kasp.xml

kasp.xml (found by default in */etc/opendnssec)* is the file that defines policies used to sign zones. Each policy comprises a series of parameters that define the way the zone is signed. This section explains the parameters by referring to the example kasp.xml file supplied with the OpenDNSSEC distribution.

Specification of Date/time durations.

Elements of the kasp.xml file

Preamble

```
<?xml version="1.0" encoding="UTF-8"?>
```

Each XML file starts with a standard element "<?xml...". As with any XML file, comments are included between the delimiters "<!–" and "-->".

Policy Description

```
<KASP>
```

The enclosing element of the XML file is the element <KASP> which, with the closing element </KASP>, brackets one or more policies.

```
<Policy name="default">
    <Description>A default policy that will amaze you and your friends</Description>
```

Each policy is included in the <Policy>...</Policy> elements. Each policy has a "name" attribute giving the name of the policy. The name is used to link a policy and the zones signed using it; each policy must have a unique name. The policy named "default" is special, as it is associated with all zones that do not have a policy explicitly associated with them.

A policy can have a description associated with it. Unlike XML comments, the description can be understood by programs and may be used to document the policy, e.g. a future GUI may display a list of policies along with their description and ask you to select one for editing.

Signatures

The next section of the file is the Signatures section, which lists the parameters for the signatures created using the policy.

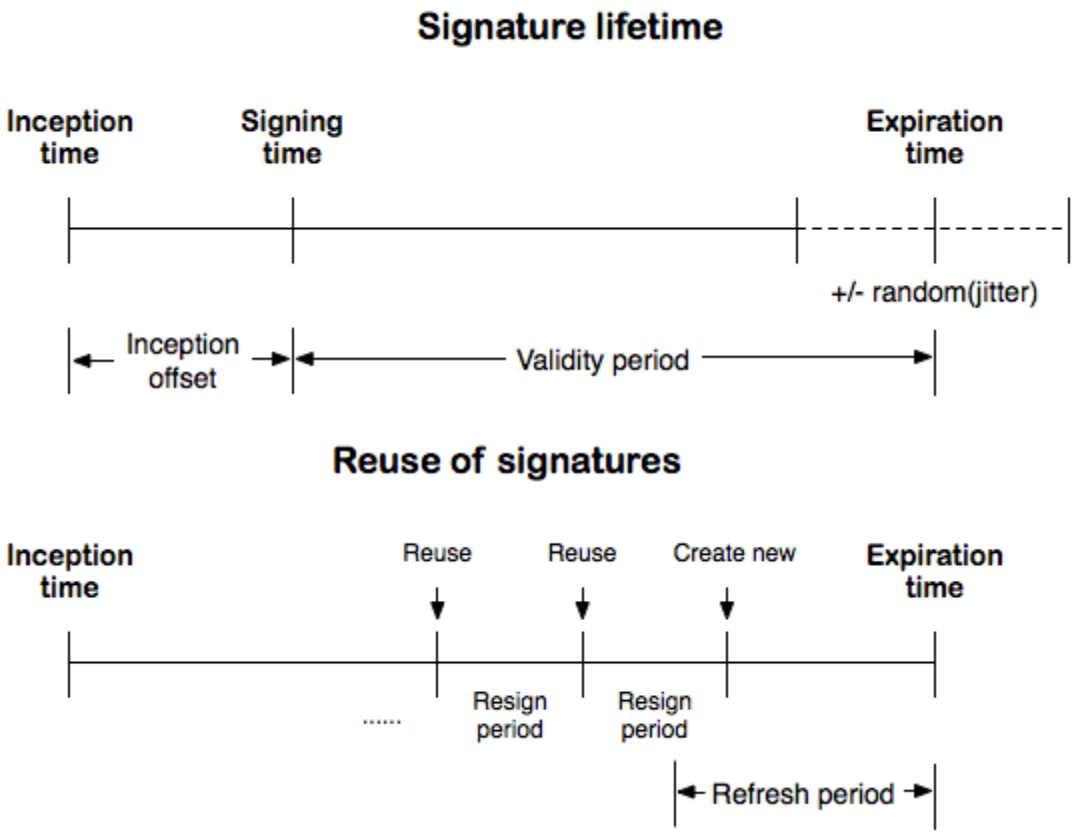
```
 <Signatures>
     <Resign>PT2H</Resign>
     <Refresh>P3D</Refresh>
     <Validity>
         <Default>P14D</Default>
         <Denial>P14D</Denial>
     </Validity>
     <Jitter>PT12H</Jitter>
     <InceptionOffset>PT300S</InceptionOffset>
</Signatures>
```

Here:

- <Resign> is the re-sign interval, which is the interval between runs of the Signer Engine.
- <Refresh> is the refresh interval, detailing when a signature should be refreshed. As signatures are typically valid for much longer than the interval between runs of the signer, there is no need to re-generate the signatures each time the signer is run if there is no change to the data being signed. The signature will be refreshed when the time until the signature expiration is closer than the refresh interval. Set it to zero if you want to refresh the signatures each re-sign interval.

- <Validity> groups two elements of information related to how long the signatures are valid for - <Default> is the validity interval for all RRSIG records *except* those related to NSEC or NSEC3 records. In this case, the validity period is given by the value in the <Denial> element.
- <Jitter> is the value added to or extracted from the expiration time of signatures to ensure that not all signatures expire at the same time. The actual value of the <Jitter> element is the **-j + r %2j**, where j is the jitter value and r a random duration, uniformly ranging between -j and j, is added to signature validity period to get the signature expiration time.
- <InceptionOffset> is a duration subtracted from the time at which a record is signed to give the start time of the record. This is required to allow for clock skew between the signing system and the system on which the signature is checked. Without it, the possibility exists that the checking system could retrieve a signature whose start time is later than the current time.

The relationship between these elements is shown below.



Authenticated Denial of Existence

Authenticated denial of existence - proving that domain names do not exist in the zone - is handled by the <Denial> section, as shown below:

```
<Denial>
    <NSEC3>
        <TTL>PT3600S</TTL>
        <OptOut/>
        <Resalt>P100D</Resalt>
        <Hash>
            <Algorithm>1</Algorithm>
            <Iterations>5</Iterations>
            <Salt length="8"/>
        </Hash>
    </NSEC3>
</Denial>
```

<Denial> includes one element, either <NSEC3> (as shown above) or <NSEC>.

NSEC3

<NSEC3> tells the signer to implement NSEC3 scheme for authenticated denial of existence (described in RFC 5155). The elements are:

- <TTL>, if present, this is the time-to-live value for the NSEC3PARAM resource records. If not present, PT0S (0) will be used as TTL. This will only affect the time-to-live value for the NSEC3PARAM resource records. The time-to-live value for NSEC3 records is set to the value of the SOA <Minimum>.
- <OptOut/>, if present, enable "opt out". This is an optimization that means that NSEC3 records are only created for authoritative data or for secure delegations; insecure delegations have no NSEC3 records. For zones where a majority of the entries are delegations that are not signed - typically TLDs during the take-up phase of DNSSEC - this reduces the number of DNSSEC records in the zone.
- <Resalt>, the interval between generating new salt values for the hashing algorithm.
- <Algorithm>, <Iterations> and <Salt> are parameters to the hash algorithm, described in RFC 5155.

NSEC

Should instead NSEC be used as the authenticated denial of existence scheme, the <Denial> element will contain the single element <NSEC/>.There are no other parameters.

Key Information

Parameters relating to keys can be found in the <Keys> section.

Common Parameters

The section starts with a number of parameters relating to both zone-signing keys (ZSK) and key-signing keys (KSK):

```
<Keys>
    <TTL>PT3600S</TTL>
    <RetireSafety>PT3600S</RetireSafety>
    <PublishSafety>PT3600S</PublishSafety>
    <ShareKeys/>
    <Purge>P14D</Purge>
```

- <TTL> is the time-to-live value for the DNSKEY resource records.
- <PublishSafety> and <RetireSafety> are the publish and retire safety margins for the keys. These intervals are safety margins added to calculated timing values to give some extra time to cover unforeseen events, *e.g.* in case external events prevent zone publication.
- If multiple zones are associated with a policy, the presence of <ShareKeys/> indicates that a key can be shared between zones. E.g. if you have 10 zones then you will only use one set of keys instead of 10 sets. This will save space in your HSM. If this tag is absent, keys are not shared between zones.
- If <Purge> is present, keys marked as dead will be automatically purged from the database after this interval.

Key-Signing Keys

Parameters for key-signing keys are held in the <KSK> section:

```
<KSK>
    <Algorithm length="2048">8</Algorithm>
    <Lifetime>P1Y</Lifetime>
    <Repository>softHSM</Repository>
    <ManualRollover/>
</KSK>
```

- <Algorithm> determines the algorithm used for the key (the numbers reserved for each algorithm can be found in the appropriate IANA registry).
- <Lifetime> determines how long the key is used for before it is rolled.
- <Repository> determines the location of the keys. Keys are stored in "repositories", which are defined in the conf.xml. In the example above, the key is stored in softHSM. The example configuration file distributed with OpenDNSSEC defines this as being the software emulation of an HSM distributed as part of OpenDNSSEC.
- <ManualRollover/> is an optional tag. This tag indicate that the key rollover will only be initiated on the command by the operator. There is still a second step for the KSK, where the key needs to be published to the parent before the rollover is completed. Read more in the chapter "Running OpenDNSSEC". The ZSK rollover will although be fully automatic if this tag is not present.

Standby Keys

Before version 2.0 the KSK section could have a <StandbyKeys> element. Key rollovers are a process that can be interrupted at any time in OpenDNSSEC 2.0 and therefore the notion of standby keys was dropped. The element is ignored but still accepted to ease migration.

Zone-Signing Keys

Parameters for zone-signing keys are held in the <ZSK> section, and have the same meaning as for the KSK:

```
<ZSK>
    <Algorithm length="1024">8</Algorithm>
    <Lifetime>P90D</Lifetime>
    <Repository>softHSM</Repository>
</ZSK>
```

The ZSK information completes the contents of the <Keys> section.

Zone Information

General information concerning the zones can be found in the <Zone> section:

```
<Zone>
    <PropagationDelay>PT9999S</PropagationDelay>
    <SOA>
        <TTL>PT3600S</TTL>
        <Minimum>PT3600S</Minimum>
        <Serial>unixtime</Serial>
    </SOA>
</Zone>
```

- <PropagationDelay> is the amount of time needed for information changes at the master server for the zone to work its way through to all the secondary nameservers.
- The <SOA> element gives values of parameters for the SOA record in the signed zone. The values are:
    - <TTL> - TTL of the SOA record.
    - <Minimum> - value for the SOA's "minimum" parameter.
    - <Serial> - the format of the serial number in the signed zone. This is one of:
        - counter - use an increasing counter (but use the serial from the unsigned zone if possible)
        - datecounter - use increasing counter in YYYYMMDDxx format (xx is incremented within each day)
        - unixtime - the serial number is set to the "Unix time" (seconds since 00:00 on 1 January 1970 (UTC)) at which the signer is run.
        - keep - keep the serial from the unsigned zone (do not resign unless it has been incremented)

These values will override values set for the SOA record in the input zone file and the serial in signed and unsigned zone is likely to go out of sync.

Parent Zone Information

If a DNSSEC zone is in a chain of trust, digest information about the KSKs used in the zone will be stored in DS records in the parent zone. To properly roll keys, timing information about the parent zone must be configured in the <Parent> section:

```
<Parent>
    <PropagationDelay>PT9999S</PropagationDelay>
    <DS>
        <TTL>PT3600S</TTL>
    </DS>
    <SOA>
        <TTL>PT3600S</TTL>
        <Minimum>PT3600S</Minimum>
    </SOA>
</Parent>
```

- <PropagationDelay> is the interval between the time a new KSK is published in the zone and the time that the DS record appears in the parent zone.
- The <DS> tag holds information about the DS record in the parent. It contains a single element, <TTL>, which should be set to the TTL of the DS record in the parent zone.
- <SOA> gives information about parameters of the parent's SOA record, used by KASP in its calculations. As before, <TTL> is the TTL of the SOA record and <Minimum> is the value of the "minimum" parameter.

This is the last section of the policy specification, so the next element is the policy closing tag. If there are any additional policies, they could be entered here, starting with <Policy> and ending with </Policy>. However, in this case there are no additional policies, so the file is ended by closing the </KASP> tag.