Hong Kong Internet Registration Corporation Limited

DNSSEC Enabling Guide

V1.1
30-06-17

Contents

Introduction 1

A. DNS Hosting Service DNSSEC Architecture and Design 2

Method 1: "Bump in the Wire" 2

Method 2: Master Server 3

Recommendation 3

Example Configuration 4

Basic Setup Steps 5

Consideration on protecting private key 6

Implementation Consideration 6

Maintenance Tasks 7

B. Enable DNSSEC Validation in DNS resolver for Internet Access Service and Testing 8

Other Information 10

Appendix A - DNSSEC Parameters 11

Introduction

Many companies likely have some level of digital security measures in place and that include antivirus or network security solutions. In fact, there is also a fundamental way to protect your business from cyber security threats through your website. That is where Domain Name System Security Extensions (DNSSEC) comes to play.

It is the intention of this document to present to those who wish to evaluate or even implement a DNSSEC Infrastructure:

- What is required and an estimation on it cost
- How to implement a simple infrastructure, with example
- Where to find further information on the operation and maintenance of this infrastructure

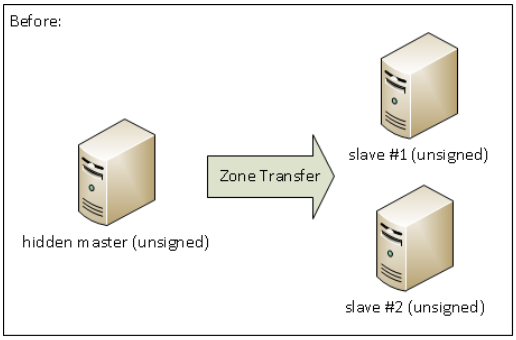
This document is target for an audience with a technical background on DNS. This document also applies to whoever needs to provide DNSSEC enabled DNS service eg. ISP, registrar and other hosting service company. If you are an ISP or evaluating/ plan to provide a DNSSEC enabled DNS service, please refer to Section A. If you are providing a DNS query service and want to enable DNSSEC validation, please refers to Section B.

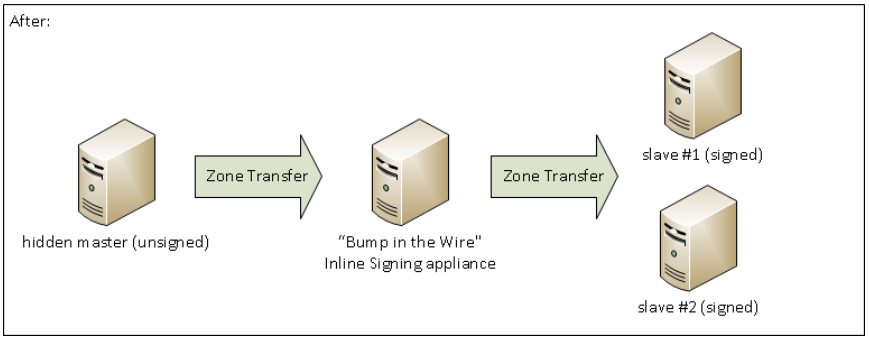
A. DNS Hosting Service DNSSEC Architecture and Design

There are 2 methods to serve your authoritative zones as DNSSEC signed zone. Below section will explain briefly and describe some pros and cons.

Method 1: "Bump in the Wire"

This method inserts a signing unit between master and slaves as show as below.





The good thing is:

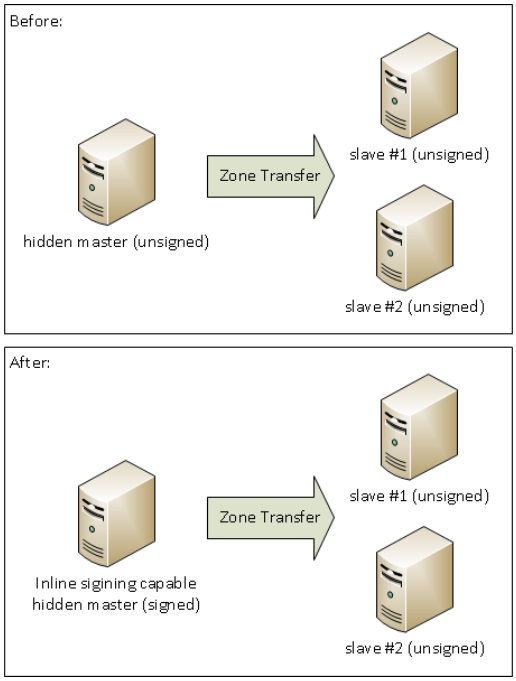
- As the signing process is separated from master, no (or only very little) change business logic on zone management are need to adjust in masters.
- Also separated signing unit help archive higher level security (when use of HSM), availability (when using HA) and flexibility (switching signing appliance)

The bad thing is:

- You need to manage new devices (the signer) and zones hosted inside.
- Amendments on zone transfer setting are required (on hidden master, signer and slaves).

Method 2: Master Server

This method sign zones directly in master (as show as above).



The good thing is:

- No additional device/unit required.
- Zone transfer settings remain unchanged.

The bad thing is:

- Its only limited choices of master name server which with signing ability.
- DNSSEC signed zone requires periodic re-signing, which is a cryptographic function that is CPU intensive. If your DNS zone is dynamic or changes frequently, it also adds to higher CPU loads on your master.
- If higher security is preferable, you may want to use HSM to protect DNSSEC keys (KSK, ZSK). Master must be capable to make use of HSM stored DNSSEC keys (say support PKCS #11).

Recommendation

In general "Bump in the Wire" is more recommended for DNS Hosting providers as it allows higher flexibility, manageability, security and availability.

Example Configuration

Capex	Description	Cost Estimation
Hardware		
1	Signer with Soft HSM system: - either physical or virtual machine - at least 16G RAM - 500G Hard Disk - Quad core 1.8GHz CPU or above	HK$100,000
2	Bind 9	FREE
3	SoftHSM	FREE
2nd Year Opex		
1	Signer system HW Maintenance - 24x7, ~20% of purchase price	HK$10,000
2	OS Support for signer and HSM	HK$6,300

Please note apart from the above example, there are many commercially available DNS/DNSSEC solution which can ease the setting up and integration. Gartner has a report on DDI (DHCP, DNS, IPAM) Solution:

https://www.gartner.com/doc/2991220/market-guide-dns-dhcp-ip

For choosing solution, please refer to:

Choosing a DNSSEC Solution – ZyTrax

http://www.zytrax.com/books/dns/info/choose-dnssec.html

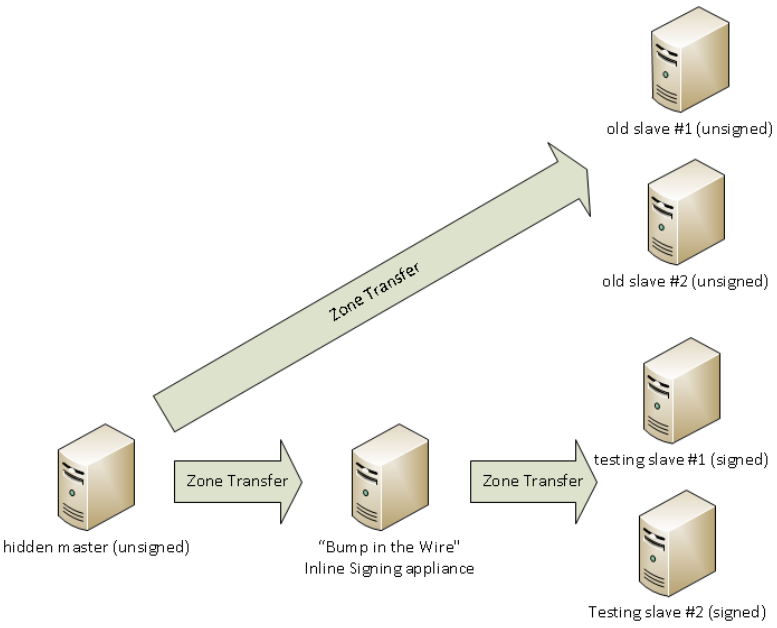
Basic Setup Steps

Below steps assumed that zone is signed using "Bump in the Wire" method:

1. Setup master to allow transfer zone from signing unit.
   (Note: for BIND, its done by set up allow-transfer, also-notify in named.conf)
2. Setup signing unit to request zone transfer from master
   (Note: for BIND, its done by setup masters in named.conf)
3. Setup signer unit to sign zones.
   (Note: for BIND, its done by setup key-directory, inline-signing, auto-dnssec in named.conf and use of dnssec-keygen rndc commands)
4. Setup signer unit to allow transfer zone from slaves
   (Note: for BIND, its done by set up allow-transfer, also-notify in named.conf)
5. Setup slaves to request zone transfer from signing unit
   (Note: for BIND, its done by setup masters in named.conf)
6. Update the DS records in parent zone
   (Note: for BIND, DS records can be retrieved by dnssec-keygen command)

Steps are similar for deployment, testing or even production field. However you are free to add variations in different stage to suit the need. For more detail steps using BIND, please refers to BIND DNSSEC Guide, ISC [1]

For example, during deploying DNSSEC to production, you can slightly change the steps (show as below) to allow DNSSEC zones to testing slaves which help you to final testing (say drill on key rollover, failover of master/signing unit, etc) before it go live.



Consideration on protecting private key

It's very important to protect the keys (KSK and ZSK) from being compromised. Because if hackers somehow get hold of the private key of your DNSKEY, they can serve all DNSSEC validating resolvers bogus record and make it look like valid one! That's why key should be regular rotated and well protected. Below are some available options:

- Stored in file system
- Stored and protected in signing appliance
- Stored and protected in Hardware Security Module (HSM)
- Stored offline (in HSM)

Organizations can select option that suit their own depends on what security level they would like to achieve. For those would like to have highest security standard, HSM are what they need. But there are so many types and vendors of HSM. Below are some links that we think useful for your reference:
https://wiki.opendnssec.org/display/DOCREF/HSM+Buyers%27+Guide
https://wiki.opendnssec.org/display/DOCREF/HSM

Implementation Consideration

For each DNSSEC Implementation, there are a few parameters that will need to be considered before implementation. See Appendix A for the list of parameters and their consideration. Settings for .hk were included for reference.

For operation and guidelines for implementation a DNSSEC using these parameters, please refers to:

- Chapter 9-11 of the Secure Domain Name System (DNS) Deployment Guide, NIST [2]
- DNSSEC Operational Practices, Version 2, rfc6781 [3]
- DNSSEC Operations: Setting the Parameters [4]

Please note, these are guidelines and each user should choose these parameters to suit their own need and environment.

Please also note, this guide does not include any steps to integrate your existing DNS provisioning system and the new DNSSEC signer system. You will also need to integrate your new DNSSEC signer system to update your DS records to parent zone through HKIRC's accredited registrar.

Maintenance Tasks

In order to maintenance an operating DNSSEC infrastructure there is a list of operational task that is required. These are:

1. Generation of public key-private key pair
2. Secure storage of private keys
3. Public key distribution
4. Zone signing
5. Key rollover (changing of keys)
6. Zone re-signing.

For more detail on the above task, refers to:

- BIND DNSSEC Guide, ISC [1]
- Chapter 9 of Secure Domain Name System (DNS) Deployment Guide, NIST [2]
- DNSSEC Operational Practices [3]

B. Enable DNSSEC Validation in DNS resolver for Internet Access Service and Testing

To do this you will need a DNS Server with DNSSEC validation enable. To enable a BIND based Recursive Server please follows the steps in Chapter 3 of BIND DNSSEC Guide, ISC [1].

During enabling DNSSEC validation on your Recursive DNS server, one key step is to let the server trust the root zone KSK. In the old days, it's statically defined and need to be updated every time root zone perform KSK rollover. Therefore it's recommended to configure your recursive DNS Server to automated updates of DNSSEC trust anchors which mentioned in RFC 5011. Most DNS resolver application already implemented it and configuration should be as simple as using static trusted key. Say BIND, you can enable it by setup managed-keys as below:

```
// Note 1: managed-keys is only supported since BIND 9.7
// Note 2: Below managed-keys make use current root zone KSK
//         (tagid: 19036) as initial-key, it may not work
//         after root zone KSK roll.

managed-keys {
   "." initial-key 257 3 8
    "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF
     FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
     bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD
     X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
     W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
     Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
     QxA+Uk1ihz0=";
};
```

It is recommended that you turn on DNSSEC Validation in DNS resolver on all your resolver, one-by-one, to monitor the effect of your signed zone.

Once your zone (s) is signed, you should be able to validate this using the below steps:

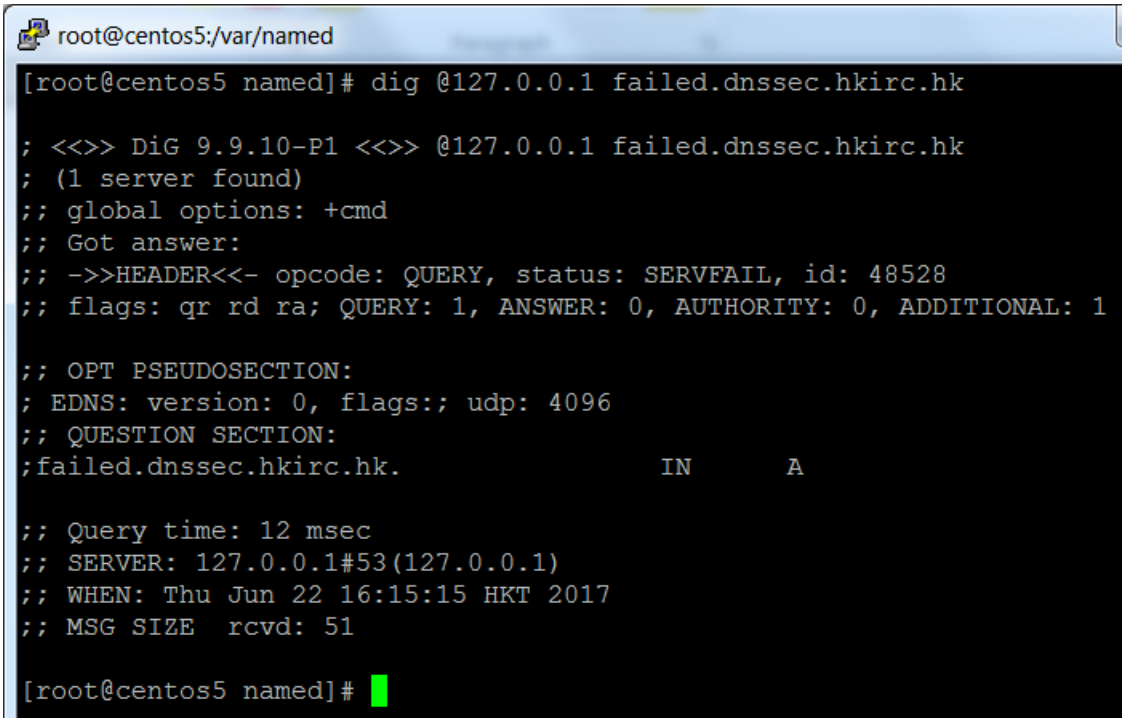
For a DNSSEC validation enable DNS Server, there are three possible states of RRsets:

- Insecure (it is not DNSSEC enabled)
- Secure (it is DNSSEC enabled and verification is successful)
- Bogus (it is DNSSEC enabled and verification is failed)

HKIRC setup 3 DN aim to facility you to testing your DNSSEC resolvers show as below:

	Testing Domain	Explanations
Insecure	disabled.dnssec.hkirc.hk	http://disabled.dnssec.hkirc.hk/
Secure	enabled.dnssec.hkirc.hk	http://enabled.dnssec.hkirc.hk/
Bogus	failed.dnssec.hkirc.hk	http://failed.dnssec.hkirc.hk/

You can use dig command to check whether your resolver behaviors as expected. For example, dig failed.dnssec.hkirc.hk to a DNSSEC validating resolver should return SERVFAIL and no RRsets returned.



HKIRC will offer free domains for testing purpose. Please feel free to contact our support team (email: info@hkirc.hk, tel.: 2319 2303) for this.

There are also other Web tools to check zones and signing

- http://dnssec-debugger.verisignlabs.com
- http://dnsviz.net
- https://www.zonemaster.fr
- http://zonecheck.org

Other Information

There is other DNSSEC information on the HKIRC's web site:

https://www.hkirc.hk/content.jsp?id=297

Below information are available from the above link:

- DNSSEC Practice Statement
- HKIRC DNSSEC Example Domain Names
- DNSSEC Example Domain Names:
    - disabled.dnssec.hkirc.hk
    - enabled.dnssec.hkirc.hk
    - failed.dnssec.hkirc.hk
- HKIRC DNSSEC Quick Start Guide

Ref.:-

1. BIND DNSSEC Guide, ISC: https://ftp.isc.org/isc/dnssec-guide/dnssec-guide.pdf
2. Secure Domain Name System (DNS) Deployment Guide, NIST: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-81-2.pdf
3. DNSSEC Operational Practices, Version 2, rfc6781: https://tools.ietf.org/rfc/rfc6781.txt
4. DNSSEC Operations: Setting the Parameters: http://www.dnssec-deployment.org/wp-content/uploads/sites/2/2012/02/Setting-the-Parameters-20091124032.pdf

Appendix A - DNSSEC Parameters

Parameter	Example value from .hk zone	Remarks
KSK Algorithm	8 (RSA/SHA256)	RSASHA1 and RSASHA256 are both common for TLD zones. But as SHA-1 hash has cryptanalysis issues, RSASHA256 was selected. Newer signature algorithms like Elliptic Curve are also being standardized and implemented which can provide similar level of trust with shorter key
KSK Key Length	2048 bit	Longer key make it harder to break the key, 2048 bit was selected as its common choice for TLD zones. But according to RFC 6781, it is estimated that most zones can safely use 1024-bit keys for at least the next ten years. Also noted same key length of difference algorithms cannot be direct compared
KSK DNSKEY TTL	1 day	It affect how long to wait for old DNSKEY records to expire from resolvers caches. DNSKEY only will change when KSK roll. Selection of this value only is to balance of operational cost (short TTL means more query) and recovery time (longer TTL block emergency rollover) so no fit-for-all values. As researched, even TLD zone ranged from 15 min to 6 days.
Publish DNSKEY of standby KSK	yes	Not a must. As research shows, not many TLD publish standby DNSKEY to zone. Only possible if signing solution generated standby key
KSK Keyroll Frequency	yearly	No fit-for-all value, from few-months to never
KSK Keyroll Mechanism	Double Signature Rollover Method	KSK signs only the DNSKEY RRset, using the double signature method does not substantially increase the size of the zone. In addition, the pre-publish method is more complicated for parent zones in a chain of trust and is therefore not recommended for KSK rollover.
ZSK Algorithm	8 (RSA/SHA256)	Refer to "KSK Algorithm"
ZSK Key Length	1024 bit	Originally 1024 bit was selected as ZSK will roll every month, so a shorter key should be enough. But as common TLDs are switching towards 2048 bit, we are researching on whether should we follow. More info for key length, see also "KSK Key Length"
ZSK DNSKEY TTL	1 day	Refer to "KSK DNSKEY TTL"
Publish DNSKEY of standby ZSK	yes	Refer to "Publish DNSKEY of standby KSK"

ZSK Keyroll Frequency	monthly	As this key will be used to sign lots of data, higher chance the key will be compromised, suggest to roll in a high frequency then KSK. As it does not involve parent DS update, full automation possible.
ZSK Keyroll Mechanism	Pre-publish Rollover Method	Because the ZSK signs all of the zone data and is used more frequently than the KSK, pre-publishing the new key prior to rollover is more efficient than double signing all of the zone data.
Authenticated Denial of Existence	NSEC3	If zone walking (enumeration) is not preferable, use NSEC3.
NSEC3PARAM Hash Algorithm	1 (SHA-1)	It's the only option
NSEC3PARAM Opt-Out flag	0 (clear, use Opt-Out)	NSEC3 setting. This option control whether create NSEC3 records for all delegations (no opt-out) or secure delegations only (opt-out). Opt-Out is designed for zones with large number of insecure delegations (say large TLDs). In most other circumstances, opt-out should not be deployed.
NSEC3PARAM Iterations	10	NSEC3 setting. Further increase difficulty of zone walking by increasing number of round of hashing performing (iterations=0, hashed once). Noted it also makes sending negative answers more expensive for authoritative DNS servers as well as validating answers on receiver side. A too high number can invite denial-of-service attacks again the zones authoritative servers. RFC 6781 suggest 100, but most TLD zones only use 0 to 20 iterations. Also see "Authenticated Denial of Existence"
NSEC3PARAM Salt	Used	NSEC3 setting. Further increase difficulty of zone walking by append "salt" on domain name before hashing it and occasionally change salt. See also "Authenticated Denial of Existence"
NSEC3 TTL	1 day	It should have the same TTL value as the SOA minimum TTL field (RFC 5155). See also "SOA Minimum".
NSEC3PARAM TTL	15 min	No guideline found on this value .hk selected to use same TTL value as SOA record. See also "SOA TTL"
SOA TTL	15 min	The zone TTL should be less than the signature validity period. The zone TTL should be long enough for a caching server to obtain and verify all RRs in the trust chain.

SOA Refresh	30 min	The SOA refresh value should be less than the signature validity period.
SOA Retry	15 min	It's typically some fraction of the refresh interval (RFC 1912)
SOA Expire	14 days	The SOA expiration should be one-third or one-fourth of the signature validity period
SOA Minimum	1 day	This value also control how long a negative response should be cached. The maximum value suggested by RFC 2308 for this parameter is 1 day. Our search show common TLD zone selected 15 min to 1 day.
RRSIG TTL	Follow RR	No point to use different TTL for RR and its RRSIG records.
RRSIG Signature Validity Period	30 days	Value should be large enough to ensure re-signing occurred before it expired, See also "RRSIG Signature Re-signing Frequency".
RRSIG Signature Re-signing Frequency	daily	Re-signing should occur at least one zone TTL before the signature validity period expires. See also "RRSIG Signature Validity Period"
Public DS of KSK with 2 algorithms	yes	Optionally. 1 DNSKEY can have 2 DS records (in different algorithms) published in parent zone. Most TLD only have 1 DS record.