Work Breakdown Document

OpenDNSSEC Enforcer

René Post, rene@xpt.nl

December 15, 2010

Contents

1 ods-enforcer

Kind	Language	Origin
bin	C	signer

1.1 Description

Client that connects to the ods-enforcerd daemon and allows you to submit commands to it, responses from the daemon are printed back to commandline.

1.2 Responsibilities

1.2.1 Configuration

Read configuration file. Set flags based on configuration.

1.2.2 Connect to daemon

Establish a bi-directional channel to the daemon. Report failure to connect back to the user via stderr. Report successfull connect back to the user via stdout.

1.2.3 Send commands

Pass the commandline arguments to the daemon without processing.

1.2.4 Show command responses

Monitor availability of responses from the daemon. Pass output responses received from the daemon to stdout without processing. Pass error responses received from the daemon to stderr without processing. Terminate the program when an 'end of response' message is received.

1.3 Requires

log

1.4 Work

Hours	Confidence	Description
4	0.9	Copy whole program from ods-signer.
8	0.7	Refactor code to remove signer specific stuff leaving generic connection code.
8	0.7	Add enforcer specific code in a separate module that uses the generic connection code.

2 ods-enforcerd

Kind	Language	Origin
bin	C	new

2.1 Description

Key and signing policy enforcer deals with key rollover and key generation. It is a daemon that reads the policy associated with a zone and then performs the key management according to that policy. The enforcer generates configuration files for the ods-signer to perform the actual signing of RR sets. The ods-enforcer client can connect to this daemon to initiate enforcer commands.

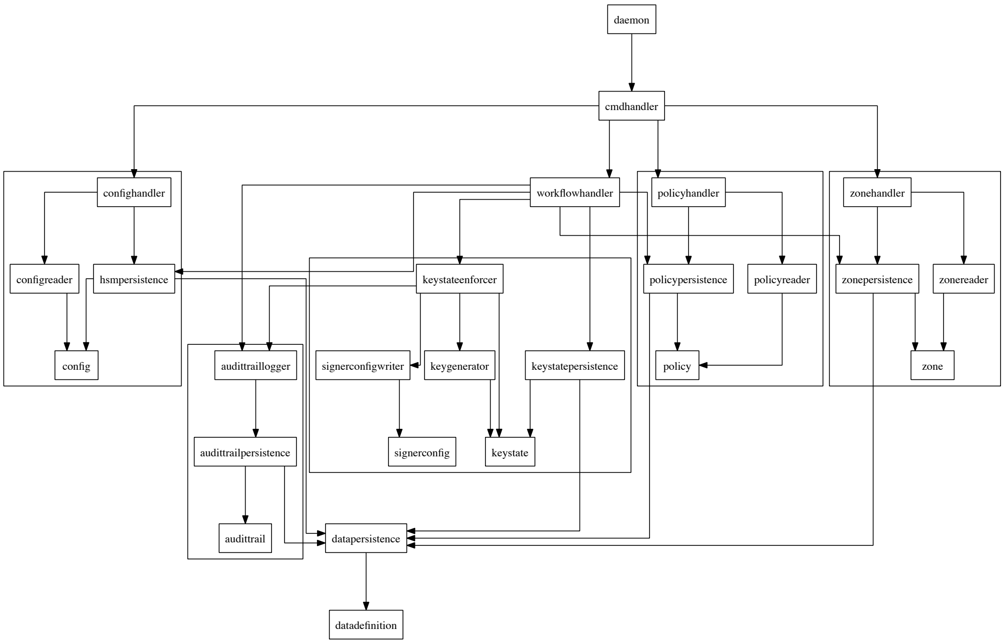

Figure 1: Enforcer Dependencies

2.2 Requires

daemon
cmdhandler
workflowhandler
config
configreader
hsmpersistence
confighandler
policy
policyreader
policypersistence

policyhandler
zone
zonereader
zonepersistence
zonehandler
keystate
keystatepersistence
keystateenforcer
keygenerator
singerconfig
signerconfigwriter
audittrail
audittrailpersistence
audittraillogger
datapersistence
datadefinition
privdrop
log

2.3 Work

Hours	Confidence	Description
8	0.9	Setup skeleton and hookup with the targets that actually implement the functionality.

3 daemon

Kind	Language	Origin
lib	C	signer

3.1 Description

Code that makes the ods-enforcerd actually behave as a proper daemon process.

3.2 Responsibilities

3.2.1 Daemon commandline options

Setup the commandline options supported by the daemon. Proces the commandline during startup and retrieve the options that were set on the commandline.

3.2.2 Daemon

Setup signal handlers appropriate for the daemon operation like SIGHUP, SIGTERM to reload config and terminate respectively. Properly handle daemon mode by detaching from the console. Support being started in interactive mode, i.e. not going into daemon mode.

3.3 Requires

getopt
log

3.4 Work

Hours	Confidence	Description
4	0.7	Daemon commandline options, remove signer specific code.
8	0.6	Daemon, remove signer specific code from the daemon code.

4 cmdhandler

Kind	Language	Origin
lib	C	signer

4.1 Description

Command handler listening for commands coming in from an endpoint it has setup. A client program can connect to the endpoint and send commands to the command handler.

4.2 Responsibilities

4.2.1 Command connection endpoint

Setup an endpoint for a client to connect to. Listen on the endpoint for incoming connections. Cleanup endpoint on termination. Accept connections on the endpoint. Kill other active connections when a new connection is established.

4.2.2 Command processing

Wait for a command to be send and wait for the 'end of command' indicator before actually starting processing If another command is currently busy, respond with 'command already active error' but keep the connection open to allow the active command to output results Whenever a running command reports a (partial) respons, send it straight back to the client via the endpoint

4.2.3 Command handling

Process a command until it is complete When a command is generating a stream of output data, stream that directly back to the user. After a command is finished send the 'end of response' indicator.

4.2.4 Enforcer specific command handling

Setup usage text and command options for the cmdhandler to return to the client Handle actual commands send to the enforcer Command handling only can control the workflow by modifying the persistent state of zones, keystates, policies etc.

4.3 Requires

log

4.4 Work

Hours	Confidence	Description
8	0.8	Remove signer specific code and create a generic cmdhandler.
2	0.7	Command connection endpoint, adapt to enforcer requirements.
8	0.7	Command processing.
8	0.8	Command handling, allow the command handler to be hookable with callbacks for hooking up actual commands.
8	0.5	Enforcer specific command handling.

5 workflowhandler

Kind	Language	Origin
lib	C++	new

5.1 Description

During normal operation the workflowhandler will actually be monitoring workflows that are currently running. The idea is to have no persistent state for the workflow itself, but that it will look at policies, zones, keystates and goals and decides what to do based on that information. The cmdhandler can only influence the workflowhandler by changing persistent data and triggering the workflowhandler.

5.2 Responsibilities

5.2.1 Handle cmdhandler trigger

Wakeup normally and decide task to perform.

5.2.2 Decide task to perform

Based on the persisten information at hand decide the task to perform. Select the most pressing task first when there are more that 1 task to choose from.

5.2.3 Perform tasks

Perform a task deduced from the current persistent state of the system. This can be something like generating a key, introducing new keys for a zone etc.

5.2.4 Deduce next wakeup

Based on the analysis of all current workflows, decide when to wakeup to perform another task. Don't persiste the wakeup, on a restart of the enforcer the deduction will generate a new wakeup time.

5.3 Requires

hsmpersistence
policypersistence
zonepersistence
keystatepersistence
audittrailpersistence
keystateenforcer
log

5.4 Work

Hours	Confidence	Description
4	0.7	Handle cmdhandler trigger
16	0.4	Decide task to perform
16	0.4	Perform tasks
8	0.7	Deduce next wakeup

6 config

Kind	Language	Origin
lib	C++	new

6.1 Description

Contains class declarations that represent the contents of the conf.xml global configuration file.

6.2 Responsibilities

6.2.1 Data Declaration

For every distinguishable separate element of data from the global configuration file, introduce a class declaration.

6.3 Requires

log

6.4 Work

Hours	Confidence	Description
8	0.9	Create data declarations.

7 configreader

Kind	Language	Origin
lib	C++	new

7.1 Description

Reads the conf.xml file into config classes.

7.2 Responsibilities

7.2.1 Read configuration

Use libxml2 to read the conf.xml file.

7.3 Requires

config
log
libxml2

7.4 Work

Hours	Confidence	Description
12	0.7	Read the configuration from the conf.xml file into the config classes.

8 hsmpersistence

Kind	Language	Origin
lib	C++	new

8.1 Description

Update the database with a HSM entry read from the config file.

8.2 Responsibilities

8.2.1 Store HSM entry

Store a HSM entry in the database.

8.2.2 Delete HSM entry

Delete a HSM entry from the database.

8.3 Requires

config
datapersistence
log

8.4 Work

Hours	Confidence	Description
12	0.7	Write a HSM entry into the database via datapersistence.

9 confighandler

Kind	Language	Origin
lib	C++	new

9.1 Description

Uses the config reader to read in the configuration and then uses hsm persistence to update the list of active hsms in the database.

9.2 Responsibilities

9.2.1 Config Processing

Load configuration at startup. Reload configuration when triggered by a command. Reload config when triggered by a SIGHUP signal. Read configuration file using configreader. Interpret the configuration file contents and update the database accordingly. Verify sanity of configuration before actually applying the results.

9.2.2 HSM List Update

Update the list of hsm in the database with the contents of the configuration. Removes hsm records from the database that are no longer present in the configuration.

9.3 Requires

configreader
hsmpersistence
config
log

9.4 Work

Hours	Confidence	Description
16	0.7	Implement config processing.
8	0.6	Implement HSM list update.

10 policy

Kind	Language	Origin
lib	C++	new

10.1 Description

Classes for the policy information used by the enforcer.

10.2 Responsibilities

10.2.1 Declare Policy Classes

Declare the classes to hold policy information.

10.3 Requires

policyreader
policypersistence
log

10.4 Work

Hours	Confidence	Description
8	0.9	Create data declarations.

11 policyreader

Kind	Language	Origin
lib	C++	new

11.1 Description

Allows the policy to be read from the kasp.xml file.

11.2 Responsibilities

11.2.1 Policy Loading

Read the policy from the kasp.xml file into the policy classes.

11.3 Requires

log
libxml2

11.4 Work

Hours	Confidence	Description
12	0.7	Read policy objects from the kasp.xml file.

12 policypersistence

Kind	Language	Origin
lib	C++	new

12.1 Description

Allows the policy information to be persisted to the database. Takes care of turning policy objects into persistent data and vice versa.

12.2 Responsibilities

12.2.1 Persist Policies

Maps an object to (1 or more) records in the database. Handles reading and writing of the policy.

12.3 Requires

datapersistence
log

12.4 Work

Hours	Confidence	Description
12	0.7	Implement code to write a policy into the database.

13 policyhandler

Kind	Language	Origin
lib	C++	new

13.1 Description

Implements commands to import policies from the kasp.xml configuration file into the system.

13.2 Responsibilities

13.2.1 Kasp Import

Use the policyreader to read policies from kasp.xml and then persist them into the database via policypersistence. Verify that the imported policy is sane.

13.3 Requires

policy
policyreader
policypersistence
log

13.4 Work

Hours	Confidence	Description
16	0.8	Perform a key and signing policy import

14 zone

Kind	Language	Origin
lib	C++	new

14.1 Description

Contains class declarations that represents the zone data that is used by the enforcer.

14.2 Responsibilities

14.2.1 Data Declaration

For every distinguishable separate element of data from the global configuration file, introduce a class declaration.

14.3 Requires

log

14.4 Work

Hours	Confidence	Description
8	0.9	Create data declarations.

15 zonereader

Kind	Language	Origin
lib	C++	new

15.1 Description

Reads the zonelist.xml.

15.2 Responsibilities

15.2.1 Zonelist Loading

Loads zone objects from the zonelist.xml file.

15.3 Requires

log
libmxml2

15.4 Work

Hours	Confidence	Description
12	0.7	Read zone objects from the zonelist.xml file.

16 zonepersistence

Kind	Language	Origin
lib	C++	new

16.1 Description

Allows the zone information to be persisted to the database. Takes care of turning zone objects into persistent data and vice versa.

16.2 Responsibilities

16.2.1 Persist Zone

Maps an object to (1 or more) records in the database.

16.3 Requires

datapersistence
log

16.4 Work

Hours	Confidence	Description
12	0.7	Implement code to write a zone into the database.

17 zonehandler

Kind	Language	Origin
lib	C++	new

17.1 Description

17.2 Responsibilities

17.2.1 Zone Import

Read zone from zonelist.xml file using zonereader. Persist zones in the database using zonepersistence.

17.2.2 Zone Export

Write all zones to a zonelist.xml file

17.3 Requires

zonereader
zonepersistence
zone
log

17.4 Work

Hours	Confidence	Description
8	0.6	Implement zone import
8	0.6	Implement zone export

18 keystate

Kind	Language	Origin
lib	C++	new

18.1 Description

Classes for the keystate information used by the keystateenforcer.

18.2 Responsibilities

18.2.1 Declare keystate classes

Declare the classes for holding keystate information.

18.3 Requires

keystatepersistence
log

18.4 Work

Hours	Confidence	Description
8	0.9	Create data declarations.

19 keystatepersistence

Kind	Language	Origin
lib	C++	new

19.1 Description

Allows the key state in the key state enforcer to be persisted to the database. Takes care of turning key state objects into persistent data and vice versa.

19.2 Responsibilities

19.2.1 persists keystate

maps an object to (1 or more) records in the database

19.3 Requires

log

19.4 Work

Hours	Confidence	Description
12	0.7	Implement code to write a keystate into the database.

20 keystateenforcer

Kind	Language	Origin
lib	C++	new

20.1 Description

State machine that is configured with keys for zones and the goals for the keys based on the policies configured for a specific zone. The machine is then provided with the current time and events that may have occured and then asked to determine the transitions that are allowed and the actions that need to be taken by the enforcer.

20.2 Responsibilities

20.2.1 Transition Rules

A mathematical formalization of the states the DNSKEY, RRSIG, and DS records associated with a key can go through. The key state enforcer can tell the higher level layers which state transitions are currently allowed for the DNSKEY,RRSIG and DS records associated with a key.

20.3 Requires

keystate
log

20.4 Work

Hours	Confidence	Description
16	0.7	Update to current state of the key state document.
8	0.6	Add functionality to allow a key state to persist itself.

21 keygenerator

Kind	Language	Origin
lib	C++	new

21.1 Description

Generate keypairs using the HSM.

21.2 Responsibilities

21.2.1 Generate Keys

Generate a keypair.

21.3 Requires

keystate
log
libhsm

21.4 Work

Hours	Confidence	Description
12	0.81	Generate a key in the HSM and create keystate information based on it.

22 signerconfig

Kind	Language	Origin
lib	C++	new

22.1 Description

Specification of data needed in order to be able to write a singer configuration to an XML file.

22.2 Responsibilities

22.2.1 Signer Configuration Declaration

Declare classes associated with signer configuration.

22.3 Requires

log

22.4 Work

Hours	Confidence	Description
8	0.7	Signer configuration declaration.

23 signerconfigwriter

Kind	Language	Origin
lib	C++	new

23.1 Description

"Writes signer configuration (one per zone) to the signerconf directory."

23.2 Responsibilities

23.2.1 gather signer information

access the data objects to collect information for the signer configuration

23.2.2 Write configuration

Write the configuration to a file in the singerconf directory Use a name that is deterministically derived from the zone name

23.3 Requires

signerconfig
log

23.4 Work

Hours	Confidence	Description
16	0.7	Write a signer configuration to the database.

24 audittrail

Kind	Language	Origin
lib	C++	new

24.1 Description

Specification of data needed in order to be able to reconstruct the actions the enforcer has performed.

24.2 Responsibilities

24.2.1 Audit Trail Declaration

Declare classes associated with audit trail information

24.3 Requires

log

24.4 Work

Hours	Confidence	Description
16	0.7	Audit trail declaration.

25 audittrailpersistence

Kind	Language	Origin
lib	C++	new

25.1 Description

Store an audit trail entry into the database.

25.2 Responsibilities

25.2.1 Persist Audit Trail

Write audit trail entries into the database. Audit trails always include denormalized data so they can be evaluated separately from the database. Don't rely on the number of record in the database, data archiving can empty the audit trail tables.

25.3 Requires

datapersistence
log

25.4 Work

Hours	Confidence	Description
8	0.6	Persist audit trail.

26 audittraillogger

Kind	Language	Origin
lib	C++	new

26.1 Description

26.2 Responsibilities

26.2.1 Persist Audit Trail Entries

Based on the information passed to this lib, contruct appropriate audit trail entries and use audittrailpersistence to log them in a database.

26.3 Requires

audittrailpersistence
audittrail
log

26.4 Work

Hours	Confidence	Description
24	0.6	Implement functions for logging specific audit trail entries

27 datapersistence

Kind	Language	Origin
lib	C++	new

27.1 Description

Data access library for the enforcer persistent data. Classes representing the data access that is needed for storing, querying and retrieving the enforcer data from the database.

27.2 Responsibilities

27.2.1 Data Abstraction

All SQL queries are done inside the datamodel classes make sure universally unique identification of objects is handled correctly

27.2.2 dynamic loading

Refactor data persistence into an interface with 2 implementations for MySQL and SQLite Turn the separate implementations into dynamically loaded libaries. Dynamically load either MySQL or SQLite driver depending on the database configured.

27.3 Requires

MySQL
SQLite
log

27.4 Work

Hours	Confidence	Description
60	0.6	Implement data abstraction
20	0.8	implement dynamic loading

28 datadefinition

Kind	Language	Origin
datadef	SQL	enforcer

28.1 Description

Definition of all the tables and fields in the enforcer database.

28.2 Responsibilities

28.2.1 table creation

create the tables for the enforcer and fill it with initial data

28.3 Work

Hours	Confidence	Description
80	0.5	Update table creation definitions to reflect the changes needed for storing new data associated with keystates
20	0.7	update table creation definitions to remove the meta-data tables from the database

29 privdrop

Kind	Language	Origin
lib	C	signer

29.1 Description

Allows a program to drop root privileges and run as a less privileged user or group.

29.2 Responsibilities

29.2.1 Drop Privileges

Drop privileges to a less privileged user or group

29.3 Requires

log

30 log

Kind	Language	Origin
lib	C	signer

30.1 Description

Wrapper around syslog that allows differentiated logging of errors, warnings and information.

30.2 Responsibilities

30.2.1 Wrap syslog

Just wrap syslog with a simple library of reusable logging code.

30.3 Requires

syslog

31 Work Overview

Target	Description	Hours	Confidence
ods-enforcer	Copy whole program from ods-signer.	4	0.9
ods-enforcer	Refactor code to remove signer specific stuff leaving generic connection code.	8	0.7
ods-enforcer	Add enforcer specific code in a separate module that uses the generic connection code.	8	0.7
ods-enforcerd	Setup skeleton and hookup with the targets that actually implement the functionality.	8	0.9
daemon	Daemon commandline options, remove signer specific code.	4	0.7
daemon	Daemon, remove signer specific code from the daemon code.	8	0.6
cmdhandler	Remove signer specific code and create a generic cmdhandler.	8	0.8
cmdhandler	Command connection endpoint, adapt to enforcer requirements.	2	0.7
cmdhandler	Command processing.	8	0.7
cmdhandler	Command handling, allow the command handler to be hookable with callbacks for hooking up actual commands.	8	0.8
cmdhandler	Enforcer specific command handling.	8	0.5
workflowhandler	Handle cmdhandler trigger	4	0.7
workflowhandler	Decide task to perform	16	0.4
workflowhandler	Perform tasks	16	0.4
workflowhandler	Deduce next wakeup	8	0.7
config	Create data declarations.	8	0.9
configreader	Read the configuration from the conf.xml file into the config classes.	12	0.7
hsmpersistence	Write a HSM entry into the database via datapersistence.	12	0.7
confighandler	Implement config processing.	16	0.7
confighandler	Implement HSM list update.	8	0.6
policy	Create data declarations.	8	0.9
policyreader	Read policy objects from the kasp.xml file.	12	0.7
policypersistence	Implement code to write a policy into the database.	12	0.7
policyhandler	Perform a key and signing policy import	16	0.8
zone	Create data declarations.	8	0.9
zonereader	Read zone objects from the zonelist.xml file.	12	0.7
zonepersistence	Implement code to write a zone into the database.	12	0.7
zonehandler	Implement zone import	8	0.6
zonehandler	Implement zone export	8	0.6
keystate	Create data declarations.	8	0.9
keystatepersistence	Implement code to write a keystate into the database.	12	0.7
keystateenforcer	Update to current state of the key state document.	16	0.7
keystateenforcer	Add functionality to allow a key state to persist itself.	8	0.6

Target	Description	Hours	Confidence
keygenerator	Generate a key in the HSM and create keystate information based on it.	12	0.81
signerconfig	Signer configuration declaration.	8	0.7
signerconfigwriter	Write a signer configuration to the database.	16	0.7
audittrail	Audit trail declaration.	16	0.7
audittrailpersistence	Persist audit trail.	8	0.6
audittraillogger	Implement functions for logging specific audit trail entries	24	0.6
datapersistence	Implement data abstraction	60	0.6
datapersistence	implement dynamic loading	20	0.8
datadefinition	Update table creation definitions to reflect the changes needed for storing new data associated with keystates	80	0.5
datadefinition	update table creation definitions to remove the meta-data tables from the database	20	0.7
	total	578	0.66