliant with your organization policy.

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]:Mycompany
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:172.16.177.147
```

```

Email Address []:
example@linux-modi:~/CA> openssl x509 -x509toreq -in serverreq.pem -
signkey serverkey.pem -out tmp.pem
Getting request Private Key
Generating certificate request
example@linux-modi:~/CA> openssl ca -config openssl.cnf -policy policy_
anything -out servercert.pem -infile 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      = Mycompany
        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

```

3. Enter the following:

```
rm tmp.pem
```

Creating a client certificate

This section describes how to create a client certificate.

NOTE: Some of the command line examples in this section are quite long. You might need to scroll the example to read the whole example.

To create a client certificate

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 -newkey rsa:4096 -out clientreq.pem -days 365 -config openssl.cnf
```

NOTE: By including and customizing the `-newkey rsa:<key size>` element in your command line, you can set the key size that is compliant with your organization policy.

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]:Mycompany
Organizational Unit Name (eg, section) []:.
Common Name (e.g. server FQDN or YOUR name) []:172.16.177.129
Email Address []:

example@linux-modi:~/CA> openssl x509 -x509toreq -in clientreq.pem -
signkey clientkey.pem -out tmp.pem
Getting request Private Key
Generating certificate request
example@linux-modi:~/CA> openssl ca -config openssl.cnf -policy policy_
anything -out clientcert.pem -infile 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      = Mycompany
    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
```

3. Enter the following:

```
rm tmp.pem
```

Configuring syslog-ng PE

Once you are ready with generating CA, server and client certificates, copy them to the respective machines and configure syslog-ng PE 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 PE 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 PE can read them without modifying the related access rules.

Configuring the syslog-ng PE server

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.

To configure the syslog-ng PE server

1. As a first step, create two new directories under the syslog-ng PE configuration directory:

```
mkdir cert.d ca.d
```
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.
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
```
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 {
    network(
        ip(0.0.0.0)
        port(6514)
        transport("tls")
        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);
};
```

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

Configuring syslog-ng PE clients

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.

To configure syslog-ng PE clients

1. As a first step, create two new directories under the syslog-ng PE configuration directory:

```
mkdir cert.d ca.d
```
2. Copy `clientkey.pem` and `clientcert.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.
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
```

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 {
    network("172.16.177.147"
    port(6514)
    transport("tls")
    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); };
```

5. Finally, restart `syslog-ng` PE for the configuration changes to take effect.

Testing what you have done

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

To test the configuration

1. On the client side, enter the following command:
`logger "This is a test message"`
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 email address in the client and server certificates.
- For more information about TLS-related error messages, see "[Error messages](#)" in the [Administration Guide](#).

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 PE, visit the [syslog-ng project page](#).

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit <https://www.oneidentity.com/company/contact-us.aspx>.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at <https://support.oneidentity.com/>.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product