Current status of work on OpenDNSSEC 2.0

February 2014

History

- The 1.4 enforcer component was re-written for the 2.0 release as ‘enforcer-ng' with 2 major design changes
  - A task based daemon (as used in the signer) for scalability
  - Generic key rollover engine to support multiple rollover mechanisms, policy and algorithm rollover, CSKs, sign/unsigned transition, etc.
- A further requirement was that it should be a ‘drop in replacement’ for 1.4 wherever practicable.
- An 2.0.0a3 release with core functionality was made in June 2012.
- Work on the project has been slow but steady since then due to limited resources.

Recent Development work

Development work re-started in ernest on 2.0 in the second half of 2103 and since then progress has been made with the following functionality:

- Zone handling
  - Compatibly with 1.4 commands for single and bulk adds/deletes
  - Change to enforcer/signer interface to exchange zone information
- Extensions and compatibility changes to `key list` and `rollover list` implemented
- Full policy handling implemented (`policy import/export/purge`)
- Key `backup` commands implemeted
- Review and enhancements to daemon lifecycle, `update` commands and parameter reload
- Port of C implementation of `ods-kaspcheck` fro 1.4 to 2.0
- A number of low level scheduling issues have been resolved
- Manual rollover has been implemented and tested
- Basic regression tests have been enabled

An analysis of the existing 2.0 API has been performed, and the last few months have seen a significant increase in knowledge of the current codebase among the development team.

Initial benchmarking measurements have also recently been performed and are available in a separate document.

A 2.0.0a4 snapshot release is planned very shortly.

Challenges that remain

This section attempts to highlight the most important tasks remaining to reach beta status for 2.0. It is not a comprehensive list, but tries to convey the current status in terms of meeting the release requirements.

1. **The API is not yet complete, for example:**

   A. 5 commands are not yet implemented or are implemented incorrectly

   B. 7 commands do not have all options available in 1.4 (13 options still to be implemented)

   C. In 1.4 the command line utilities provide a return code that indicates success of failure of the operation, greatly simplifying scripting of these commands. The current implementation of the 2.0 command function handler has no explicit error handling mechanism, all commands that find a matching command method return 0 regardless of the outcome. A task to re-work the command handler to provide status return codes is in the development plan.

D. The logging output is significantly different to 1.4 (specific text, events logged, stdout/stderr vs syslog, and logging levels). A task to review the current logging to provide optimal output and more compatibility with 1.4 is in the development plan.

   Improvements to the 1.4 API was high on the user survey last year. 2.0 has not yet attained parity with the 1.4 API and this is an excellent opportunity to address some of the issues with the API, while maintaining compatibility where possible.

2. **Database layer**

   The database layer code used in 2.0 is not commented, there is no design documentation and its underlying design is not well understood by any active member of the team. It is quite abstracted in areas and can be difficult to debug. Efforts are underway to remedy these problems but the contractor that originally developed this layer is no longer funded to work on the project and can only help in his spare time.

   There are also concerns about performance of the database in general….

3. **Quality and testing**

   A. The new rollover engine has been manually tested in isolation, but manual test coverage of the rest of the system appears quite low. Several basic, critical issues have been found and fixed in the last few months.

   B. Due to changes in the basic workflow compared to 1.4 the enforcer regression tests require re-working to be enabled for 2.0. This is time consuming and we do not dedicated testing resources. Currently only 11 of 71 total regression tests are running on 2.0, of these only 2 of 30 enforcer ones are running.

   C. The initial effort in development was directed at producing a functional system to support the new daemon and rollover engine.
      1. It appears that much of the detailed and operational experience gained in 1.4 has not bend ported to 2.0. For example numerous sanity checks, optimisations, bug fixes, etc. that are in 1.4 are not yet implemented in 2.0.
      2. Some code quality and consistency issues have also been observed particularly in the API layer.
      3. To the authors knowledge no code analysis or profiling has yet been done on the 2.0 code base.

      For all these reasons a full code review (with reference to the 1.4 implementation) is recommended before a beta release.

   D. The upgrade path from 1.4 to 2.0 has not yet been fully tested

4. **Operational impression**

   This section notes some functional difference between 1.4 and 2.0 that will be apparent to users and discusses the impact/mitigation.

   A. In 1.4 it was possible to predict the exact time of the next key state transition for each key on each zone, and the new state to which the key will transition and this was reported to the user via a `key list` command. Due to the nature of the generic rollover engine in 2.0 it is no longer possible to do this. In 2.0 it is only possible to specify the time of the earliest possible transition of any key for a zone. This needs to be fully documented for users.

   B. Due to the nature of the generic rollover engine there are slight timing differences in key rollovers for the default rollover mechanisms compared to 1.4. For example, there is an additional delay for the KSK for a new zone reaching the `ds-seen` state and then when the `ds-seen` command is given for the key to become active. This could be an issue for users where the publication time of new, signed zones is business critical. A task to investigate if this can be mitigated for the default rollover path is in the development plan as is detailed documentation of any changes in timings compared to 1.4.

C. The generic rollover engine uses a new (fine grained) set of state information to manage keys. In order to present a compatible experience with 1.4 these states have been mapped back to the higher-level key states that 1.4 users are familiar with (generate, publish, ready, active, retire, dead). This is not completely straightforward, for example in the current implementation for the default rollover mechanisms this can result in what appear to be different key lifecycles compared to 1.4 including a time when a zone has no active keys! Again, a task to investigate this is on the development plan.

D. In 1.4, a manual step was required to progress a KSK through a full rollover (i.e. a `ds-seen` command had to be issued). In 2.0, the new rollover engine requires 3 manual steps:
   i) ds-submit
   ii) ds-seen
   iii) ds-retract

   Given that increased automation of DS handling in 1.4 was a high priority in the recent user survey this could unfortunately serve as a deterrent to some users adopting 2.0. A task to investigate how this can be mitigated via either special logic or increased automation in 2.0 is in the development plan.

E. The documentation of the detailed changes in 2.0 is still a work in progress: https://wiki.opendnssec.org/pages/viewpage.action?pageId=2621764

5. Benchmarking

Recent benchmarking has been performed, which provided a baseline for 1.4 performance and performed initial measurements on 2.0. This work highlighted several problems in 2.0 that appear to lead to non-optimal performance, some of which will be addressed before the release so the that benchmarking can be re-run. Please refer to the separate document for more details.