Part II. Installation and Configuration Guide

Table of Contents

Preface

This guide should describe all installation and administration issues of OpenCA. Some answers are perhaps in the FAQ but every detail which you can find in our docs about the istallation you can find here.

Chapter 3. Installation

1. Preparations

1.1. Software

OpenCA is not a complete monolithic system. It uses several software products from other developers of the Open Source community. The following things are used:

- Apache
- mod_ssl
- OpenSSL
- OpenLDAP
- Perl

We use a lot of different Perl modules. Beginning with OpenCA 0.9.2 we no longer install all foreign modules. This is the normal behaviour of every Open Source project. The following should give you an overview about the required modules. Please note that you must install at minimum the listed version because some earlier versions like for example Net::Server include serious bugs.

Table 3.1. External Perl modules

Module	Version	Comment
Authen::SASL	2.04	required by Net::LDAP for SASL authentication - if you do not use SASL then you do not need it
CGI::Session	3.95	required for our own session handling
Convert::ASN1	0.18	???
Digest::HMAC	1.01	required by Authen::SASL
Digest::MD5	2.24	this is usually part of Perl itself
Digest::SHA1	2.02	required by OpenCA itself
Encode::Unicode	???	required by OpenCA for the internationalization stuff
IO::Socket::SSL	0.92	???
IO::stringy	2.108	???
MIME::Base64	2.20	required for Base64 encoding and decoding
MIME::Lite	3.01	required for OpenCA mail handling
MIME-tools	5.411	required for OpenCA mail handling
MailTools	1.58	required for OpenCA mail handling
Net-Server	0.86	required for OpenCA daemon - the version is important

Module	Version	Comment
Parse::RecDescent	1.94	required by X500::DN
URI	1.23	???
X500::DN	0.28	we use a modified version here
XML::Twig	3.09	used for XML parsing **Warning** Please read the file `README` in the distribution of XML::Twig which you use really carefully. There are several incompatibilities with some versions of XML::Parser and expat. The used version of Perl is heavily important too.
libintl-perl	1.10	this is our interface for the i18n stuff
perl-ldap	0.28	Perl's LDAP interface

1.2. Hardware

OpenCA was tested on several softwarearchitectures but not on so many hardware architectures. Therefore we publish a list of used hardware. Please remember that OpenCA can be used on any system which support Apache, mod_ssl, OpenSSL and Perl. So if you have Unix box then it is usually possible to run an OpenCA on it.

- i386 with Linux, FreeBSD, OpenBSD and NetBSD
- UltraSparc with Solaris 8 and Linux
- PowerPC with AIX

2. Configure

OpenCA uses the usual Open Source method to **configure** the source. We only use **configure** to compile and install the software but we don't use **configure** for the configuration of the installed system. **configure** make some defaults settings but the real configuration is described in the post-install section.

We will describe the ideas and options in the next section grouped by such things like path settings, mail, web-server related stuff. If you don't understand an explanation then please contact `<openca-user@lists.sf.org>`. The install options are now lesser because we changed the installation process from 0.9.1 to 0.9.2 to get usable packages and better internationalization.

We don't document the general options of **configure** because it is not our job to document autoconf. We will only describe OpenCA specific options.

2.1. Host System Configuration

You should define the used system before you start configureing OpenCA itself. OpenCA must know several parameters about your system to work properly.

Table 3.2. Supported parameters for host configuration

Parameter	Description
`--with-openssl-prefix=`*`DIR`*	Usually OpenSSL is present on the most Unix systems because it is the best available Open Source crypto-toolkit. The problem is that several old distributions only include support for OpenSSL 0.9.6 but OpenCA needs version 0.9.7. If you install an OpenSSL from source then it installs in `/usr/local/ssl`. This is the directory which you must specify. If you system already includes a proper version then you have not to use this option or you can enter `/usr` on the most linux boxes.
`--with-openca-user=`*`ARG`*	OpenCA installs several files which should not be owned by the webserver user. Usually the owner can be root or special OpenCA user. It is recommended to use another user than root.
`--with-openca-group=`*`ARG`*	OpenCA installs several files which should not be owned by the webserver group. Usually the group can be root or special OpenCA group. It is recommended to use another user than root. If you install several CA you can setup a group openca or *`pki`* for *`example`*.

2.2. Filesystem paths

We have three different groups of paths - common stuff, prefixes for the different components of OpenCA and the paths for files of the webserver.

One path cannot be classified - `--with-module-prefix=`*`DIR`*. This path can be used to put all Perlmodules which OpenCA installs in one directory to be able to remove OpenCA from your system without any residues. It is also a good idea to use this option if you need different OpenCA installations with different versions of OpenCA on your system. Later versions of OpenCA can have different modules with different interfaces which are not backwards compatible.

2.2.1. Common Prefixes

OpenCA includes a directorystructure to store all relevant data in one central place. This place can be specified with `--with-openca-prefix`. This installation option is recommended for normal installations from the source code. Secure or not the most users want to install packages (e.g. RPM or DEB). Packages have the big advantage that you remove or add a software without any risks. In this case we have to support the package maintainers with configuration options to build packages which conform with the guidelines for the distros. Therefore you can use `--with-etc-prefix`, `--with-lib-prefix` and `--with-var-prefix` too.

2.2.2. Component Prefixes

Today there are six different components - ca, ra, ldap, pub, node and scep. Every component must have a different name to have distinguished configuration files and distinguished paths. All the names will be calculated automatically. You have only to edit these prefixes if you need a special configuration like a second RA on the same machine.

2.3. Webserver specific stuff

The webserver configuration is the most complex and most simple part of the configuration too. If you have single http-server for OpenCA then you only need four options to configure OpenCA for this server. If you have a full featured corporate portal then you can integrate this software seemlessly in the the server. Therefore you can configure a lot of details. So we hope you find a good tradeoff ;-)

2.3.1. Common server informations

Every webserver needs some basic informations. These informations are the hostname (`--with-web-host`), the user (`--with-httpd-user`) and the group of the server (`--with-httpd-group`). These are the rudimentary informations which OpenCA needs before you can start configuring the paths. The defaults are an empty hostname, `nobody` and `nogroup`.

The most trivial installation case is the default apache installation. In this case you have only to set `--with-httpd-fs-prefix` to the directory where your apache is. All other directories will be set automtically.

2.3.2. Filesystem Paths

The standard webserver doesn't use Apache's default installation. Therefore it is possible to configure every detail of the installation. The first splitting is into CGI (`--with-cgi-fs-prefix`) and HTDOCS (`--with-htdocs-fs-prefix`). The most test systems don't need the other options. They have only to know where the appropriate directories are.

Our software was designed for really big companies and organizations too. They have usually portals for their employees and customers. If you have to integrate an OpenCA interface into such a portal then there are good news for you - you don't have to edit paths and links by hand. You can configure the placement of CGI and HTDOCS area of every interface seperately. The options are `--with-(ca|ra|ldap|pub|node|scep)-(cgi|htdocs)-fs-prefix`). We think that more flexibility is not necessary. So if you think OpenCA is to unflexible then write a mail to us with your ideas.

2.3.3. URL Paths

OpenCA 0.9.1 supports a lot of options to configure the URLs like the filesystem paths. This is possible with OpenCA 0.9.2 too but it is deprecated to do this with **configure**. Please read the post-install section. It can happen that these options will be removed from configure.

2.4. Email

The mailoptions are deprecated too. Please read the post-install section to understand how to configure mail. Please don't use the configure option because they can be removed in the next releases.

2.5. Compiling features

You can enable three extra features for compilation and installation. SCEP and OCSP can be enabled because they are extra softwarepackages which can work independently from OpenCA but they are included in the distribution. The option `--enable-package-build` is used to support package maintainers. If it is activated all common parts of OpenCA are not installed automatically. This allows packagers to build seperate conflict free packages for every interfaces because all Perl modules and the common stuff can be put into seperate packages.

3. Installation

First run **make**, second run **make test** and then run the different install commands. **make ca** and **make ext** are the same like **make** You have the following install options for **make**.

- `install-ca`

- `install-common` is only interesting for package maintainers because they can install the common stuff seperately from the rest.
- `install-ext` is the same like install online. This install target is deprecated.
- `install-ldap`
- `install-node`
- `install-offline` installs ca and node
- `install-online` installs ra, ldap, pub, scep and node
- `install-pub`
- `install-ra`
- `install-scep`
- `install-docs`

OpenCA incudes a startup script for it's daemons. The script is named `OPENCADIR/etc/openca_start`. The script starts the XML cache and the main server loop of OpenCA. Please remember to run this script after every boot operation. It is recommended to integrate a script `openca` to the appropriate runlevel.

4. config.xml (for RPMs and DEBs too)

After the installation all necessary files are in the correct directories but there are hunderts of files called `*.template`. These files contain placeholders which can be configured in `OPENCADIR/etc/config.xml`. Before you start using OpenCA check this file and run **OPENCADIR/etc/configure_etc.sh**.

OPENCADIR/etc/configure_etc.sh loads `OPENCADIR/etc/config.xml` and creates the correct files. If you use packages from distribution then OPENCADIR is usually empty because they create a directory `/etc/openca`.

`config.xml` contains seven big sections which will be described first. Second we describe how to setup an installation with only one common area but two management interfaces. This is useful if you want to test the dataexchange.

4.1. Configuration sections of `config.xml`

4.1.1. General options

Here you have to define some options which are relevant for several interfaces. The ca locality, organization and country affects the distinguished name and the preconfiguration of the LDAP stuff. Nevertheless you should read the LDAP section too. The values which you enter are directly used for `l`, `o` and `c`.

The mailpart is used for the node and RA interface. The `sendmail` field defines the command which will be used to send mails. You must have a mailprogram with a sendmail interface (e.g. postfix). You can enter every program which works like **sendmail -n**. There are several people which like postfix more than sendmail and we don't like do decide which mailprogram is the best one. The option `send_mail_automatic` configures the node interface. If the value is *YES* then OpenCA sends all incoming mails during an import automatically. This can be nice but it is dangerous too if you make a mistake. The `service_mail_account` is used as `From` for all sended mails. Usually this should be something like `<Registration Authority &lt;pki@your_org.edu>>`. You can replace `>` by `&gt;` to but this is not required by the XML specification.

The last option `policy_link` defines a link to your policy. It is highly recommended to don't ignore this value. If

you have such a reference then you can modify the page request_success.html (add a hint) and the users can read all about the PKI at every time they want. If you have no such link then you receive dozens questions which are really simple but cost a lot of time and you have no base for your operations. Ok, I think I have not to explain the advantages of a policy here ...

4.1.2. web server configuration

Sometimes you need to run a CA on really unusual ports or you have to use https. It is also a little bit difficult for us to guess your correct hostname. Therefore you can specify these parameters in `config.xml`. The `httpd_port` should be the default port of the protocol and in this case it can be empty. If you need to run for example a http server on port 8080 then you have to use the option. Please remember to set the colons if you specify a port.

CRL distribution points (CDPs) can be specified extra. This is necessary because there are softwares which has problems with https in general or other softwares which try to download a CRL and before they start the download they want to check the CRL of the webserver but the CRL is not present (or why should somebody tries to download it :)) and so an endless loop starts - Microsoft CAPI is such a software.

If you setup a real CA then it is highly recommended to edit all files in `OPENCADIR/etc/openssl/extfiles` and `OPENCADIR/etc/openssl.cnf` too. Every certificate should contain at minimum two CDPs. It is best practice to have two http CDPs and two ldap CDPs. Such a solution allows fast migration and a good reliability.

4.1.3. ldap server configuration

Before you start working with OpenCA's LDAP code please be sure that your LDAP server knows the objectclasses `pkiUser`, `pkiCA` and `uniquelyIdentifiedUser`. The last objectclass was introduced by Entrust Technolgies to have a clean way to include serialnumbers into the subject of the certificate. Yes, it is proprietary but there is no other way to do it.

The following list is identical with the listof the list in the tech guide where you can find more informations about OpenCA's LDAP code and how to configure the details in the configuration files.

useLDAP	useLDAP If you set this option to "yes" then the LDAP code will be activated.
update_ldap_automatic	update_ldap_automatic If you want that the LDAP server will be updated during the import from a higher level of the hierarchy then you must set this option to YES.
ldap_host	ldap_host This is the hostname of your LDAP server.
ldap_port	ldap_port This is the port where your LDAP server listens.
ldaproot	ldaproot The bind DN of the user which OpenCA uses to add data to the server.
ldaprootpwd	ldaprootpwd The passphrase for OpenCA's ldap account.

4.1.4. database configuration

First you have to decide which database module you want to use. OpenCA supports two modules - one for DBM

files and one for SQL databases. `DB` activates support for DBM files and `DBI` activates the SQL support.

If you want to use SQL databases then you have to setup some additional parameters:

db_type	db_type This is the type of the DBD driver. We support Pg, MySQL, Oracle and DB2. If you need support for another database then please contact us.
db_name	db_name name of the database which OpenCA should use
db_host	db_host host of the database but sometimes the drivers don't need the host.
db_port	db_port same as for host
db_user	db_user the database user for OpenCA
db_passwd	db_passwd has not to be explained :)

4.1.5. module configuration

OpenCA has a mechanism to isolate the different interfaces from eachother to avoid conflicts especially double serialnumbers. The `module_shift` is the number of bits reserved for the IDs. You can use IDs from 0 to (2^`module_shift` - 1). 0 is the ID of the CA. All the other `_module_id`s must be in the scope of the module shift. Please be careful you cannot change the `module_shift` after the first definition.

Example 3.1. Module ID calculation

```
request serial ::= order number * 2^(MODULE_SHIFT) + MODULE_ID

Module ID    ::= 2
Module shift ::= 8

order number :   real serial
-------------:-----------------
      1      : 1 * 2^8 + 2 = 258
      2      : 2 * 2^8 + 2 = 514
      3      : 3 * 2^8 + 2 = 770
```

4.1.6. configuration of relative paths

The `_url_prefix` options define the exact positions in the webserver. This depends highly on the positions of the files in the filesystem but you can configure aliases in the `httpd.conf`. So OpenCA is fully flexible.

4.1.7. configuration of SCEP

SCEP is really simple to configure but please don't forget the access control configuration. It is strongly recommended to restrict the source addresses which can access the SCEP server.

SCEP_RA_KEY	SCEP_RA_KEY This is the PEM encoded private key of the SCEP interface. It has the same format like for mod_ssl.
SCEP_RA_CERT	SCEP_RA_CERT This is the PEM encoded certificate of the SCEP interface. It has the same format like for mod_ssl.
SCEP_RA_PASSWD	SCEP_RA_PASSWD This is the passphrase for the private key of the SCEP server. If you use a not encrypted private key (what is not recommended - then please set an empty string here. interface. It has the same format like for mod_ssl.

4.1.8. Dataexchange

The configuration of the dataexchange is really complex in OpenCA. You can find a description in the section about the configuration of the dataexchange (see Section 9, “Dataexchange”). If it is your first OpenCA installation then please use one of the templates. If you setup a production level PKI then you must understand this configuration before you use it. This is one of the most important configuration options to guarantee the security of the PKI.

4.2. How to setup two management interfaces on one server?

Before the explanations start please notice that it is important to first install the online components and then the offline components if you follow the instructions because the configuration of the offline components take care about the already configured online components.

Additonally please remember to set the configure option `--with-node-prefix` to different names during the configuration of the online and offline installation. This is important because otherwise you have only one management interface.

4.2.1. Online Components

The first installation uses only the normal steps - **./configure --with-node-prefix=online_node --with-your-options**, **make**, **make test**, **make install-online**, edit `OPENCADIR/etc/config.xml` and **OPENCADIR/etc/configure_etc.sh**. Please use your options to configure the software and use the hierarchy level *ra*.

4.2.2. Offline Components

The first step is the protection of the already installed configuration files. Please set no permissions to the later needed configuration files in `OPENCADIR/etc/servers`.

```
chmod 000 etc/servers/*.conf*
```

The first four steps are the same as for the online components except of the configuration options where you should change at minimum the hierarchy level to *CA*. So first you do **./configure --with-node-prefix=offline_node --with-your-options**, **make**, **make test**, **make install-offline** and edit `OPENCADIR/etc/config.xml`.

The next step is really important you have to edit the file `etc/configure_etc.sh`. The directory with the serverconfigurations is protected because of the first step but all the other directories should only contain configura-

tion files of the ca. Usually there should be the following directories:

```
/Test/OpenCA/etc/
/Test/OpenCA/lib/servers/offline_node
/Test/OpenCA/lib/servers/ca
/Test/htdocs/ca
/Test/htdocs/offline_node
```

After you fixed the script please run it. Now the offline components are installed and configured.

4.2.3. OPENCADIR/etc/menu.xml

`menu.xml` must be fixed manually because it includes only a basic configuration. You have to copy a complete `menu` section. The section must be renamed from *`offline_node`* to *`online_node`*. The `cgi_prefix` must be fixed too. Please verifies the menus with the `name`s *`ra`*, *`ldap`* and *`pub`* to use the correct links to the node interface. If all values are correct then you have now a working testinstallation with two management interfaces.