Testing DNS Adapters

Overview

Support for AXFR and IXFR is being introduced in OpenDNSSEC version 1.4. This makes handling the zone input and output a lot more complex.

Requirements

AXFR is specified in RFC 1034 and RFC 1035. If is updated by RFC 5936. IXFR is specified in RFC 1995. The NOTIFY mechanism is described in RFC 1996.

Input

AXFR is first mentioned in RFC 1034. OpenDNSSEC can send a query with a special QTYPE=AXFR to request a full zone transfer. AXFR is being requested over TCP, and the query is answered by a sequence of response messages. OpenDNSSEC can detect a change in the zone by comparing the local serial and the SERIAL field of the SOA, as described in RFC 1982. OpenDNSSEC uses a poll strategy, based on the REFRESH, RETRY and EXPIRE fields of the SOA.

- **[14-dns-input-adapter]** OpenDNSSEC must be able to request AXFR.
- **[14-dns-input-adapter]** OpenDNSSEC must be able to request IXFR.
- **[14-dns-ixfr-notimpl]** If IXFR is not implemented, OpenDNSSEC should fall back to requesting AXFR.
- **[14-dns-input-adapter]** If IXFR does not fit in a UDP packet, OpenDNSSEC must fall back to TCP.
- Detecting a change is done by querying for the SOA RR of the zone, and comparing the SERIAL field of the SOA RR with the local serial, RFC 1982 style.
- If the SERIAL of the SOA RR increments the local serial, a zone transfer must be requested.
- **[14-dns-refresh]** When a new zone is loaded, OpenDNSSEC needs to wait REFRESH seconds before checking with the primary for a new serial.
- **[14-dns-retry-expire]** If the request fails, a new check is being made after RETRY seconds.
- **[14-dns-retry-expire]** If no such check succeeds within EXPIRE seconds, OpenDNSSEC must stop serving the zone.
- **[14-dns-input-adapter]** If a NOTIFY message is received, a NOTIFY OK reply must be send and a zone transfer must be requested.

Output

- **[14-dns-output-adapter]** OpenDNSSEC must be able to serve AXFR.
- **[14-dns-output-adapter]** OpenDNSSEC must be able to serve IXFR.
- OpenDNSSEC may condense an IXFR response.
- **[14-dns-output-adapter]** If IXFR does not fit in a UDP packet, OpenDNSSEC must send a packet that indicates UDP would overflow.
- **[14-dns-output-adapter]** When IXFR is not available, OpenDNSSEC should fall back to AXFR.
- **[14-dns-output-adapter]** OpenDNSSEC must be able to answer SOA queries.
- **[14-dns-output-notify]** When OpenDNSSEC has updated the signed zone, it should send NOTIFY messages to the configured slave servers.
- **[14-dns-output-adapter]** When a slave server does not respond with a NOTIFY OK message, OpenDNSSEC should retry a reasonable number of times.
- **[14-dns-output-adapter]** Occluded names must be included in AXFR responses, while they should not appear in signed zonefiles.

TCP Management

- OpenDNSSEC should not block other activities when waiting for TCP data.
- OpenDNSSEC should support multiple connections.
- OpenDNSSEC should delay closing all outstanding client requests have been satisfied.
- OpenDNSSEC should close a dormant connection after an idle period of two minutes.

Access Control

- OpenDNSSEC should be able to have an ACL for zone transfer requests, responses and notify messages.
- Such packets must be able to be signed with TSIG.
- Such packets must be able to be TSIG verified.
- The ACL must be able to allow access based on Address, TSIG or both.

Test Environment

We need to be able to do zone transfers over UDP, TCP, IPv4, IPv6, from different masters, to different slaves, with and without TSIG. Therefore, our test environment needs to contain multiple machines:

- Two masters, MA and MB that are zone transfer servers (they will serve zone transfers to OpenDNSSEC).
- MA is a Bind9 server, and MB is NSD (which does not serve IXFR).
- An OpenDNSSEC instance, ODS, that will sign zones, retrieved from file or from one of the two masters through AXFR or IXFR.
- Two secondaries, SA and SB, that are zone transfer clients (they will request zone transfers from OpenDNSSEC through AXFR or IXFR).
- SA is a Bind9 server, and MB is NSD. We configure NSD so that it will not request IXFR.

Test Cases

One large zone

First, we start with a simple zone and configure it with DNS Input and File Output Adapters. This zone source of authority is at MA and will produce a signed zonefile on the OpenDNSSEC box. Focus in on the Input requirements, listed in RFCs. The zone should be large, to enforce that the sequence of response messages consists of more than one packet.

- OpenDNSSEC polls for a new zone by sending a SOA query to MA.
- OpenDNSSEC implements the REFRESH feature.
- OpenDNSSEC implements the RETRY feature.
- OpenDNSSEC implements the EXPIRE feature.
- OpenDNSSEC triggers a zone transfer request upon a NOTIFY from MA.
- OpenDNSSEC sends a NOTIFY OK upon a NOTIFY to MA.
- OpenDNSSEC refuses a NOTIFY that is not from MA.
- OpenDNSSEC refuses a NOTIFY that is not TSIG verified.
- OpenDNSSEC requests a zone transfer by sending an IXFR query to MA, with the current unsigned SOA RR in the Authority Section.
- OpenDNSSEC signs the zone transfer request with TSIG.
- If IXFR does not fit in a UDP packet, OpenDNSSEC must fall back to TCP.
- OpenDNSSEC correctly reconstructs the zone upon receiving an IXFR response.

Second, we start with a simple zone and configure it with File Input and DNS Output Adapters. This zone source of authority is at the OpenDNSSEC box and will be served to SA. Focus in on the Output requirements, listed in RFCs.

- OpenDNSSEC sends a NOTIFY to SA.
- OpenDNSSEC signs the NOTIFY with TSIG.
- OpenDNSSEC accepts a NOTIFY OK from SA.
- OpenDNSSEC drops other NOTIFY OK responses.
- OpenDNSSEC must be able to answer SOA queries.
- OpenDNSSEC refuses zone transfer requests not originating from SA.
- OpenDNSSEC refuses zone transfer requests that cannot be TSIG verified
- OpenDNSSEC sends an IXFR response to SA upon a IXFR request, if the requested SERIAL can be found.
- OpenDNSSEC sends an AXFR response to SA upon a IXFR request, if the requested SERIAL cannot be found.
- OpenDNSSEC sends an IXFR UDP Overflow response to SA upon a IXFR request, if the zone transfer does not fit into an UDP message.
- Occluded names are present in the AXFR/IXFR response.

If both tests succeed, we can retry with both DNS Input and Output Adapters.

Many zones

We define a set of zones with different source of authorities, different ACLs, different secondaries and thus different zone transfer strategies.
Note that the table below does not list all possibilities, but it should cover more than enough to go through all code paths.

MASTER	SLAVE	Inbound TSIG	Outbound TSIG
MA	ODS	-	n/a
MA	ODS	MD5	n/a
MA	ODS	SHA1	n/a
MA	ODS	SHA256	n/a
MB	ODS	-	n/a
MB	ODS	MD5	n/a
MB	ODS	SHA1	n/a
MB	ODS	SHA256	n/a
MA+MB	ODS	-	n/a
MA+MB	ODS	SHA1	n/a
ODS	SA	n/a	-
ODS	SA	n/a	MD5
ODS	SA	n/a	SHA1
ODS	SA	n/a	SHA256
ODS	SB	n/a	-
ODS	SB	n/a	MD5
ODS	SB	n/a	SHA1
ODS	SB	n/a	SHA256

ODS	SA+SB	n/a	-
ODS	SA+SB	n/a	SHA1
MA	SA+SB	-	SHA256
MB	SA+SB	MD5	-
MA	SA+SB	SHA1	MD5
MB	SA+SB	SHA256	SHA1

Focus is on that all zones are getting signed correctly. Also the following requirements should be checked:

- OpenDNSSEC falls back to AXFR if IXFR is not available (in case of MB).
- OpenDNSSEC correctly reconstructs the zone upon receiving an AXFR response (from MB).
- OpenDNSSEC should not block other activities when waiting for TCP data.
- OpenDNSSEC should support multiple connections.
- OpenDNSSEC should delay closing all outstanding client requests have been satisfied.
- OpenDNSSEC should close a dormant connection after an idle period of two minutes.
- When a slave server does not respond with a NOTIFY OK message, OpenDNSSEC should retry a reasonable number of times.