

# DNSSEC性能確認手順書 ver. 1.2

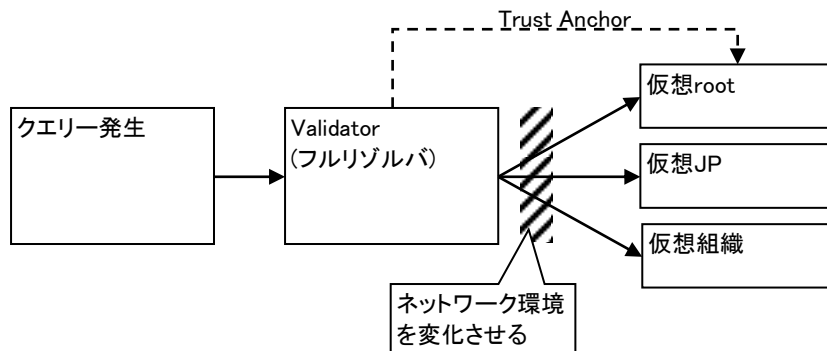
## 1. 目的

- ・DNSSEC検証によるフルリゾルバへの負荷、および権威DNSサーバへのトラフィックの変化を把握する。
- ・フルリゾルバと権威DNSサーバ間の通信路にある機器の影響を把握する。
- ・現在想定できる一般的な構成のハードウェア上での権威サーバの基本性能を計測する。

## 2. 検証環境

### 2. 1. サーバ構成

Validatorの検証および計測を行うためのネームサーバおよび負荷の構成は次のとおりである。



図中の四角はサーバを表す。

上図の構成において、通信路の影響をみるためValidatorと権威サーバ群の間でネットワーク環境を変化させる。(3. 1. 検証パターンを参照)

今回の計測では権威サーバとの通信路の影響を見ることを目的としているため、クライアント(クエリー発生)とValidatorの間のネットワーク環境は変化させない。

- ・仮想root 検証で使用する仮想的なrootゾーンをDNSSEC署名つきで保持する。  
仮想JPへの委任(NS, グルー)とDSを保持する。  
JP以外のクエリのためのワイルドカードを持つ。
- ・仮想JP 実JPのドメイン数に基づいた仮想JPゾーンをDNSSEC署名つきで保持する。  
各組織(~~.jp)への委任(NS, グルー)と署名つき組織に対するDSを保持する。
- ・仮想組織 各組織のゾーンを保持する。  
計測で用意するドメインのうち15万の組織に対して署名をつける
- ・Validator 検証対象のValidator(フルリゾルバ)。  
実装としてBIND 9およびUnboundを使用する。
- ・クエリー発生 Validatorに対してdigやqueryperfでDNSクエリ負荷を発生させる。

### 2. 2. 使用ソフトウェア

OS環境として CentOS 5.4 を用いる。

本計測ではネームサーバの実装として以下のソフトウェアを使用する。

実装	Ver	説明
BIND	9.7.1	権威サーバおよびValidator用
Unbound	1.4.5	Validator用 RSASHA256およびedns-buffer-size設定を使用するため最新安定版を使用
NSD	3.2.5	署名なし組織用権威サーバ用 BIND では100万以上のゾーンを保持するのが困難なのでNSD最新安定版を使用

また、計測のためのツールとして以下のソフトウェアを使用する。

実装	Ver	説明
queryperf	改造版	BIND付属のqueryperfに対して、JPRSにて改造を行ったバージョンを使用。一定qpsでの負荷を与えるため、Nominumのdnperfで代用可。
DSC	200911111630	クエリ内訳観測用 権威サーバおよびValidatorに設置

### 3. 検証条件・データ

#### 3. 1. 検証・計測パターン

##### A) Validator検証・計測パターン

通信路の条件およびネームサーバの設定パターンの組み合わせを以下に示した。  
正しく動くValidator(フルリゾルバ)と権威サーバでのオペレーションミスによって発生しうるものを選択した。

TCP			通す						遮断					
フラグメント			通す			遮断			通す			遮断		
MTU			1500	1280	576	1500	1280	576	1500	1280	576	1500	1280	576
DO=1	bufsize	ZSK 1024	●	○	○	●	○	○	●			○		
		ZSK 2048	●	○	○	○	○	○	●					
	bufsize	ZSK 1024	●	●	●	●	●	●						
		ZSK 2048	●	●	●	●	●	●						
DO=1 TA無し	bufsize	ZSK 1024	●	●	○	●	●	○	●	●	○	●	●	○
		ZSK 2048	●	●	○	●	●	○	●	●	○	●	●	○
	bufsize	ZSK 1024	●	●	●	●	●	●	●	●	●	●	●	●
		ZSK 2048	●	●	●	●	●	●	●	●	●	●	●	●
DO=0 TA無し	ZSK	1024	○	○	○	○	○	○	○	○	○	○	○	○
		2048	○	○	○	○	○	○	○	○	○	○	○	○
署名無し, DO=0			●	●	●	●	●	●	●	●	●	●	●	●

パターン数 58 ●=BIND 9/Unboundで計測  
42 ○=BIND 9でのみ計測

上記パターンに対し、以下のValidatorおよび負荷のパターンで計測する。

項目	バリエーション
Validator	BIND 9 / Unbound
クエリ負荷	1000qps / 10000qps

検証パターンにおける条件についての説明は以下の通り。

署名無し	署名無しゾーンを権威サーバに設定し、署名がない場合の負荷の差を計測する。
TA(Trust Anchor)無し	権威サーバに署名つきゾーンが設定されているが、フルリゾルバに、rootのTrust Anchor設定がない(検証できない)パターン。
DO=0	サーバ側は署名や検証を提供するが、フルリゾルバがDO=0のリクエストが出す場合。BIND 9ではEDNS0を無効にするしかないため、bufsizeのバリエーション無し。
bufsize	Validator から権威サーバへのリクエストのEDNS0 UDPパッファサイズが小さい場合を検証
MTU	Validator から権威サーバへの通信経路において、MTU値が小さくフラグメントが起こりうるパターンを確認
フラグメント	Validator から権威サーバへの通信経路において、フラグメントが起こる条件だが、フラグメント化できずドロップする場合を検証
TCP	Validator から権威サーバへの通信経路において、TCP 53番ポートが開いていない場合を検証
ZSK	JPゾーン署名に使用するZSKの長さを1024bitか2048bitにして署名する。ゾーンデータを切り替える。

##### B) 権威サーバ計測パターン

権威サーバについては基本性能計測であるので、以下のパターンを計測する。

ネームサーバおよび通信路について

- ・通信路は正常とする(MTU1500,フラグメントの遮断はなし)
- ・署名つきゾーン(JPゾーンはZSK1024bit)

クエリ負荷条件

- ・DO=1, bufsize=4096

- ・UDPIによるクエリで計測
- ・負荷パターン: 1000qps, 10000qps, 限界負荷

### 3. 2. 署名パラメータ

2009/12/1 現在のRFCから選定し、以下の条件を用いることとした。

ゾーン	root	jp	組織
署名アルゴリズム	RSASHA256		
鍵の数	KSK	2本 (*1)	
	ZSK	2本 (*1)	
鍵の数と長さ	KSK	2048bit	
	ZSK	1024bit	1024または2048bit
鍵データ	ゾーンごとに別の鍵を用意、入れ替えはしない		
NSEC3	用しない	SHA1/Iter 10/salt固定/opt-out使用	使用しない
DSダイジェストアルゴリズム	SHA256		

\*1 署名鍵のロールオーバーを考慮して2本用意し、そのうち1本で署名する。1本は事前公開用と仮定。

### 3. 3. ゾーンデータ

ゾーンデータは、各ゾーンに対して、署名なし・署名ありの両方を用意する。  
さらに、JPゾーンの署名ありの場合はZSKが1024bitと2048bitがあり計3パターンとなる。

	署名なし	署名つき(ZSK1024bit)	署名つき(ZSK2048bit)	
仮想root	A		B,C	
仮想JP	A		B	C
仮想組織(上位15万)	A		B,C	
仮想組織(上位15万以外)	A,B,C			

A: 検証パターン「署名なし」で使用    B: 検証パターン「ZSK1024」で使用    C: 検証パターン「ZSK2048」で使用

ドメイン数(組織数)は、実JPゾーンの実ドメイン数+100万とする。  
※本計測の性格上、権威サーバの保持するゾーンのデータサイズはあまり影響を与えないと思われるため、ドメイン数のバリエーションは持たせない。

ゾーン	基本ゾーンデータ	署名つきの場合
仮想root	JPへの委任(NS,A,AAAA)	RRSIG/NSEC
	それ以外のドメインに対するwildcard	JPのDS
仮想JP	ns.example.com. A	
	各組織への委任(NS,A,AAAA) ×ドメイン数	ドメインのうち15万の組織に対するDS RRSIG/NSEC3
各組織 (ドメイン数分)	@ IN SOA	RRSIG/NSEC (署名つき組織のみ)
	@ IN DNSKEY (署名つき組織のみ)	
	@ IN NS ns	
	@ IN NS ns.example.com.	
	@ IN A	
	@ IN MX	
	ns IN A	
	www IN A	
	www IN AAAA	
	www IN MX	

署名対象とする(組織)ドメインは、実JPドメインのうちa.dns.jpで問い合わせの多い、上位15万組織とする  
実JP以外の追加するダミードメインには署名つきゾーンは作らない。

### 3. 4. クエリパターン

負荷計測時には下記のクエリデータを使用する

- ・各自で取得したクエリログをqueryperfの入力形式に変換したものを用いる。
- ・jp以外のクエリは除外する(arpaも除外)
- ・クエリのドメイン名については変換などは行わない。
  - wwwやゾーン頂点以外へのAクエリは各組織レベルでNXDOMAINエラーとなる。
  - JPLレジストリに登録されていないドメインは、JPLレベルでNXDOMAINエラーとなる。
- ・5分間、限界負荷を掛けられるだけのデータを用意する。

### 3. 5. 計測項目

本計測では以下の項目を計測する。

A) Validator検証および計測

A-1) dig による検証

- ・名前解決の成否
- ・DNSSEC検証の成否
- ・名前解決時に観測される挙動。(タイムアウト、時間がかかる etc)

A-2) 性能計測

- ・ValidatorのCPU負荷、メモリ使用量  
ps による%CPU,VSZ,RSS  
uptime によるload average
- ・Validator-権威サーバ間のクエリ量、内訳  
DSC による権威サーバ毎のクエリ数
- ・各権威サーバ、Validatorのクエリ統計量  
BIND 9のrndc statsによる出力結果 (※NSD, Unboundでは取得できない)

B) 権威サーバ計測

- ・権威サーバのCPU負荷、メモリ使用量
- ・限界負荷計測時の qps

4. 準備

OSおよび検証に使用するソフトウェア(BIND, Unbound, OpenSSL等)のインストールについてはここでは記載しない。

※以下の処理において、組織ドメインの数に関する部分はスクリプトなどにより自動化・並列化を行う。

4. 1. ゾーンデータの準備

1)未署名ゾーンの作成

「3. 3. ゾーンデータ」に示した内容から基本ゾーンデータに示したゾーンファイルを作成する。  
各グルーのIPアドレスはそれぞれ下位のネームサーバを指すように指定する。

- ・root ゾーン
- ・JP ゾーン (実ドメイン + ダミードメイン100万 を保持)
- ・各組織ゾーン群 (実ドメイン数 + ダミードメイン100万個のゾーン)

2)署名鍵の準備

root, JP, および署名対象の組織に対する署名鍵を「3. 2. 署名パラメータ」の通りに生成する。

- ・KSKの生成(example.jp.は実際のゾーン名に置き換える)  
`dnssec-keygen -a RSASHA256 -b 2048 -f KSK -r /dev/urandom example.jp.`
- ・ZSKの生成(example.jp.は実際のゾーン名に置き換える、鍵長2048bitの場合は-b 2048とする)  
`dnssec-keygen -a RSASHA256 -b 1024 -r /dev/urandom example.jp.`

※ デフォルトの/dev/randomを使用すると大量の鍵生成が困難であるので、/dev/urandom を使用する

3)ゾーンの署名

各ゾーンに対し、署名つきゾーンを生成する。

なお、署名の有効期間はdnssec-signzoneコマンドのデフォルト(1ヵ月後)では試験期間中に有効期限切れを起こす可能性があるため、一年(31536000秒)後とする。  
DSレコードを上位のゾーンに追加するときはSHA256(アルゴリズム番号=2)のものを採用する。

- ・各組織の署名 (署名つき組織)  
作成したDNSKEYデータ(K**example.jp.**+008+**01234**.key)をゾーンファイルに追加する。  
example.jp は実際のゾーン名、example.jp.zoneは作成したゾーンファイルを指定する。  
`dnssec-signzone -g -e now+31536000 -o example.jp example.jp.zone`
- ・JPの署名

1024bit, 2048bit の両方のZSKに対して以下の署名処理を行う。  
作成したDNSKEYデータ(Kjp.+008+01234.key)ゾーンファイルに追加する。  
各組織へのdnssec-signzoneで生成されたDSレコードをゾーンに追加する。  
以下のコマンドで署名する。cafe の部分は使用するソルトを指定する

```
dnssec-signzone -g -e now+31536000 -o jp. -3 cafe -H 10 -A jp.zone
```

・rootの署名

作成したDNSKEYデータをゾーンファイルに追加する。  
JPへのdnssec-signzoneで生成されたDSレコードをゾーンに追加する。  
以下のコマンドで署名する。

```
dnssec-signzone -e now+31536000 -o . root.zone
```

#### 4. 2. ネームサーバの準備

1) 権威サーバについては各ゾーンを提供するように設定する。

・仮想root

rootの署名なし、署名つきゾーンを提供する named.conf 2パターンを作成する。

・仮想JP

jpの署名なし、署名つき(ZSK1024bit)、署名つき(ZSK2048bit)の3パターンのゾーンを提供するnamed.confを作成する。

・仮想組織

署名対象組織(上位15万)については、署名つき、署名無しの2パターンのゾーンを提供するnamed.confを作成する。

署名なし組織については、NSDで提供する(\*2)ため、nsd.confを作成し、zonecでゾーンDBを作成する。

\*2 BIND 9 で100万単位のゾーンをホストするのが難しいため。

2) Validatorについては以下のように設定する。

・BIND 9/Unbound の2種類を設定する

・rootヒントは構築した仮想rootサーバのIPアドレスを記述する。

・Trust Anchor としてrootの署名にしようとしたKSKを trusted-keys に記述する。

Unbound の場合、trust-anchor/trust-anchor-file/trusted-keys-fileを使用する。

#### 4. 3. クエリデータの準備

3. 4. に示した、queryperf用のクエリデータを作成する。

#### 5. 計測手順

##### 5. 1. Validatorの各パターンによる挙動変化の計測

Validatorに対する計測は、「名前解決と検証の確認」および「性能計測」からなる。

以下の手順は、Validatorが保持するキャッシュの影響を排除するため、1つのパターンをテストするたびに、Validatorを起動しなおす。

a) digによるDNS名前解決と、DNSSEC検証の確認

「3. 1. 検証・計測パターン」について、考えられる組み合わせに対して、以下の確認を行う。

1. パターンによって、以下の設定を変更する。

・Validatorサーバのネットワーク設定を変更する。(MTU, TCP, フラグメント)

・Validatorサーバの設定ファイルを変更する。(DO=0/1, TAの設定)

・権威サーバに設定するゾーンデータを変更する。(ZSK=1024/2048, 署名なし)

2. 上記の設定後、Validatorおよび権威サーバが起動している状態で、Validatorサーバ上で dig により下記の例にあるコマンドにて確認を行う。  
dig の出力結果をみて、名前解決の成否・検証の成否を確かめる。

署名無しの場合

```
dig @localhost example.jp. A
```

署名ありの場合

```
dig @localhost +dnssec example.jp. A
```

b) 名前解決ができるパターンに対し、Validatorの負荷と権威サーバへのクエリ内容を計測する。

「3. 1. 検証・計測パターン」について、対象となっている組み合わせに対し以下の確認を行う。

1. パターンによって、a)と同様にネットワークおよびサーバの設定を変更する。
2. Validator サーバ上で、負荷計測ツール(※)を起動しCPU使用率およびメモリ使用量の計測を開始する。

※ CPU使用率、メモリ使用量、ロードアベレージなどを計測するスクリプトを用意する。

3. Validatorおよび権威サーバ上で、DSC(DSC Collector)を稼働させる。
4. クエリー発生機上でqueryperf(改造版)による負荷テストを行う。DOビットによってコマンドを使い分ける。  
-i オプションに送信間隔をミリ秒単位で指定する。下記の例では0.1msなので10000qpsで送信する。  
#適時、dnspertなどをツールと読み替える

```
DO=0
queryperf -d query.txt -s 192.0.2.1 -l 300 -i 0.1
DO=1
queryperf -d query.txt -s 192.0.2.1 -D -l 300 -i 0.1
```

5. 負荷を掛け終わったら、Validatorおよび権威サーバで起動した負荷計測ツール/DSCを終了させる。

上記手順において、検証および計測の間でのパターンの切り替えは以下のように行う。

・権威サーバのゾーン設定の切り替え

今回のパターンでは使用するゾーンについて、  
署名なし / 1024bit ZSKによる署名つき / 2048bit ZSKによる署名つき  
の3パターンがある。  
「3. 3. ゾーンデータ」に示したパターン毎にゾーン設定を変更後、ネームサーバを再起動させる。

・ValidatorのTrust Anchor設定の切り替え

Trust Anchor 無しの場合は、Validator の trusted-keys 設定を外してネームサーバを再起動する。

・Validatorのbufsizeの切り替え

Validator の設定 edns-udp-size の値を修正して、ネームサーバを再起動する。

・ネットワーク環境の切り替え

権威サーバとValidatorサーバ間の通信路の状況を変化させるため、  
Validator サーバのネットワーク設定を、iptables および ifconfig によって変更する。  
以下の説明では、対象となるネットワークI/Fをeth0としている  
試験用の環境に応じてipfilterや外部ファイアウォールなどの機能を使う必要がある。

TCP (ValidatorからTCP接続できないようにする)  
iptables -I OUTPUT -p tcp -m tcp --dport 53 -j DROP

フラグメントを落とす(権威サーバからの応答パケットのフラグメントを落とす)  
iptables -I INPUT -p udp -m udp --sport 53 -m length --length 1480:65535 -j DROP  
※1480の部分はMTUから20引いた値を指定する。  
(conntrackモジュールにより、iptables -I INPUT -f -j DROPでは落とせないため)

MTUの変更  
ifconfig eth0 mtu 1280

## 5. 2. 権威サーバの性能計測

権威サーバに対する負荷計測は、負荷パターンごとに以下の手順で行う。

1. JP権威サーバを稼働させる。
2. Validator計測と同様にCPU負荷、メモリ使用量を観測するための計測ツールを起動する。

3. queryperf で負荷をかける。負荷の掛け方について5. 1 b)の手順と同様。  
負荷のパターンは、1000qps / 10000qps / 限界負荷の3パターンを行う。限界負荷の場合は -i オプションを外す。
4. 計測ツールを終了させる。

## 6. 結果

計測によって得られた結果をグラフなどを用いてまとめる。